

**Information technology—  
Telecommunications and information exchange between systems—  
Local and metropolitan area networks—Specific requirements—**

**Part 3: Carrier Sense Multiple Access with  
Collision Detection (CSMA/CD) access method  
and physical layer specifications**

SECTION TWO: This section includes Clauses 21 through 33 and Annexes 22A through 32A.

**21. Introduction to 100 Mb/s baseband networks, type 100BASE-T**

**21.1 Overview**

100BASE-T couples the IEEE 802.3<sup>®</sup> CSMA/CD MAC with a family of 100 Mb/s Physical Layers. While the MAC can be readily scaled to higher performance levels, new Physical Layer standards are required for 100 Mb/s operation.

The relationships between 100BASE-T, the existing IEEE 802.3<sup>®</sup> (CSMA/CD MAC), and the ISO/IEC Open System Interconnection (OSI) reference model is shown in Figure 21-1.

100BASE-T uses the existing IEEE 802.3<sup>®</sup> MAC layer interface, connected through a Media-Independent Interface layer to a Physical Layer entity (PHY) sublayer such as 100BASE-T4, 100BASE-TX, or 100BASE-FX.

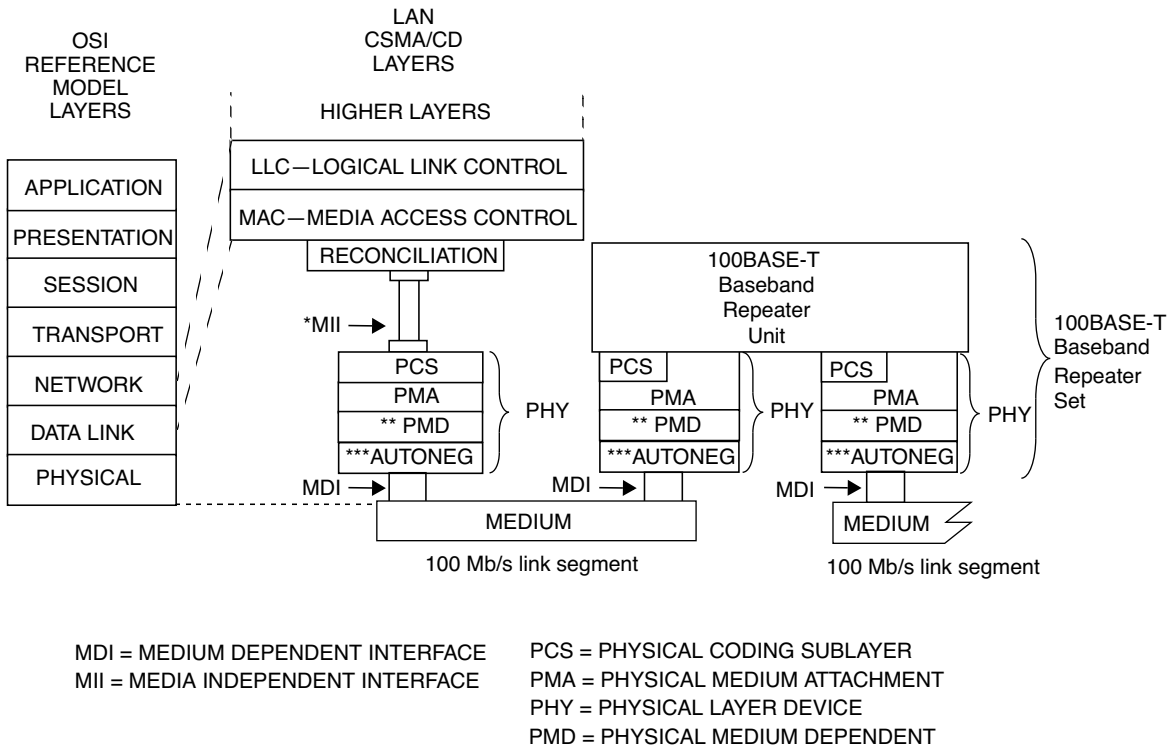
100BASE-T extends the IEEE 802.3<sup>®</sup> MAC to 100 Mb/s. The bit rate is faster, bit times are shorter, packet transmission times are reduced, and cable delay budgets are smaller—all in proportion to the change in bandwidth. This means that the ratio of packet duration to network propagation delay for 100BASE-T is the same as for 10BASE-T.

**21.1.1 Reconciliation Sublayer (RS) and Media Independent Interface (MII)**

The Media Independent Interface (Clause 22) provides an interconnection between the Media Access Control (MAC) sublayer and Physical Layer entities (PHY) and between PHY Layer and Station Management (STA) entities. This MII is capable of supporting both 10 Mb/s and 100 Mb/s data rates through four bit wide (nibble wide) transmit and receive paths. The Reconciliation sublayer provides a mapping between the signals provided at the MII and the MAC/PLS service definition.

**21.1.2 Physical Layer signaling systems**

The following portion of this standard specifies a family of Physical Layer implementations. 100BASE-T4 (Clause 23) uses four pairs of ISO/IEC 11801 Category 3, 4, or 5 balanced cabling. 100BASE-TX (Clauses



\* MII is optional for 10 Mb/s DTEs and for 100 Mb/s systems and is not specified for 1 Mb/s systems.  
 \*\* PMD is specified for 100BASE-X only; 100BASE-T4 does not use this layer.  
 Use of MII between PCS and Baseband Repeater Unit is optional.  
 \*\*\* AUTONEG is optional.

**Figure 21-1 — Architectural positioning of 100BASE-T**

24 and 25) uses two pairs of Category 5 balanced cabling or 150 Ω shielded balanced cabling as defined by ISO/IEC 11801. 100BASE-FX (Clauses 24 and 26) uses two multi-mode fibers. FDDI (ISO/IEC 9314 and ANSI X3T12) Physical Layers are used to provide 100BASE-TX and 100BASE-FX physical signaling channels, which are defined under 100BASE-X (Clause 24). 100BASE-T2 (Clause 32) uses two pairs of ISO/IEC 11801 Category 3, 4, or 5 balanced cabling.

### 21.1.3 Repeater

Repeater sets (Clause 27) are an integral part of any 100BASE-T network with more than two DTEs in a collision domain. They extend the physical system topology by coupling two or more segments. Multiple repeaters are permitted within a single collision domain to provide the maximum path length.

### 21.1.4 Auto-Negotiation

Auto-Negotiation (Clause 28) provides a linked device with the capability to detect the abilities (modes of operation) supported by the device at the other end of the link, determine common abilities, and configure for joint operation. Auto-Negotiation is performed out-of-band using a pulse code sequence that is compatible with the 10BASE-T link integrity test sequence.

### 21.1.5 Management

Managed objects, attributes, and actions are defined for all 100BASE-T components (Clause 30). This clause consolidates all IEEE 802.3<sup>®</sup> management specifications so that 10 Mb/s, 100 Mb/s or 10/100 Mb/s agents can be managed by existing 10 Mb/s-only network management stations with little or no modification to the agent code.

### 21.2 References

See 1.3.

### 21.3 Definitions

See 1.4.

### 21.4 Abbreviations

See 1.5.

### 21.5 State diagrams

State machine diagrams take precedence over text.

The conventions of 1.2 are adopted, with the following extensions.

#### 21.5.1 Actions inside state blocks

The actions inside a state block execute instantaneously. Actions inside state blocks are atomic (i.e., uninterruptible).

After performing all the actions listed in a state block one time, the state block then continuously evaluates its exit conditions until one is satisfied, at which point control passes through a transition arrow to the next block. While the state awaits fulfillment of one of its exit conditions, the actions inside do not implicitly repeat.

The characters • and [bracket] are *not* used to denote any special meaning.

Valid state actions may include .indicate and .request messages.

No actions are taken outside of any state block.

#### 21.5.2 State diagram variables

Once set, variables retain their values as long as succeeding blocks contain no references to them.

Setting the parameter of a formal interface message assures that, on the next transmission of that message, the last parameter value set will be transmitted.

Testing the parameter of a formal interface messages tests the value of that message parameter that was received on the last transmission of said message. Message parameters may be assigned default values that persist until the first reception of the relevant message.

### 21.5.3 State transitions

The following terms are valid transition qualifiers:

- a) Boolean expressions
- b) An event such as the expiration of a timer: timer\_done
- c) An event such as the reception of a message: PMA\_UNITDATA.indicate
- d) An unconditional transition: UCT
- e) A branch taken when other exit conditions are not satisfied: ELSE

Any open arrow (an arrow with no source block) represents a global transition. Global transitions are evaluated continuously whenever any state is evaluating its exit conditions. When a global transition becomes true, it supersedes all other transitions, including UCT, returning control to the block pointed to by the open arrow.

### 21.5.4 Operators

The state machine operators are shown in Table 21–1.

**Table 21–1 – State machine operators**

Character	Meaning
*	Boolean AND
+	Boolean OR
^	Boolean XOR
!	Boolean NOT
<	Less than
≤	Less than or equal to
=	Equals (a test of equality)
≠	Not equals
≥	Greater than or equal to
>	Greater than
()	Indicates precedence
←	Assignment operator
∈	Indicates membership
∉	Indicates nonmembership
	Catenate
ELSE	No other state condition is satisfied

## 21.6 Protocol Implementation Conformance Statement (PICS) proforma

### 21.6.1 Introduction

The supplier of a protocol implementation that is claimed to conform to any 100 Mb/s portion of this standard shall complete a Protocol Implementation Conformance Statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. A PICS is included at the end of each clause as appropriate. The PICS can be used for a variety of purposes by various parties, including the following:

- a) As a checklist by the protocol implementor, to reduce the risk of failure to conform to the standard through oversight;
- b) As a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma, by the supplier and acquirer, or potential acquirer, of the implementation;
- c) As a basis for initially checking the possibility of interworking with another implementation by the user, or potential user, of the implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICS);
- d) As the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation, by a protocol tester.

### 21.6.2 Abbreviations and special symbols

The following symbols are used in the PICS proforma:

M	mandatory field/function
!	negation
O	optional field/function
O.<n>	optional field/function, but at least one of the group of options labeled by the same numeral <n> is required
O/<n>	optional field/function, but one and only one of the group of options labeled by the same numeral <n> is required
X	prohibited field/function
<item>:	simple-predicate condition, dependent on the support marked for <item>
<item1>*<item2>:	AND-predicate condition, the requirement must be met if both optional items are implemented

### 21.6.3 Instructions for completing the PICS proforma

The first part of the PICS proforma, Implementation Identification and Protocol Summary, is to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

The main part of the PICS proforma is a fixed-format questionnaire divided into subclauses, each containing a group of items. Answers to the questionnaire items are to be provided in the right-most column, either by simply marking an answer to indicate a restricted choice (usually Yes, No, or Not Applicable), or by entering a value or a set or range of values. (Note that there are some items where two or more choices from a set of possible answers can apply; all relevant choices are to be marked.)

Each item is identified by an item reference in the first column; the second column contains the question to be answered; the third column contains the reference or references to the material that specifies the item in the main body of the standard; the sixth column contains values and/or comments pertaining to the question to be answered. The remaining columns record the status of the items—whether the support is mandatory, optional or conditional—and provide the space for the answers.

The supplier may also provide, or be required to provide, further information, categorized as either Additional Information or Exception Information. When present, each kind of further information is to be provided in a further subclause of items labeled A<i> or X<i>, respectively, for cross-referencing purposes, where <i> is any unambiguous identification for the item (e.g., simply a numeral); there are no other restrictions on its format or presentation.

A completed PICS proforma, including any Additional Information and Exception Information, is the Protocol Implementation Conformance Statement for the implementation in question.

Note that where an implementation is capable of being configured in more than one way, according to the items listed under Major Capabilities/Options, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's configuration capabilities, if that would make presentation of the information easier and clearer.

#### **21.6.4 Additional information**

Items of Additional Information allow a supplier to provide further information intended to assist the interpretation of the PICS. It is not intended or expected that a large quantity will be supplied, and the PICS can be considered complete without any such information. Examples might be an outline of the ways in which a (single) implementation can be set up to operate in a variety of environments and configurations; or a brief rationale, based perhaps upon specific application needs, for the exclusion of features that, although optional, are nonetheless commonly present in implementations.

References to items of Additional Information may be entered next to any answer in the questionnaire, and may be included in items of Exception Information.

#### **21.6.5 Exceptional information**

It may occasionally happen that a supplier will wish to answer an item with mandatory or prohibited status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No pre-printed answer will be found in the Support column for this; instead, the supplier is required to write into the Support column an X<i> reference to an item of Exception Information, and to provide the appropriate rationale in the Exception item itself.

An implementation for which an Exception item is required in this way does not conform to this standard.

Note that a possible reason for the situation described above is that a defect in the standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

#### **21.6.6 Conditional items**

The PICS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply—mandatory, optional, or prohibited—are dependent upon whether or not certain other items are supported.

Individual conditional items are indicated by a conditional symbol of the form “<item>:<s>” in the Status column, where “<item>” is an item reference that appears in the first column of the table for some other item, and “<s>” is a status symbol, M (Mandatory), O (Optional), or X (Not Applicable).

If the item referred to by the conditional symbol is marked as supported, then 1) the conditional item is applicable, 2) its status is given by “<s>”, and 3) the support column is to be completed in the usual way. Otherwise, the conditional item is not relevant and the Not Applicable (N/A) answer is to be marked.

Each item whose reference is used in a conditional symbol is indicated by an asterisk in the Item column.

### **21.7 Relation of 100BASE-T to other standards**

Suitable entries for Table G.1 of ISO/IEC 11801, Annex G would be as follows:

- a) Within the section Balanced Cabling Link Class C (specified up to 16 MHz):  
 CSMA/CD 100BASE-T2                   ISO/IEC 8802-3/DAD 1995   2  
 CSMA/CD 100BASE-T4\*                ISO/IEC 8802-3/DAD 1995   4
- b) Within the section Optical Link:  
 CSMA/CD 100BASE-FX                 ISO/IEC 8802-3/DAD 1995   2
- c) Within the section Balanced Cabling Link Class D (defined up to 100 MHz):  
 CSMA/CD 100BASE-TX                 ISO/IEC 8802-3/DAD 1995   2

\*To support 100BASE-T4 applications, Class C links shall have a NEXT value of at least 3 dB in excess of the values specified in 6.2.4.

Suitable entries for Table G.4 of ISO/IEC 11801, Annex G, would be as follows:

	Balanced cabling							Performance based cabling per Clause 6												
	per Clauses 5, 7, and 8							Class A			Class B			Class C			Class D			
	C a t 3 1 0 0 Ω	C a t 4 1 0 0 Ω	C a t 5 1 0 0 Ω	C a t 3 1 2 0 Ω	C a t 4 1 2 0 Ω	C a t 5 1 2 0 Ω	1 5 0 Ω	1 0 0 Ω	1 2 0 Ω	1 5 0 Ω	1 0 0 Ω	1 2 0 Ω	1 5 0 Ω	1 0 0 Ω	1 2 0 Ω	1 5 0 Ω	1 0 0 Ω	1 2 0 Ω	1 5 0 Ω	
802.3: 100BASE-T2	I <sup>a</sup>	I	I	I	I	I	I							I				I <sup>a</sup>	I	I
802.3: 100BASE-T4	I <sup>a</sup>	I	I		I	I								I				I <sup>a</sup>	I	
802.3: 100BASE-TX			I <sup>a</sup>				I <sup>a</sup>											I <sup>a</sup>		I <sup>a</sup>

<sup>a</sup>802.3<sup>®</sup> imposes additional requirements on propagation delay.

A suitable entry for Table G.5 of ISO/IEC 11801, Annex G, would be as follows:

	Fibre			Optical link per Clause 8								
	per 5, 7, and 8			Horizontal			Building backbone			Campus backbone		
	62.5/ 125 mm MMF	50/ 125 mm MMF	10/ 125 mm MMF	62.5/ 125 mm MM F	50/ 125 mm MM F	10/ 125 mm MM F	62.5/ 125 mm MM F	50/ 125 mm MM F	10/ 125 mm MM F	62.5/ 125 mm MM F	50/ 125 mm MM F	10/ 125 mm MM F
802.3: 100BASE-FX	N	I		N	I		N	I		N	I	

## 21.8 MAC delay constraints (exposed MII)

100BASE-T makes the following assumptions about MAC performance. These assumptions apply to any MAC operating in half duplex mode with an exposed MII.

**Table 21–2—MAC delay assumptions (exposed MII)**

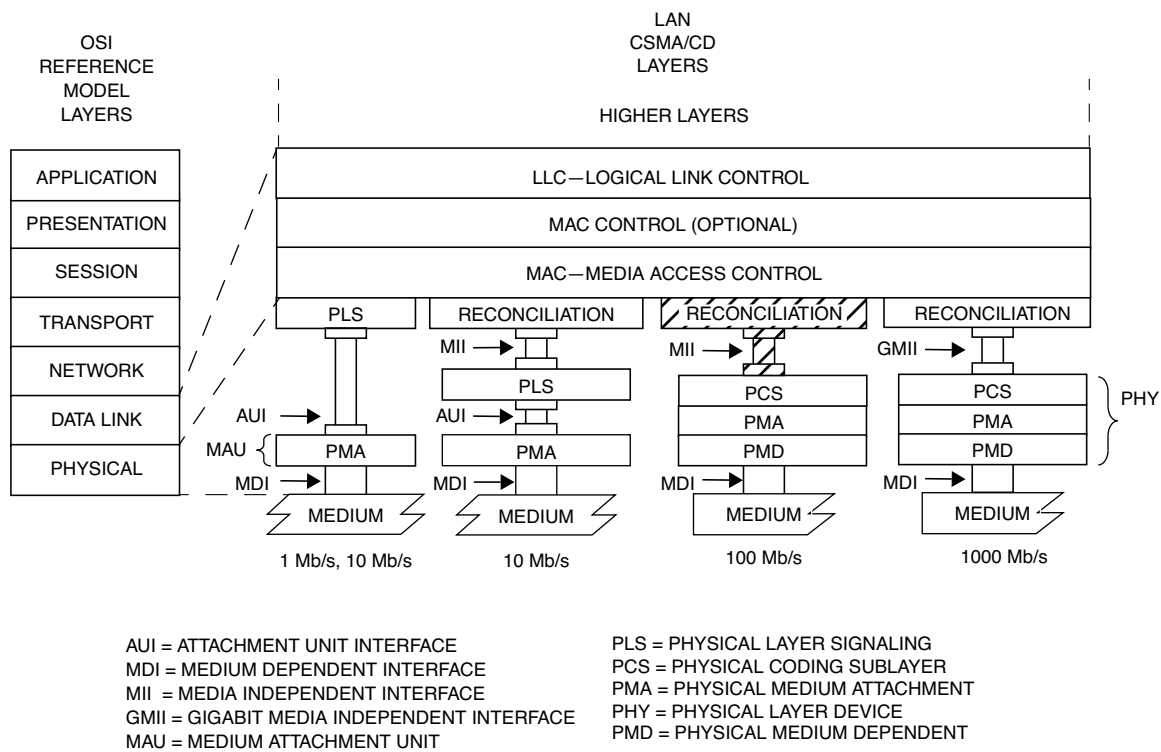
Sublayer measurement points	Event	Min (bits)	Max (bits)	Input timing reference	Output timing reference
MAC ⇔ MII	MAC transmit start to TX_EN sampled		4		TX_CLK rising
	CRS assert to MAC detect	0	8		
	CRS de-assert to MAC detect	0	8		
	CRS assert to TX_EN sampled (worst case nondeferred transmit)		16		TX_CLK rising
	COL assert to MAC detect	0	8		
	COL de-assert to MAC detect	0	8		
	COL assert to TXD = Jam sampled (worst-case collision response)		16		TX_CLK rising; first nibble of jam



## 22. Reconciliation Sublayer (RS) and Media Independent Interface (MII)

### 22.1 Overview

This clause defines the logical, electrical, and mechanical characteristics for the Reconciliation Sublayer (RS) and Media Independent Interface (MII) between CSMA/CD media access controllers and various PHYs. Figure 22–1 shows the relationship of the Reconciliation sublayer and MII to the ISO/IEC OSI reference model.



**Figure 22–1—MII location in the protocol stack**

The purpose of this interface is to provide a simple, inexpensive, and easy-to-implement interconnection between Media Access Control (MAC) sublayers and PHYs for data transfer at 10 Mb/s and 100 Mb/s, and between Station Management (STA) and PHY entities supporting data transfer at 10 Mb/s or above (see 22.2.4).

This interface has the following characteristics:

- It is capable of supporting 10 Mb/s and 100 Mb/s rates for data transfer, and management functions for PHYs supporting data transfer at 10 Mb/s or above (see 22.2.4).
- Data and delimiters are synchronous to clock references.
- It provides independent four bit wide transmit and receive data paths.
- It uses TTL signal levels, compatible with common digital CMOS ASIC processes.
- It provides a simple management interface.
- It is capable of driving a limited length of shielded cable.
- It provides full duplex operation.

### 22.1.1 Summary of major concepts

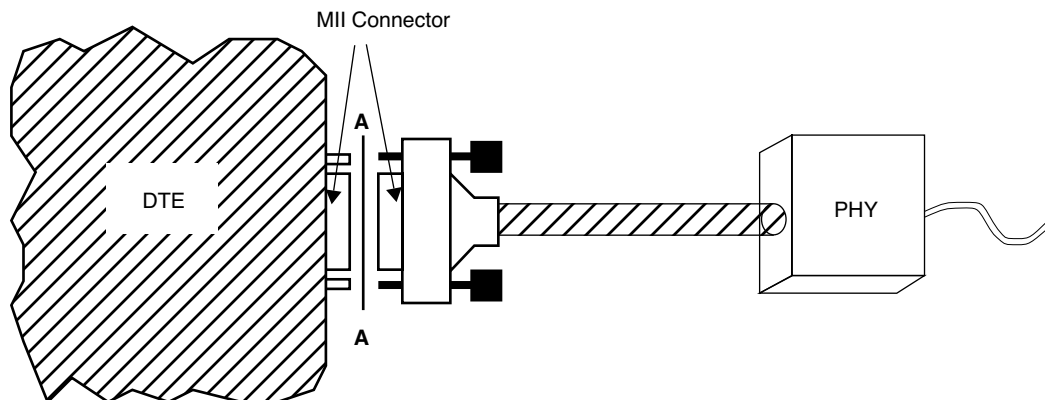
- a) Each direction of data transfer is serviced with seven (making a total of 14) signals: Data (a four-bit bundle), Delimiter, Error, and Clock.
- b) Two media status signals are provided. One indicates the presence of carrier, and the other indicates the occurrence of a collision.
- c) A management interface comprised of two signals provides access to management parameters and services.
- d) The Reconciliation sublayer maps the signal set provided at the MII to the PLS service definition specified in Clause 6.

### 22.1.2 Application

This clause applies to the interface between MAC sublayer and PHYs, and between PHYs and Station Management entities. The implementation of the interface may assume any of the following three forms:

- a) A chip-to-chip (integrated circuit to integrated circuit) interface implemented with traces on a printed circuit board.
- b) A motherboard to daughterboard interface between two or more printed circuit boards.
- c) An interface between two printed circuit assemblies that are attached with a length of cable and an appropriate connector.

Figure 22–2 provides an example of the third application environment listed above. All MII conformance tests are performed at the mating surfaces of the MII connector, identified by the line A-A.



**Figure 22–2—Example application showing location of conformance test**

This interface is used to provide media independence for various forms of unshielded twisted-pair wiring, shielded twisted-pair wiring, fiber optic cabling, and potentially other media, so that identical media access controllers may be used with any of these media.

To allow for the possibility that multiple PHYs may be controlled by a single Station Management entity, the MII management interface has provisions to accommodate up to 32 PHYs, with the restriction that a maximum of one PHY may be attached to a management interface via the mechanical interface defined in 22.6.

### 22.1.3 Rates of operation

The MII can support two specific data rates, 10 Mb/s and 100 Mb/s. The functionality is identical at both data rates, as are the signal timing relationships. The only difference between 10 Mb/s and 100 Mb/s operation is the nominal clock frequency.

PHYs that provide an MII are not required to support both data rates, and may support either one or both. PHYs must report the rates they are capable of operating at via the management interface, as described in 22.2.4.

### 22.1.4 Allocation of functions

The allocation of functions at the MII is such that it readily lends itself to implementation in both PHYs and MAC sublayer entities. The division of functions balances the need for media independence with the need for a simple and cost-effective interface.

While the Attachment Unit Interface (AUI) was defined to exist between the Physical Signaling (PLS) and Physical Media Attachment (PMA) sublayers for 10 Mb/s DTEs, the MII maximizes media independence by cleanly separating the Data Link and Physical Layers of the ISO (IEEE) seven-layer reference model. This allocation also recognizes that implementations can benefit from a close coupling of the PLS or PCS sublayer and the PMA sublayer.

### 22.1.5 Relationship of MII and GMII

The Gigabit Media Independent Interface (GMII) is similar to the MII. The GMII uses the MII management interface and register set specified in 22.2.4. These common elements of operation allow Station Management to determine PHY capabilities for any supported speed of operation and configure the station based on those capabilities. In a station supporting both MII and GMII operation, configuration of the station would include enabling either the MII or GMII operation as appropriate for the data rate of the selected PHY.

Most of the MII and GMII signals use the same names, but the width of the RXD and TXD data bundles and the semantics of the associated control signals differ between MII and GMII operation. The GMII transmit path clocking also differs significantly from MII clocking. MII operation of these signals and clocks is specified within Clause 22 and GMII operation is specified within Clause 35.

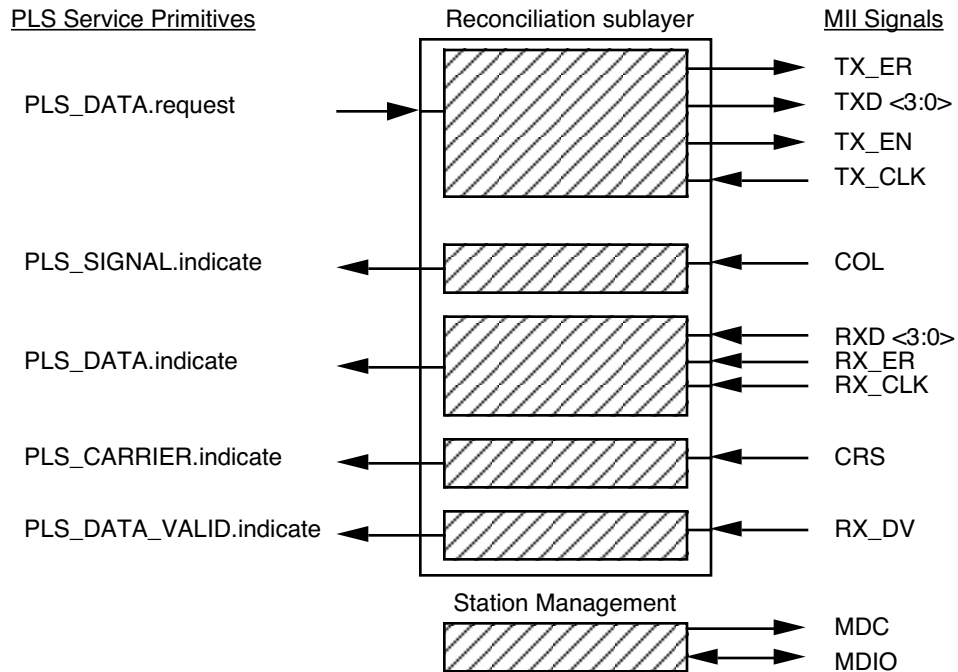
## 22.2 Functional specifications

The MII is designed to make the differences among the various media absolutely transparent to the MAC sublayer. The selection of logical control signals and the functional procedures are all designed to this end. Additionally, the MII is designed to be easily implemented at minimal cost using conventional design techniques and manufacturing processes.

### 22.2.1 Mapping of MII signals to PLS service primitives and Station Management

The Reconciliation sublayer maps the signals provided at the MII to the PLS service primitives defined in Clause 6. The PLS service primitives provided by the Reconciliation sublayer behave in exactly the same manner as defined in Clause 6. The MII signals are defined in detail in 22.2.2.

Figure 22–3 depicts a schematic view of the Reconciliation sublayer inputs and outputs, and demonstrates that the MII management interface is controlled by the Station Management entity (STA).



**Figure 22–3—Reconciliation Sublayer (RS) inputs and outputs, and STA connections to MII**

### 22.2.1.1 Mapping of PLS\_DATA.request

#### 22.2.1.1.1 Function

Map the primitive PLS\_DATA.request to the MII signals TXD<3:0>, TX\_EN and TX\_CLK.

#### 22.2.1.1.2 Semantics of the service primitive

PLS\_DATA.request (OUTPUT\_UNIT)

The OUTPUT\_UNIT parameter can take one of three values: ONE, ZERO, or DATA\_COMPLETE. It represents a single data bit. The values ONE and ZERO are conveyed by the signals TXD<3>, TXD<2>, TXD<1> and TXD<0>, each of which conveys one bit of data while TX\_EN is asserted. The value DATA\_COMPLETE is conveyed by the de-assertion of TX\_EN. Synchronization between the Reconciliation sublayer and the PHY is achieved by way of the TX\_CLK signal.

#### 22.2.1.1.3 When generated

The TX\_CLK signal is generated by the PHY. The TXD<3:0> and TX\_EN signals are generated by the Reconciliation sublayer after every group of four PLS\_DATA.request transactions from the MAC sublayer to request the transmission of four data bits on the physical medium or to stop transmission.

## 22.2.1.2 Mapping of PLS\_DATA.indicate

### 22.2.1.2.1 Function

Map the primitive PLS\_DATA.indicate to the MII signals RXD<3:0>, RX\_DV, RX\_ER, and RX\_CLK.

#### 22.2.1.2.2 Semantics of the service primitive

PLS\_DATA.indicate (INPUT\_UNIT)

The INPUT\_UNIT parameter can take one of two values: ONE or ZERO. It represents a single data bit. The values ONE and ZERO are derived from the signals RXD<3>, RXD<2>, RXD<1>, and RXD<0>, each of which represents one bit of data while RX\_DV is asserted.

The value of the data transferred to the MAC is controlled by the RX\_ER signal, see 22.2.1.5, Response to RX\_ER indication from MII.

Synchronization between the PHY and the Reconciliation sublayer is achieved by way of the RX\_CLK signal.

#### 22.2.1.2.3 When generated

This primitive is generated to all MAC sublayer entities in the network after a PLS\_DATA.request is issued. Each nibble of data transferred on RXD<3:0> will result in the generation of four PLS\_DATA.indicate transactions.

## 22.2.1.3 Mapping of PLS\_CARRIER.indicate

### 22.2.1.3.1 Function

Map the primitive PLS\_CARRIER.indicate to the MII signal CRS.

#### 22.2.1.3.2 Semantics of the service primitive

PLS\_CARRIER.indicate (CARRIER\_STATUS)

The CARRIER\_STATUS parameter can take one of two values: CARRIER\_ON or CARRIER\_OFF. The values CARRIER\_ON and CARRIER\_OFF are derived from the MII signal CRS.

#### 22.2.1.3.3 When generated

The PLS\_CARRIER.indicate service primitive is generated by the Reconciliation sublayer whenever the CARRIER\_STATUS parameter changes from CARRIER\_ON to CARRIER\_OFF or vice versa.

While the RX\_DV signal is de-asserted, any transition of the CRS signal from de-asserted to asserted must cause a transition of CARRIER\_STATUS from the CARRIER\_OFF to the CARRIER\_ON value, and any transition of the CRS signal from asserted to de-asserted must cause a transition of CARRIER\_STATUS from the CARRIER\_ON to the CARRIER\_OFF value. At any time after CRS and RX\_DV are both asserted, de-assertion of RX\_DV must cause CARRIER\_STATUS to transition to the CARRIER\_OFF value. This transition of CARRIER\_STATUS from the CARRIER\_ON to the CARRIER\_OFF value must be recognized by the MAC sublayer, even if the CRS signal is still asserted at the time.

NOTE—The behavior of the CRS signal is specified within this clause so that it can be mapped directly (with the appropriate implementation-specific synchronization) to the carrierSense variable in the MAC process Deference, which is

described in 4.2.8. The behavior of the RX\_DV signal is specified within this clause so that it can be mapped directly to the carrierSense variable in the MAC process BitReceiver, which is described in 4.2.9, provided that the MAC process BitReceiver is implemented to receive a nibble of data on each cycle through the inner loop.

#### **22.2.1.4 Mapping of PLS\_SIGNAL.indicate**

##### **22.2.1.4.1 Function**

Map the primitive PLS\_SIGNAL.indicate to the MII signal COL.

##### **22.2.1.4.2 Semantics of the service primitive**

PLS\_SIGNAL.indicate (SIGNAL\_STATUS)

The SIGNAL\_STATUS parameter can take one of two values: SIGNAL\_ERROR or NO\_SIGNAL\_ERROR. SIGNAL\_STATUS assumes the value SIGNAL\_ERROR when the MII signal COL is asserted, and assumes the value NO\_SIGNAL\_ERROR when COL is de-asserted.

##### **22.2.1.4.3 When generated**

The PLS\_SIGNAL.indicate service primitive is generated whenever SIGNAL\_STATUS makes a transition from SIGNAL\_ERROR to NO\_SIGNAL\_ERROR or vice versa.

##### **22.2.1.5 Response to RX\_ER indication from MII**

If, during frame reception, both RX\_DV and RX\_ER are asserted, the Reconciliation sublayer shall ensure that the MAC will detect a FrameCheckError in that frame.

This requirement may be met by incorporating a function in the Reconciliation sublayer that produces a result that is guaranteed to be not equal to the CRC result, as specified by the algorithm in 3.2.8, of the sequence of nibbles comprising the received frame as delivered to the MAC sublayer. The Reconciliation sublayer must then ensure that the result of this function is delivered to the MAC sublayer at the end of the received frame in place of the last nibble(s) received from the MII.

Other techniques may be employed to respond to RX\_ER, provided that the result is that the MAC sublayer behaves as though a FrameCheckError occurred in the received frame.

##### **22.2.1.6 Conditions for generation of TX\_ER**

If, during the process of transmitting a frame, it is necessary to request that the PHY deliberately corrupt the contents of the frame in such a manner that a receiver will detect the corruption with the highest degree of probability, then the signal TX\_ER may be generated.

For example, a repeater that detects an RX\_ER during frame reception on an input port may propagate that error indication to its output ports by asserting TX\_ER during the process of transmitting that frame.

Since there is no mechanism in the definition of the MAC sublayer by which the transmit data stream can be deliberately corrupted, the Reconciliation sublayer is not required to generate TX\_ER.

#### **22.2.1.7 Mapping of PLS\_DATA\_VALID.indicate**

##### **22.2.1.7.1 Function**

Map the primitive PLS\_DATA\_VALID.indicate to the MII signal RX\_DV.

### 22.2.1.7.2 Semantics of the service primitive

PLS\_DATA\_VALID.indicate (DATA\_VALID\_STATUS)

The DATA\_VALID\_STATUS parameter can take one of two values: DATA\_VALID or DATA\_NOT\_VALID. DATA\_VALID\_STATUS assumes the value DATA\_VALID when the MII signal RX\_DV is asserted, and assumes the value DATA\_NOT\_VALID when RX\_DV is de-asserted.

### 22.2.1.7.3 When generated

The PLS\_DATA\_VALID.indicate service primitive is generated by the Reconciliation sublayer whenever the DATA\_VALID\_STATUS parameter changes from DATA\_VALID to DATA\_NOT\_VALID or vice versa.

## 22.2.2 MII signal functional specifications

### 22.2.2.1 TX\_CLK (transmit clock)

TX\_CLK (Transmit Clock) is a continuous clock that provides the timing reference for the transfer of the TX\_EN, TXD, and TX\_ER signals from the Reconciliation sublayer to the PHY. TX\_CLK is sourced by the PHY.

The TX\_CLK frequency shall be 25% of the nominal transmit data rate  $\pm$  100 ppm. For example, a PHY operating at 100 Mb/s must provide a TX\_CLK frequency of 25 MHz, and a PHY operating at 10 Mb/s must provide a TX\_CLK frequency of 2.5 MHz. The duty cycle of the TX\_CLK signal shall be between 35% and 65% inclusive.

NOTE—See additional information in 22.2.4.1.5.

### 22.2.2.2 RX\_CLK (receive clock)

RX\_CLK is a continuous clock that provides the timing reference for the transfer of the RX\_DV, RXD, and RX\_ER signals from the PHY to the Reconciliation sublayer. RX\_CLK is sourced by the PHY. The PHY may recover the RX\_CLK reference from the received data or it may derive the RX\_CLK reference from a nominal clock (e.g., the TX\_CLK reference).

The minimum high and low times of RX\_CLK shall be 35% of the nominal period under all conditions.

While RX\_DV is asserted, RX\_CLK shall be synchronous with recovered data, shall have a frequency equal to 25% of the data rate of the received signal, and shall have a duty cycle of between 35% and 65% inclusive.

When the signal received from the medium is continuous and the PHY can recover the RX\_CLK reference and supply the RX\_CLK on a continuous basis, there is no need to transition between the recovered clock reference and a nominal clock reference on a frame-by-frame basis. If loss of received signal from the medium causes a PHY to lose the recovered RX\_CLK reference, the PHY shall source the RX\_CLK from a nominal clock reference.

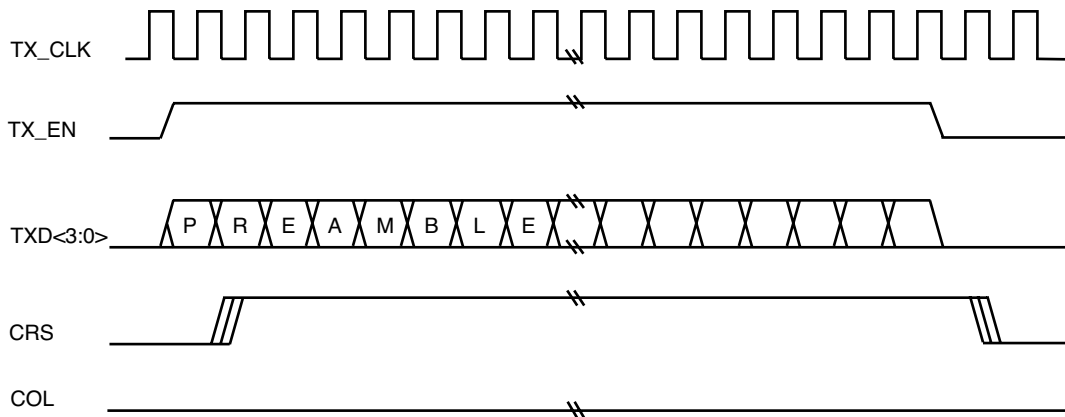
Transitions from nominal clock to recovered clock or from recovered clock to nominal clock shall be made only while RX\_DV is de-asserted. During the interval between the assertion of CRS and the assertion of RX\_DV at the beginning of a frame, the PHY may extend a cycle of RX\_CLK by holding it in either the high or low condition until the PHY has successfully locked onto the recovered clock. Following the de-assertion of RX\_DV at the end of a frame, the PHY may extend a cycle of RX\_CLK by holding it in either the high or low condition for an interval that shall not exceed twice the nominal clock period.

NOTE—This standard neither requires nor assumes a guaranteed phase relationship between the RX\_CLK and TX\_CLK signals. See additional information in 22.2.4.1.5.

**22.2.2.3 TX\_EN (transmit enable)**

TX\_EN indicates that the Reconciliation sublayer is presenting nibbles on the MII for transmission. It shall be asserted by the Reconciliation sublayer synchronously with the first nibble of the preamble and shall remain asserted while all nibbles to be transmitted are presented to the MII. TX\_EN shall be negated prior to the first TX\_CLK following the final nibble of a frame. TX\_EN is driven by the Reconciliation sublayer and shall transition synchronously with respect to the TX\_CLK.

Figure 22–4 depicts TX\_EN behavior during a frame transmission with no collisions.



**Figure 22–4—Transmission with no collision**

**22.2.2.4 TXD (transmit data)**

TXD is a bundle of 4 data signals (TXD<3:0>) that are driven by the Reconciliation sublayer. TXD<3:0> shall transition synchronously with respect to the TX\_CLK. For each TX\_CLK period in which TX\_EN is asserted, TXD<3:0> are accepted for transmission by the PHY. TXD<0 >is the least significant bit. While TX\_EN is de-asserted, TXD<3:0> shall have no effect upon the PHY.

Figure 22–4 depicts TXD<3:0> behavior during the transmission of a frame.

Table 22–1 summarizes the permissible encodings of TXD<3:0>, TX\_EN, and TX\_ER.

**Table 22–1 – Permissible encodings of TXD<3:0>, TX\_EN, and TX\_ER**

TX_EN	TX_ER	TXD<3:0>	Indication
0	0	0000 through 1111	Normal inter-frame
0	1	0000 through 1111	Reserved
1	0	0000 through 1111	Normal data transmission
1	1	0000 through 1111	Transmit error propagation

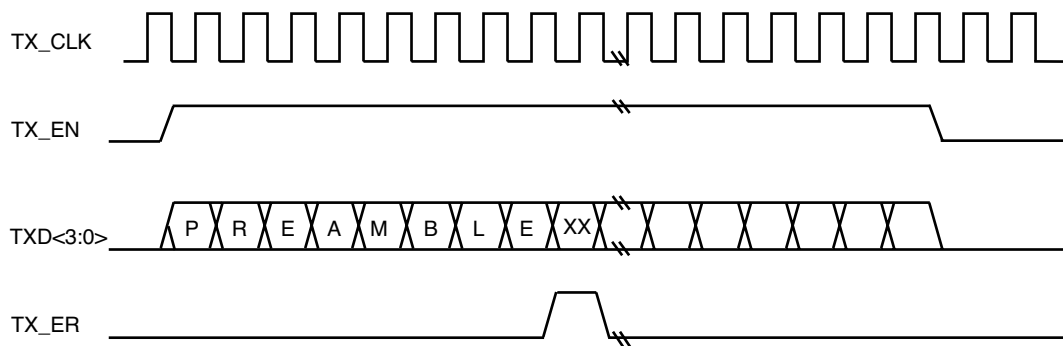


### 22.2.2.5 TX\_ER (transmit coding error)

TX\_ER shall transition synchronously with respect to the TX\_CLK. When TX\_ER is asserted for one or more TX\_CLK periods while TX\_EN is also asserted, the PHY shall emit one or more symbols that are not part of the valid data or delimiter set somewhere in the frame being transmitted. The relative position of the error within the frame need not be preserved.

Assertion of the TX\_ER signal shall not affect the transmission of data when a PHY is operating at 10 Mb/s, or when TX\_EN is de-asserted.

Figure 22–5 shows the behavior of TX\_ER during the transmission of a frame propagating an error.



**Figure 22–5—Propagating an error**

Table 22–1 summarizes the permissible encodings of TXD<3:0>, TX\_EN, and TX\_ER.

The TX\_ER signal shall be implemented at the MII of a PHY, may be implemented at the MII of a repeater that provides an MII port, and may be implemented in MAC sublayer devices. If a Reconciliation sublayer or a repeater with an MII port does not actively drive the TX\_ER signal, it shall ensure that the TX\_ER signal is pulled down to an inactive state at all times.

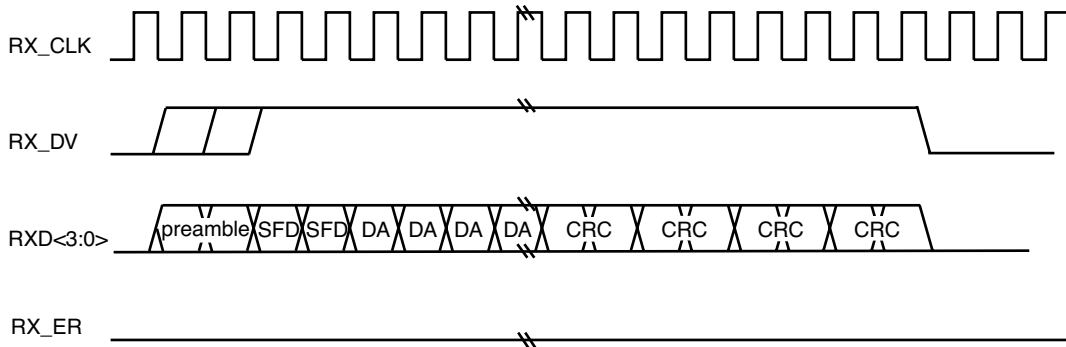
### 22.2.2.6 RX\_DV (Receive Data Valid)

RX\_DV (Receive Data Valid) is driven by the PHY to indicate that the PHY is presenting recovered and decoded nibbles on the RXD<3:0> bundle and that the data on RXD<3:0> is synchronous to RX\_CLK. RX\_DV shall transition synchronously with respect to the RX\_CLK. RX\_DV shall remain asserted continuously from the first recovered nibble of the frame through the final recovered nibble and shall be negated prior to the first RX\_CLK that follows the final nibble. In order for a received frame to be correctly interpreted by the Reconciliation sublayer and the MAC sublayer, RX\_DV must encompass the frame, starting no later than the Start Frame Delimiter (SFD) and excluding any End-of-Frame delimiter.

Figure 22–6 shows the behavior of RX\_DV during frame reception.

### 22.2.2.7 RXD (receive data)

RXD is a bundle of four data signals (RXD<3:0>) that transition synchronously with respect to the RX\_CLK. RXD<3:0> are driven by the PHY. For each RX\_CLK period in which RX\_DV is asserted, RXD<3:0> transfer four bits of recovered data from the PHY to the Reconciliation sublayer. RXD<0> is the



**Figure 22–6—Reception with no errors**

least significant bit. While RX\_DV is de-asserted, RXD<3:0> shall have no effect on the Reconciliation sub-layer.

While RX\_DV is de-asserted, the PHY may provide a False Carrier indication by asserting the RX\_ER signal while driving the value <1110> onto RXD<3:0>. See 22.2.4.4.2 for a description of the conditions under which a PHY will provide a False Carrier indication.

In order for a frame to be correctly interpreted by the MAC sublayer, a completely formed SFD must be passed across the MII. In a DTE operating in half duplex mode, a PHY is not required to loop data transmitted on TXD<3:0> back to RXD<3:0> unless the loopback mode of operation is selected as defined in 22.2.4.1.2. In a DTE operating in full duplex mode, data transmitted on TXD <3:0> must not be looped back to RXD <3:0> unless the loopback mode of operation is selected.

Figure 22–6 shows the behavior of RXD<3:0> during frame reception.

Table 22–2 summarizes the permissible encoding of RXD<3:0>, RX\_ER, and RX\_DV, along with the specific indication provided by each code.

**Table 22–2—Permissible encoding of RXD<3:0>, RX\_ER, and RX\_DV**

RX_DV	RX_ER	RXD<3:0>	Indication
0	0	0000 through 1111	Normal inter-frame
0	1	0000	Normal inter-frame
0	1	0001 through 1101	Reserved
0	1	1110	False Carrier indication
0	1	1111	Reserved
1	0	0000 through 1111	Normal data reception
1	1	0000 through 1111	Data reception with errors

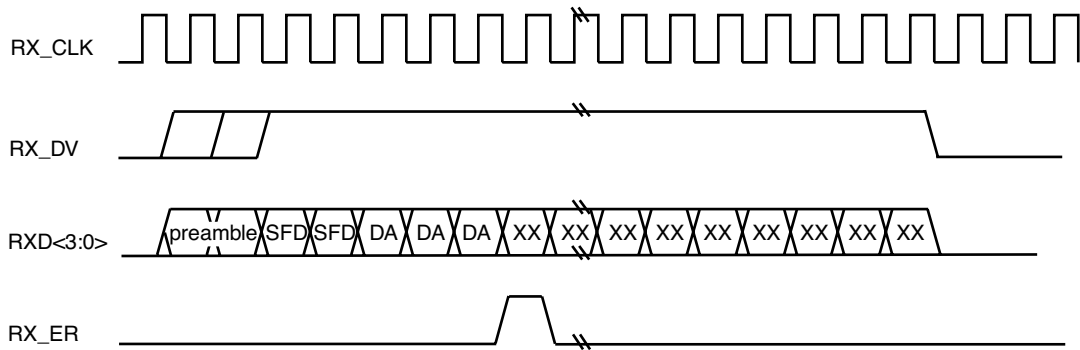
**22.2.2.8 RX\_ER (receive error)**

RX\_ER (Receive Error) is driven by the PHY. RX\_ER shall be asserted for one or more RX\_CLK periods to indicate to the Reconciliation sublayer that an error (e.g., a coding error, or any error that the PHY is capable of detecting, and that may otherwise be undetectable at the MAC sublayer) was detected somewhere in the frame presently being transferred from the PHY to the Reconciliation sublayer. RX\_ER shall transition synchronously with respect to RX\_CLK. While RX\_DV is de-asserted, RX\_ER shall have no effect on the Reconciliation sublayer.

While RX\_DV is de-asserted, the PHY may provide a False Carrier indication by asserting the RX\_ER signal for at least one cycle of the RX\_CLK while driving the appropriate value onto RXD<3:0>, as defined in 22.2.2.7. See 24.2.4.4.2 for a description of the conditions under which a PHY will provide a False Carrier indication.

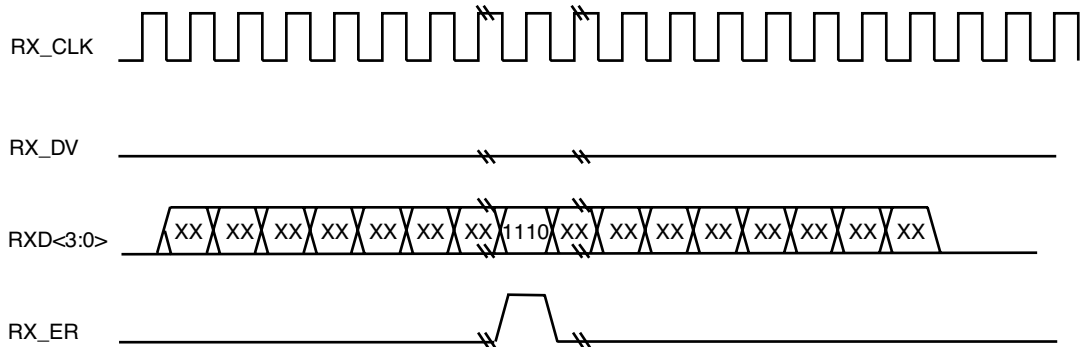
The effect of RX\_ER on the Reconciliation sublayer is defined in 22.2.1.5, Response to RX\_ER indication from MII.

Figure 22–7 shows the behavior of RX\_ER during the reception of a frame with errors.



**Figure 22–7—Reception with errors**

Figure 22–8 shows the behavior of RX\_ER, RX\_DV and RXD<3:0> during a False Carrier indication.



**Figure 22–8—False Carrier indication**

**22.2.2.9 CRS (carrier sense)**

CRS shall be asserted by the PHY when either the transmit or receive medium is nonidle. CRS shall be deasserted by the PHY when both the transmit and receive media are idle. The PHY shall ensure that CRS remains asserted throughout the duration of a collision condition.

CRS is not required to transition synchronously with respect to either the TX\_CLK or the RX\_CLK.

The behavior of the CRS signal is unspecified when the duplex mode bit 0.8 in the control register is set to a logic one, as described in 22.2.4.1.8, or when the Auto-Negotiation process selects a full duplex mode of operation.

Figure 22–4 shows the behavior of CRS during a frame transmission without a collision, while Figure 22–9 shows the behavior of CRS during a frame transmission with a collision.

**22.2.2.10 COL (collision detected)**

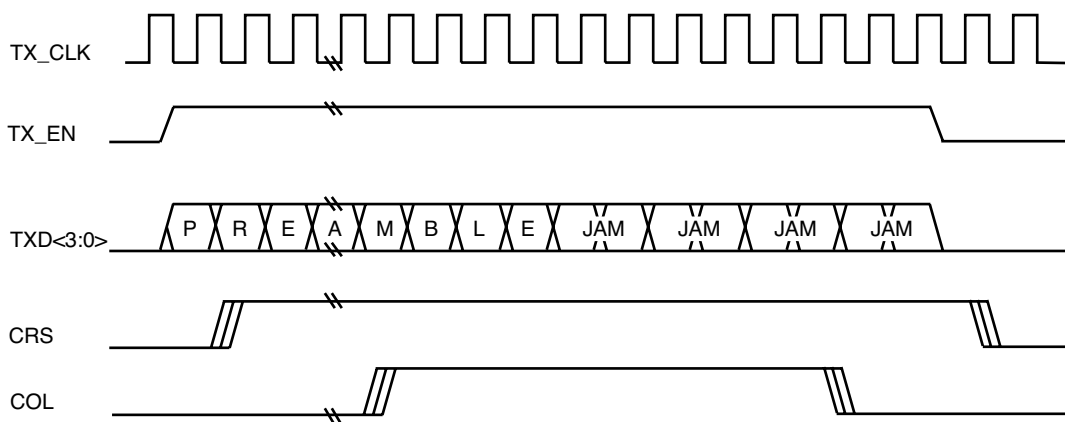
COL shall be asserted by the PHY upon detection of a collision on the medium, and shall remain asserted while the collision condition persists.

COL shall be asserted by a PHY that is operating at 10 Mb/s in response to a *signal\_quality\_error* message from the PMA.

COL is not required to transition synchronously with respect to either the TX\_CLK or the RX\_CLK.

The behavior of the COL signal is unspecified when the duplex mode bit 0.8 in the control register is set to a logic one, as described in 22.2.4.1.8, or when the Auto-Negotiation process selects a full duplex mode of operation.

Figure 22–9 shows the behavior of COL during a frame transmission with a collision.



**Figure 22–9—Transmission with collision**

NOTE—The circuit assembly that contains the Reconciliation sublayer may incorporate a weak pull-up on the COL signal as a means of detecting an open circuit condition on the COL signal at the MII. The limit on the value of this pull-up is defined in 22.4.4.2.

### 22.2.2.11 MDC (management data clock)

MDC is sourced by the Station Management entity to the PHY as the timing reference for transfer of information on the MDIO signal. MDC is an aperiodic signal that has no maximum high or low times. The minimum high and low times for MDC shall be 160 ns each, and the minimum period for MDC shall be 400 ns, regardless of the nominal period of TX\_CLK and RX\_CLK.

### 22.2.2.12 MDIO (management data input/output)

MDIO is a bidirectional signal between the PHY and the STA. It is used to transfer control information and status between the PHY and the STA. Control information is driven by the STA synchronously with respect to MDC and is sampled synchronously by the PHY. Status information is driven by the PHY synchronously with respect to MDC and is sampled synchronously by the STA.

MDIO shall be driven through three-state circuits that enable either the STA or the PHY to drive the signal. A PHY that is attached to the MII via the mechanical interface specified in 22.6 shall provide a resistive pull-up to maintain the signal in a high state. The STA shall incorporate a resistive pull-down on the MDIO signal and thus may use the quiescent state of MDIO to determine if a PHY is connected to the MII via the mechanical interface defined in 22.6. The limits on the values of these pull-ups and pull-downs are defined in 22.4.4.2.

### 22.2.3 Frame structure

Data frames transmitted through the MII shall have the frame format shown in Figure 22–10.

<inter-frame><preamble><sfd><data><efd>

Figure 22–10—MII frame format

For the MII, transmission and reception of each octet of data shall be done a nibble at a time with the order of nibble transmission and reception as shown in Figure 22–11.

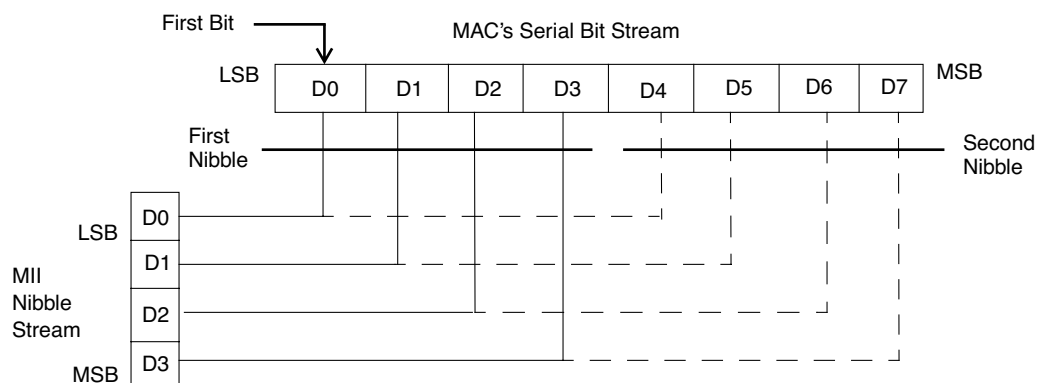


Figure 22–11—Octet/nibble transmit and receive order

The bits of each octet are transmitted and received as two nibbles, bits 0 through 3 of the octet corresponding to bits 0 through 3 of the first nibble transmitted or received, and bits 4 through 7 of the octet corresponding to bits 0 through 3 of the second nibble transmitted or received.

### 22.2.3.1 Inter-frame

The inter-frame period provides an observation window for an unspecified amount of time during which no data activity occurs on the MII. The absence of data activity is indicated by the de-assertion of the RX\_DV signal on the receive path, and the de-assertion of the TX\_EN signal on the transmit path. The MAC inter-FrameSpacing parameter defined in Clause 4 is measured from the de-assertion of the CRS signal to the assertion of the CRS signal.

### 22.2.3.2 Preamble and start of frame delimiter

#### 22.2.3.2.1 Transmit case

The preamble <preamble> begins a frame transmission. The bit value of the preamble field at the MII is unchanged from that specified in 7.2.3.2 and shall consist of 7 octets with the following bit values:

10101010 10101010 10101010 10101010 10101010 10101010 10101010

In the preceding example, the preamble is displayed using the bit order it would have if transmitted serially. This means that for each octet the leftmost 1 value represents the LSB of the octet, and the rightmost 0 value the octet MSB.

The SFD (Start Frame Delimiter) <sfd> indicates the start of a frame and follows the preamble. The bit value of the SFD at the MII is unchanged from that specified in 7.2.3.3 and is the bit sequence:

10101011

The preamble and SFD shall be transmitted through the MII as nibbles starting from the assertion of TX\_EN as shown in Table 22–3.

**Table 22–3—Transmitted preamble and SFD**

Signal	Bit values of nibbles transmitted through MII																				
TXD0	X	1 <sup>a</sup>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1 <sup>b</sup>	1	D0 <sup>c</sup>	D4 <sup>d</sup>	
TXD1	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	D1	D5	
TXD2	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	D2	D6	
TXD3	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	D3	D7
TX_EN	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

<sup>a</sup>1st preamble nibble transmitted.

<sup>b</sup>1st SFD nibble transmitted.

<sup>c</sup>1st data nibble transmitted.

<sup>d</sup>D0 through D7 are the first eight bits of the data field from the Protocol Data Unit (PDU).

#### 22.2.3.2.2 Receive case

The conditions for assertion of RX\_DV are defined in 22.2.2.6.

The alignment of the received SFD and data at the MII shall be as shown in Table 22–4 and Table 22–5. Table 22–4 depicts the case where no preamble nibbles are conveyed across the MII, and Table 22–5 depicts the case where the entire preamble is conveyed across the MII.

**Table 22–4—Start of receive with no preamble preceding SFD**

Signal	Bit values of nibbles received through MII											
RXD0	X	X	X	X	X	X	X	X	1 <sup>a</sup>	1	D0 <sup>b</sup>	D4 <sup>c</sup>
RXD1	X	X	X	X	X	X	X	X	0	0	D1	D5
RXD2	X	X	X	X	X	X	X	X	1	1	D2	D6
RXD3	X	X	X	X	X	X	X	X	0	1	D3	D7
RX_DV	0	0	0	0	0	0	0	0	1	1	1	1

<sup>a</sup>1st SFD nibble received.<sup>b</sup>1st data nibble received.<sup>c</sup>D0 through D7 are the first eight bits of the data field from the PDU.**Table 22–5—Start of receive with entire preamble preceding SFD**

Signal	Bit values of nibbles received through MII																				
RXD0	X	1 <sup>a</sup>	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1 <sup>b</sup>	1	D0 <sup>c</sup>	D4 <sup>d</sup>
RXD1	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	D1	D5
RXD2	X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	D2	D6
RXD3	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	D3	D7
RX_DV	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

<sup>a</sup>1st preamble nibble received.<sup>b</sup>1st SFD nibble received.<sup>c</sup>1st data nibble received.<sup>d</sup>D0 through D7 are the first eight bits of the data field from the PDU.

### 22.2.3.3 Data

The data in a well formed frame shall consist of N octets of data transmitted as 2N nibbles. For each octet of data the transmit order of each nibble is as specified in Figure 22–11. Data in a collision fragment may consist of an odd number of nibbles.

### 22.2.3.4 End-of-Frame delimiter (EFD)

De-assertion of the TX\_EN signal constitutes an End-of-Frame delimiter for data conveyed on TXD<3:0>, and de-assertion of RX\_DV constitutes an End-of-Frame delimiter for data conveyed on RXD<3:0>.

### 22.2.3.5 Handling of excess nibbles

An excess nibble condition occurs when an odd number of nibbles is conveyed across the MII beginning with the SFD and including all nibbles conveyed until the End-of-Frame delimiter. Reception of a frame containing a non-integer number of octets shall be indicated by the PHY as an excess nibble condition.

Transmission of an excess nibble may be handled by the PHY in an implementation-specific manner. No assumption should be made with regard to truncation, octet padding, or exact nibble transmission by the PHY.

### 22.2.4 Management functions

The management interface specified here provides a simple, two-wire, serial interface to connect a management entity and a managed PHY for the purposes of controlling the PHY and gathering status from the PHY. This interface is referred to as the MII Management Interface.

The management interface consists of a pair of signals that physically transport the management information across the MII or GMII, a frame format and a protocol specification for exchanging management frames, and a register set that can be read and written using these frames. The register definition specifies a basic register set with an extension mechanism. The MII uses two basic registers. The GMII also uses the same two basic registers and adds a third basic register.

The MII basic register set consists of two registers referred to as the Control register (Register 0) and the Status register (Register 1). All PHYs that provide an MII shall incorporate the basic register set. All PHYs that provide a GMII shall incorporate an extended basic register set consisting of the Control register (Register 0), Status register (Register 1), and Extended Status register (Register 15). The status and control functions defined here are considered basic and fundamental to 100 Mb/s and 1000 Mb/s PHYs. Registers 2 through 10 are part of the extended register set. The format of Registers 4 through 10 are defined for the specific Auto-Negotiation protocol used (Clause 28 or Clause 37). The format of these registers is selected by the bit settings of Registers 1 and 15.

The full set of management registers is listed in Table 22–6.

**Table 22–6—MII management register set**

Register address	Register name	Basic/Extended	
		MII	GMII
0	Control	B	B
1	Status	B	B
2,3	PHY Identifier	E	E
4	Auto-Negotiation Advertisement	E	E
5	Auto-Negotiation Link Partner Base Page Ability	E	E
6	Auto-Negotiation Expansion	E	E
7	Auto-Negotiation Next Page Transmit	E	E
8	Auto-Negotiation Link Partner Received Next Page	E	E
9	MASTER-SLAVE Control Register	E	E
10	MASTER-SLAVE Status Register	E	E
11 through 14	Reserved	E	E
15	Extended Status	Reserved	B
16 through 31	Vendor Specific	E	E



### 22.2.4.1 Control register (Register 0)

The assignment of bits in the Control Register is shown in Table 22–7 below. The default value for each bit of the Control Register should be chosen so that the initial state of the PHY upon power up or reset is a normal operational state without management intervention.

**Table 22–7—Control register bit definitions**

Bit(s)	Name	Description	R/W <sup>a</sup>															
0.15	Reset	1 = PHY reset 0 = normal operation	R/W SC															
0.14	Loopback	1 = enable loopback mode 0 = disable loopback mode	R/W															
0.13	Speed Selection (LSB)	<table style="margin-left: 20px;"> <tr> <td>0.6</td> <td>0.13</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td>= Reserved</td> </tr> <tr> <td>1</td> <td>0</td> <td>= 1000 Mb/s</td> </tr> <tr> <td>0</td> <td>1</td> <td>= 100 Mb/s</td> </tr> <tr> <td>0</td> <td>0</td> <td>= 10 Mb/s</td> </tr> </table>	0.6	0.13		1	1	= Reserved	1	0	= 1000 Mb/s	0	1	= 100 Mb/s	0	0	= 10 Mb/s	R/W
0.6	0.13																	
1	1	= Reserved																
1	0	= 1000 Mb/s																
0	1	= 100 Mb/s																
0	0	= 10 Mb/s																
0.12	Auto-Negotiation Enable	1 = Enable Auto-Negotiation Process 0 = Disable Auto-Negotiation Process	R/W															
0.11	Power Down	1 = power down 0 = normal operation <sup>b</sup>	R/W															
0.10	Isolate	1 = electrically Isolate PHY from MII or GMII 0 = normal operation <sup>b</sup>	R/W															
0.9	Restart Auto-Negotiation	1 = Restart Auto-Negotiation Process 0 = normal operation	R/W SC															
0.8	Duplex Mode	1 = Full Duplex 0 = Half Duplex	R/W															
0.7	Collision Test	1 = enable COL signal test 0 = disable COL signal test	R/W															
0.6	Speed Selection (MSB)	<table style="margin-left: 20px;"> <tr> <td>0.6</td> <td>0.13</td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td>= Reserved</td> </tr> <tr> <td>1</td> <td>0</td> <td>= 1000 Mb/s</td> </tr> <tr> <td>0</td> <td>1</td> <td>= 100 Mb/s</td> </tr> <tr> <td>0</td> <td>0</td> <td>= 10 Mb/s</td> </tr> </table>	0.6	0.13		1	1	= Reserved	1	0	= 1000 Mb/s	0	1	= 100 Mb/s	0	0	= 10 Mb/s	R/W
0.6	0.13																	
1	1	= Reserved																
1	0	= 1000 Mb/s																
0	1	= 100 Mb/s																
0	0	= 10 Mb/s																
0.5:0	Reserved	Write as 0, ignore on Read	R/W															

<sup>a</sup>R/W = Read/Write, SC = Self-Clearing.

<sup>b</sup>For normal operation, both 0.10 and 0.11 must be cleared to zero; see 22.2.4.1.5.

#### 22.2.4.1.1 Reset

Resetting a PHY is accomplished by setting bit 0.15 to a logic one. This action shall set the status and control registers to their default states. As a consequence this action may change the internal state of the PHY and the state of the physical link associated with the PHY. This bit is self-clearing, and a PHY shall return a value of one in bit 0.15 until the reset process is completed. A PHY is not required to accept a write transaction to the control register until the reset process is completed, and writes to bits of the control register other than 0.15 may have no effect until the reset process is completed. The reset process shall be completed within 0.5 s from the setting of bit 0.15.

The default value of bit 0.15 is zero.

NOTE—This operation may interrupt data communication.

#### **22.2.4.1.2 Loopback**

The PHY shall be placed in a loopback mode of operation when bit 0.14 is set to a logic one. When bit 0.14 is set, the PHY receive circuitry shall be isolated from the network medium, and the assertion of TX\_EN at the MII or GMII shall not result in the transmission of data on the network medium. When bit 0.14 is set, the PHY shall accept data from the MII or GMII transmit data path and return it to the MII or GMII receive data path in response to the assertion of TX\_EN. When bit 0.14 is set, the delay from the assertion of TX\_EN to the assertion of RX\_DV shall be less than 512 BT. When bit 0.14 is set, the COL signal shall remain deasserted at all times, unless bit 0.7 is set, in which case the COL signal shall behave as described in 22.2.4.1.9. Clearing bit 0.14 to zero allows normal operation.

The default value of bit 0.14 is zero.

NOTE—The signal path through the PHY that is exercised in the loopback mode of operation is implementation specific, but it is recommended that the signal path encompass as much of the PHY circuitry as is practical. The intention of providing this loopback mode of operation is to permit a diagnostic or self-test function to perform the transmission and reception of a PDU, thus testing the transmit and receive data paths. Other loopback signal paths through a PHY may be enabled via the extended register set, in an implementation-specific fashion.

#### **22.2.4.1.3 Speed selection**

Link speed can be selected via either the Auto-Negotiation process, or manual speed selection. Manual speed selection is allowed when Auto-Negotiation is disabled by clearing bit 0.12 to zero. When Auto-Negotiation is disabled and bit 0.6 is cleared to a logic zero, setting bit 0.13 to a logic one configures the PHY for 100 Mb/s operation, and clearing bit 0.13 to a logic zero configures the PHY for 10 Mb/s operation. When Auto-Negotiation is disabled and bit 0.6 is set to a logic one, clearing bit 0.13 to a logic zero selects 1000 Mb/s operation. The combination of both bits 0.6 and 0.13 set to a logic one is reserved for future standardization. When Auto-Negotiation is enabled, bits 0.6 and 0.13 can be read or written, but the state of bits 0.6 and 0.13 have no effect on the link configuration, and it is not necessary for bits 0.6 and 0.13 to reflect the operating speed of the link when it is read. If a PHY reports via bits 1.15:9 and bits 15.15:12 that it is not able to operate at all speeds, the value of bits 0.6 and 0.13 shall correspond to a speed at which the PHY can operate, and any attempt to change the bits to an invalid setting shall be ignored.

The default value of bits 0.6 and 0.13 are the encoding of the highest data rate at which the PHY can operate as indicated by bits 1.15:9 and 15.15:12.

#### **22.2.4.1.4 Auto-Negotiation enable**

The Auto-Negotiation process shall be enabled by setting bit 0.12 to a logic one. If bit 0.12 is set to a logic one, then bits 0.13, 0.8, and 0.6 shall have no effect on the link configuration, and station operation other than that specified by the Auto-Negotiation protocol. If bit 0.12 is cleared to a logic zero, then bits 0.13, 0.8, and 0.6 will determine the link configuration, regardless of the prior state of the link configuration and the Auto-Negotiation process.

If a PHY reports via bit 1.3 that it lacks the ability to perform Auto-Negotiation, the PHY shall return a value of zero in bit 0.12. If a PHY reports via bit 1.3 that it lacks the ability to perform Auto-Negotiation, bit 0.12 should always be written as zero, and any attempt to write a one to bit 0.12 shall be ignored.

The default value of bit 0.12 is one, unless the PHY reports via bit 1.3 that it lacks the ability to perform Auto-Negotiation, in which case the default value of bit 0.12 is zero.

### 22.2.4.1.5 Power down

The PHY may be placed in a low-power consumption state by setting bit 0.11 to a logic one. Clearing bit 0.11 to zero allows normal operation. The specific behavior of a PHY in the power-down state is implementation specific. While in the power-down state, the PHY shall respond to management transactions. During the transition to the power-down state and while in the power-down state, the PHY shall not generate spurious signals on the MII or GMII.

A PHY is not required to meet the RX\_CLK and TX\_CLK signal functional requirements when either bit 0.11 or bit 0.10 is set to a logic one. A PHY shall meet the RX\_CLK and TX\_CLK signal functional requirements defined in 22.2.2 within 0.5 s after both bit 0.11 and 0.10 are cleared to zero.

The default value of bit 0.11 is zero.

### 22.2.4.1.6 Isolate

The PHY may be forced to electrically isolate its data paths from the MII or GMII by setting bit 0.10 to a logic one. Clearing bit 0.10 allows normal operation. When the PHY is isolated from the MII or GMII it shall not respond to the TXD data bundle, TX\_EN, TX\_ER and GTX\_CLK inputs, and it shall present a high impedance on its TX\_CLK, RX\_CLK, RX\_DV, RX\_ER, RXD data bundle, COL, and CRS outputs. When the PHY is isolated from the MII or GMII it shall respond to management transactions.

A PHY that is connected to the MII via the mechanical interface defined in 22.6 shall have a default value of one for bit 0.10 so as to avoid the possibility of having multiple MII output drivers actively driving the same signal path simultaneously.

NOTE—This clause neither requires nor assumes any specific behavior at the MDI resulting from setting bit 0.10 to a logic one.

### 22.2.4.1.7 Restart Auto-Negotiation

If a PHY reports via bit 1.3 that it lacks the ability to perform Auto-Negotiation, or if Auto-Negotiation is disabled, the PHY shall return a value of zero in bit 0.9. If a PHY reports via bit 1.3 that it lacks the ability to perform Auto-Negotiation, or if Auto-Negotiation is disabled, bit 0.9 should always be written as zero, and any attempt to write a one to bit 0.9 shall be ignored.

Otherwise, the Auto-Negotiation process shall be restarted by setting bit 0.9 to a logic one. This bit is self-clearing, and a PHY shall return a value of one in bit 0.9 until the Auto-Negotiation process has been initiated. The Auto-Negotiation process shall not be affected by writing a zero to bit 0.9.

The default value of bit 0.9 is zero.

### 22.2.4.1.8 Duplex mode

The duplex mode can be selected via either the Auto-Negotiation process, or manual duplex selection. Manual duplex selection is allowed when Auto-Negotiation is disabled by clearing bit 0.12 to zero. When Auto-Negotiation is disabled, setting bit 0.8 to a logic one configures the PHY for full duplex operation, and clearing bit 0.8 to a logic zero configures the PHY for half duplex operation. When Auto-Negotiation is enabled, bit 0.8 can be read or written, but the state of bit 0.8 has no effect on the link configuration. If a PHY reports via bits 1.15:9 and 15.15:12 that it is able to operate in only one duplex mode, the value of bit 0.8 shall correspond to the mode in which the PHY can operate, and any attempt to change the setting of bit 0.8 shall be ignored.

When a PHY is placed in the loopback mode of operation via bit 0.14, the behavior of the PHY shall not be affected by the state of bit 0.8.

The default value of bit 0.8 is zero, unless a PHY reports via bits 1.15:9 and 15.15:12 that it is able to operate only in full duplex mode, in which case the default value of bit 0.8 is one.

#### **22.2.4.1.9 Collision test**

The COL signal at the MII or GMII may be tested by setting bit 0.7 to a logic one. When bit 0.7 is set to one, the PHY shall assert the COL signal within 512 BT in response to the assertion of TX\_EN. While bit 0.7 is set to one, the PHY shall de-assert the COL signal within 4 BT when connected to an MII, or 16 BT when connected to a GMII, in response to the de-assertion of TX\_EN. Clearing bit 0.7 to zero allows normal operation.

The default value of bit 0.7 is zero.

NOTE—It is recommended that the Collision Test function be used only in conjunction with the loopback mode of operation defined in 22.2.4.1.2.

#### **22.2.4.1.10 Speed selection**

Bit 0.6 is used in conjunction with bits 0.13 and 0.12 to select the speed of operation as described in 22.2.4.1.3.

#### **22.2.4.1.11 Reserved bits**

Bits 0.5:0 are reserved for future standardization. They shall be written as zero and shall be ignored when read; however, a PHY shall return the value zero in these bits.

### **22.2.4.2 Status register (Register 1)**

The assignment of bits in the Status register is shown in Table 22–8. All of the bits in the Status register are read only, a write to the Status register shall have no effect.

#### **22.2.4.2.1 100BASE-T4 ability**

When read as a logic one, bit 1.15 indicates that the PHY has the ability to perform link transmission and reception using the 100BASE-T4 signaling specification. When read as a logic zero, bit 1.15 indicates that the PHY lacks the ability to perform link transmission and reception using the 100BASE-T4 signaling specification.

#### **22.2.4.2.2 100BASE-X full duplex ability**

When read as a logic one, bit 1.14 indicates that the PHY has the ability to perform full duplex link transmission and reception using the 100BASE-X signaling specification. When read as a logic zero, bit 1.14 indicates that the PHY lacks the ability to perform full duplex link transmission and reception using the 100BASE-X signaling specification.

#### **22.2.4.2.3 100BASE-X half duplex ability**

When read as a logic one, bit 1.13 indicates that the PHY has the ability to perform half duplex link transmission and reception using the 100BASE-X signaling specification. When read as a logic zero, bit 1.13 indicates that the PHY lacks the ability to perform half duplex link transmission and reception using the 100BASE-X signaling specification.

**Table 22–8—Status register bit definitions**

Bit(s)	Name	Description	R/W <sup>a</sup>
1.15	100BASE-T4	1 = PHY able to perform 100BASE-T4 0 = PHY not able to perform 100BASE-T4	RO
1.14	100BASE-X Full Duplex	1 = PHY able to perform full duplex 100BASE-X 0 = PHY not able to perform full duplex 100BASE-X	RO
1.13	100BASE-X Half Duplex	1 = PHY able to perform half duplex 100BASE-X 0 = PHY not able to perform half duplex 100BASE-X	RO
1.12	10 Mb/s Full Duplex	1 = PHY able to operate at 10 Mb/s in full duplex mode 0 = PHY not able to operate at 10 Mb/s in full duplex mode	RO
1.11	10 Mb/s Half Duplex	1 = PHY able to operate at 10 Mb/s in half duplex mode 0 = PHY not able to operate at 10 Mb/s in half duplex mode	RO
1.10	100BASE-T2 Full Duplex	1 = PHY able to perform full duplex 100BASE-T2 0 = PHY not able to perform full duplex 100BASE-T2	RO
1.9	100BASE-T2 Half Duplex	1 = PHY able to perform half duplex 100BASE-T2 0 = PHY not able to perform half duplex 100BASE-T2	RO
1.8	Extended Status	1 = Extended status information in Register 15 0 = No extended status information in Register 15	RO
1.7	Reserved	ignore when read	RO
1.6	MF Preamble Suppression	1 = PHY will accept management frames with preamble suppressed. 0 = PHY will not accept management frames with preamble suppressed.	RO
1.5	Auto-Negotiation Complete	1 = Auto-Negotiation process completed 0 = Auto-Negotiation process not completed	RO
1.4	Remote Fault	1 = remote fault condition detected 0 = no remote fault condition detected	RO/ LH
1.3	Auto-Negotiation Ability	1 = PHY is able to perform Auto-Negotiation 0 = PHY is not able to perform Auto-Negotiation	RO
1.2	Link Status	1 = link is up 0 = link is down	RO/ LL
1.1	Jabber Detect	1 = jabber condition detected 0 = no jabber condition detected	RO/ LH
1.0	Extended Capability	1 = extended register capabilities 0 = basic register set capabilities only	RO

<sup>a</sup>RO = Read Only, LL = Latching Low, LH = Latching High

#### 22.2.4.2.4 10 Mb/s full duplex ability

When read as a logic one, bit 1.12 indicates that the PHY has the ability to perform full duplex link transmission and reception while operating at 10 Mb/s. When read as a logic zero, bit 1.12 indicates that the PHY lacks the ability to perform full duplex link transmission and reception while operating at 10 Mb/s.

#### **22.2.4.2.5 10 Mb/s half duplex ability**

When read as a logic one, bit 1.11 indicates that the PHY has the ability to perform half duplex link transmission and reception while operating at 10 Mb/s. When read as a logic zero, bit 1.11 indicates that the PHY lacks the ability to perform half duplex link transmission and reception while operating at 10 Mb/s.

#### **22.2.4.2.6 100BASE-T2 full duplex ability**

When read as a logic one, bit 1.10 indicates that the PHY has the ability to perform full duplex link transmission and reception using the 100BASE-T2 signaling specification. When read as a logic zero, bit 1.10 indicates that the PHY lacks the ability to perform full duplex link transmission and reception using the 100BASE-T2 signaling specification.

#### **22.2.4.2.7 100BASE-T2 half duplex ability**

When read as a logic one, bit 1.9 indicates that the PHY has the ability to perform half duplex link transmission and reception using the 100BASE-T2 signaling specification. When read as a logic zero, bit 1.9 indicates that the PHY lacks the ability to perform half duplex link transmission and reception using the 100BASE-T2 signaling specification.

#### **22.2.4.2.8 Reserved bits**

Bit 1:7 is reserved for future standardization, shall be written as zero, and shall be ignored when read; however, a PHY shall return the value zero in this bit.

#### **22.2.4.2.9 MF preamble suppression ability**

When read as a logic one, bit 1.6 indicates that the PHY is able to accept management frames regardless of whether they are or are not preceded by the preamble pattern described in 22.2.4.5.2. When read as a logic zero, bit 1.6 indicates that the PHY is not able to accept management frames unless they are preceded by the preamble pattern described in 22.2.4.5.2.

#### **22.2.4.2.10 Auto-Negotiation complete**

When read as a logic one, bit 1.5 indicates that the Auto-Negotiation process has been completed, and that the contents of the extended registers implemented by the Auto-Negotiation protocol (either Clause 28 or Clause 37) are valid. When read as a logic zero, bit 1.5 indicates that the Auto-Negotiation process has not been completed, and that the contents of the extended registers are as defined by the current state of the Auto-Negotiation protocol, or as written for manual configuration. A PHY shall return a value of zero in bit 1.5 if Auto-Negotiation is disabled by clearing bit 0.12. A PHY shall also return a value of zero in bit 1.5 if it lacks the ability to perform Auto-Negotiation.

#### **22.2.4.2.11 Remote fault**

When read as a logic one, bit 1.4 indicates that a remote fault condition has been detected. The type of fault as well as the criteria and method of fault detection is PHY specific. The Remote Fault bit shall be implemented with a latching function, such that the occurrence of a remote fault will cause the Remote Fault bit to become set and remain set until it is cleared. The Remote Fault bit shall be cleared each time Register 1 is read via the management interface, and shall also be cleared by a PHY reset.

If a PHY has no provision for remote fault detection, it shall maintain bit 1.4 in a cleared state. Further information regarding the remote fault indication can be found in 37.2.1.5, 22.2.1.2, and in 24.3.2.1.

#### **22.2.4.2.12 Auto-Negotiation ability**

When read as a logic one, bit 1.3 indicates that the PHY has the ability to perform Auto-Negotiation. When read as a logic zero, bit 1.3 indicates that the PHY lacks the ability to perform Auto-Negotiation.

#### **22.2.4.2.13 Link Status**

When read as a logic one, bit 1.2 indicates that the PHY has determined that a valid link has been established. When read as a logic zero, bit 1.2 indicates that the link is not valid. The criteria for determining link validity is PHY specific. The Link Status bit shall be implemented with a latching function, such that the occurrence of a link failure condition will cause the Link Status bit to become cleared and remain cleared until it is read via the management interface. This status indication is intended to support the management attribute defined in 30.5.1.1.4, aMediaAvailable.

#### **22.2.4.2.14 Jabber detect**

When read as a logic one, bit 1.1 indicates that a jabber condition has been detected. This status indication is intended to support the management attribute defined in 30.5.1.1.6, aJabber, and the MAU notification defined in 30.5.1.3.1, nJabber. The criteria for the detection of a jabber condition is PHY specific. The Jabber Detect bit shall be implemented with a latching function, such that the occurrence of a jabber condition will cause the Jabber Detect bit to become set and remain set until it is cleared. The Jabber Detect bit shall be cleared each time Register 1 is read via the management interface, and shall also be cleared by a PHY reset.

PHYs specified for 100 Mb/s operation or above do not incorporate a Jabber Detect function, as this function is defined to be performed in the repeater unit at these speeds. Therefore, PHYs specified for 100 Mb/s operation and above shall always return a value of zero in bit 1.1.

#### **22.2.4.2.15 Extended capability**

When read as a logic one, bit 1.0 indicates that the PHY provides an extended set of capabilities which may be accessed through the extended register set. When read as a logic zero, bit 1.0 indicates that the PHY provides only the basic register set.

#### **22.2.4.2.16 Extended status**

When read as a logic one, bit 1.8 indicates that the base register status information is extended into Register 15. All PHYs supporting 1000 Mb/s operation shall have this bit set to a logic one. When read as a logic zero, bit 1.8 indicates that the extended status is not implemented and that the PHY lacks the ability to perform transmission and reception at 1000 Mb/s.

### **22.2.4.3 Extended capability registers**

In addition to the basic register set defined in 22.2.4.1 and 22.2.4.2, PHYs may provide an extended set of capabilities that may be accessed and controlled via the MII management interface. Nine registers have been defined within the extended address space for the purpose of providing a PHY-specific identifier to layer management, and to provide control and monitoring for the Auto-Negotiation process.

If an attempt is made to perform a read transaction to a register in the extended register set, and the PHY being read does not implement the addressed register, the PHY shall not drive the MDIO line in response to the read transaction. If an attempt is made to perform a write transaction to a register in the extended register set, and the PHY being written does not implement the addressed register, the write transaction shall be ignored by the PHY.

**22.2.4.3.1 PHY Identifier (Registers 2 and 3)**

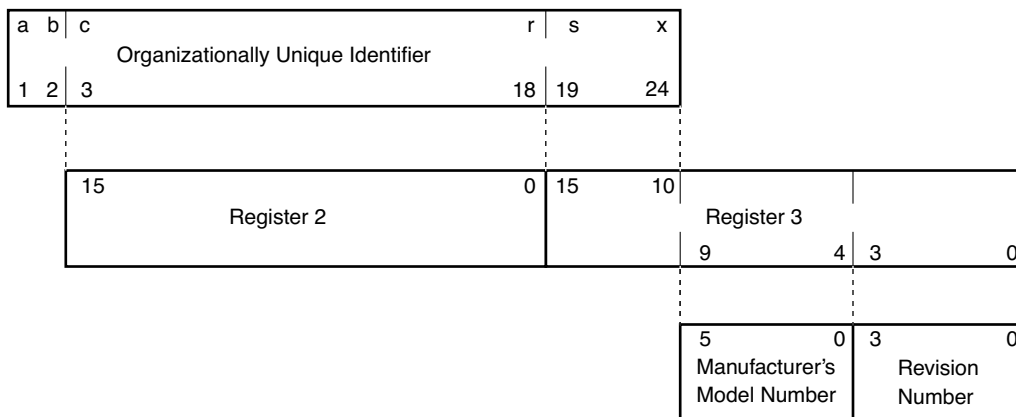
Registers 2 and 3 provide a 32-bit value, which shall constitute a unique identifier for a particular type of PHY. A PHY may return a value of zero in each of the 32 bits of the PHY Identifier.

Bit 2.15 shall be the MSB of the PHY Identifier, and bit 3.0 shall be the LSB of the PHY Identifier.

The PHY Identifier shall be composed of the third through 24th bits of the Organizationally Unique Identifier (OUI) assigned to the PHY manufacturer by the IEEE,<sup>1</sup> plus a six-bit manufacturer’s model number, plus a four-bit manufacturer’s revision number. The PHY Identifier is intended to provide sufficient information to support the oResourceTypeID object as required in 30.1.2.

The third bit of the OUI is assigned to bit 2.15, the fourth bit of the OUI is assigned to bit 2.14, and so on. Bit 2.0 contains the eighteenth bit of the OUI. Bit 3.15 contains the nineteenth bit of the OUI, and bit 3.10 contains the twenty-fourth bit of the OUI. Bit 3.9 contains the MSB of the manufacturer’s model number. Bit 3.4 contains the LSB of the manufacturer’s model number. Bit 3.3 contains the MSB of the manufacturer’s revision number, and bit 3.0 contains the LSB of the manufacturer’s revision number.

Figure 22–12 depicts the mapping of this information to the bits of Registers 2 and 3. Additional detail describing the format of OUIs can be found in IEEE Std 802-1990™.



**Figure 22–12—Format of PHY Identifier**

**22.2.4.3.2 Auto-Negotiation advertisement (Register 4)**

Register 4 provides 16 bits that are used by the Auto-Negotiation process. See 28.2.4.1 and 37.2.5.1.

**22.2.4.3.3 Auto-Negotiation link partner ability (Register 5)**

Register 5 provides 16 bits that are used by the Auto-Negotiation process. See 28.2.4.1 and 37.2.5.1.

**22.2.4.3.4 Auto-Negotiation expansion (Register 6)**

Register 6 provides 16 bits that are used by the Auto-Negotiation process. See 28.2.4.1 and 37.2.5.1.

<sup>1</sup>Interested applicants should contact the IEEE Standards Department, Institute of Electrical and Electronics Engineers, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, USA.



### 22.2.4.3.5 Auto-Negotiation next page (Register 7)

Register 7 provides 16 bits that are used by the Auto-Negotiation process. See 28.2.4.1 and 37.2.5.1.

### 22.2.4.3.6 Auto-Negotiation link partner Received Next Page (Register 8)

Register 8 provides 16 bits that are used by the Auto-Negotiation process. See 32.5.1 and 37.2.5.1.

### 22.2.4.3.7 MASTER-SLAVE control register (Register 9)

Register 9 provides bit values by 100BASE-T2 (as specified in 32.5) and 1000BASE-T (as specified in 40.5).

### 22.2.4.3.8 MASTER-SLAVE status register (Register 10)

Register 10 provides bit values by 100BASE-T2 (as specified in 32.5) and 1000BASE-T (as specified in 40.5).

### 22.2.4.3.9 PHY specific registers

A particular PHY may provide additional registers beyond those defined above. Register addresses 16 through 31 (decimal) may be used to provide vendor-specific functions or abilities. The definition of Registers 4 through 14 are dependent on the version (Clause 28 or Clause 37) of Auto-Negotiation protocol used by the PHY.

### 22.2.4.4 Extended Status register (Register 15)

The Extended Status register is implemented for all PHYs capable of operation at speeds above 100 Mb/s. The assignment of bits in the Extended Status register is shown in Table 22–9 below. All of the bits in the Extended Status register are read only; a write to the Extended Status register shall have no effect.

**Table 22–9—Extended status register bit definitions**

Bit(s)	Name	Description	R/W <sup>a</sup>
15.15	1000BASE-X Full Duplex	1 = PHY able to perform full duplex 1000BASE-X 0 = PHY not able to perform full duplex 1000BASE-X	RO
15.14	1000BASE-X Half Duplex	1 = PHY able to perform half duplex 1000BASE-X 0 = PHY not able to perform half duplex 1000BASE-X	RO
15.13	1000BASE-T Full Duplex	1 = PHY able to perform full duplex 1000BASE-T 0 = PHY not able to perform full duplex 1000BASE-T	RO
15.12	1000BASE-T Half Duplex	1 = PHY able to perform half duplex 1000BASE-T 0 = PHY not able to perform half duplex 1000BASE-T	RO
15.11:0	Reserved	ignore when read	RO

<sup>a</sup>RO = Read Only

#### 22.2.4.4.1 1000BASE-X full duplex ability

When read as a logic one, bit 15.15 indicates that the PHY has the ability to perform full duplex link transmission and reception using the 1000BASE-X signaling specification. When read as a logic zero, the bit

15.15 indicates that the PHY lacks the ability to perform full duplex link transmission and reception using the 1000BASE-X signaling specification.

**22.2.4.4.2 1000BASE-X half duplex ability**

When read as a logic one, bit 15.14 indicates that the PHY has the ability to perform half duplex link transmission and reception using the 1000BASE-X signaling specification. When read as a logic zero, the bit 15.14 indicates that the PHY lacks the ability to perform half duplex link transmission and reception using the 1000BASE-X signaling specification.

**22.2.4.4.3 1000BASE-T full duplex ability**

When read as a logic one, bit 15.13 indicates that the PHY has the ability to perform full duplex link transmission and reception using the 1000BASE-T signaling specification. When read as a logic zero, the bit 15.13 indicates that the PHY lacks the ability to perform full duplex link transmission and reception using the 1000BASE-T signaling specification.

**22.2.4.4.4 1000BASE-T half duplex ability**

When read as a logic one, bit 15.12 indicates that the PHY has the ability to perform half duplex link transmission and reception using the 1000BASE-T signaling specification. When read as a logic zero, the bit 15.12 indicates that the PHY lacks the ability to perform half duplex link transmission and reception using the 1000BASE-T signaling specification.

**22.2.4.4.5 Reserved bits**

Bits 15:11:0 are reserved for future standardization. They shall be written as zero and shall be ignored when read; however, a PHY shall return the value zero in these bits.

**22.2.4.5 Management frame structure**

Frames transmitted on the MII Management Interface shall have the frame structure shown in Table 22–10. The order of bit transmission shall be from left to right.

**Table 22–10—Management frame format**

	Management frame fields							
	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
READ	1...1	01	10	AAAAA	RRRRR	Z0	DDDDDDDDDDDDDDDDDD	Z
WRITE	1...1	01	01	AAAAA	RRRRR	10	DDDDDDDDDDDDDDDDDD	Z

**22.2.4.5.1 IDLE (IDLE condition)**

The IDLE condition on MDIO is a high-impedance state. All three state drivers shall be disabled and the PHY’s pull-up resistor will pull the MDIO line to a logic one.

**22.2.4.5.2 PRE (preamble)**

At the beginning of each transaction, the station management entity shall send a sequence of 32 contiguous logic one bits on MDIO with 32 corresponding cycles on MDC to provide the PHY with a pattern that it can

use to establish synchronization. A PHY shall observe a sequence of 32 contiguous one bits on MDIO with 32 corresponding cycles on MDC before it responds to any transaction.

If the STA determines that every PHY that is connected to the MDIO signal is able to accept management frames that are not preceded by the preamble pattern, then the STA may suppress the generation of the preamble pattern, and may initiate management frames with the ST (Start of Frame) pattern.

#### 22.2.4.5.3 ST (start of frame)

The start of frame is indicated by a <01> pattern. This pattern assures transitions from the default logic one line state to zero and back to one.

#### 22.2.4.5.4 OP (operation code)

The operation code for a read transaction is <10>, while the operation code for a write transaction is <01>.

#### 22.2.4.5.5 PHYAD (PHY Address)

The PHY Address is five bits, allowing 32 unique PHY addresses. The first PHY address bit transmitted and received is the MSB of the address. A PHY that is connected to the station management entity via the mechanical interface defined in 22.6 shall always respond to transactions addressed to PHY Address zero <00000>. A station management entity that is attached to multiple PHYs must have a priori knowledge of the appropriate PHY Address for each PHY.

#### 22.2.4.5.6 REGAD (Register Address)

The Register Address is five bits, allowing 32 individual registers to be addressed within each PHY. The first Register Address bit transmitted and received is the MSB of the address. The register accessed at Register Address zero <00000> shall be the control register defined in 22.2.4.1, and the register accessed at Register Address one <00001> shall be the status register defined in 22.2.4.2.

#### 22.2.4.5.7 TA (turnaround)

The turnaround time is a 2 bit time spacing between the Register Address field and the Data field of a management frame to avoid contention during a read transaction. For a read transaction, both the STA and the PHY shall remain in a high-impedance state for the first bit time of the turnaround. The PHY shall drive a zero bit during the second bit time of the turnaround of a read transaction. During a write transaction, the STA shall drive a one bit for the first bit time of the turnaround and a zero bit for the second bit time of the turnaround. Figure 22–13 shows the behavior of the MDIO signal during the turnaround field of a read transaction.

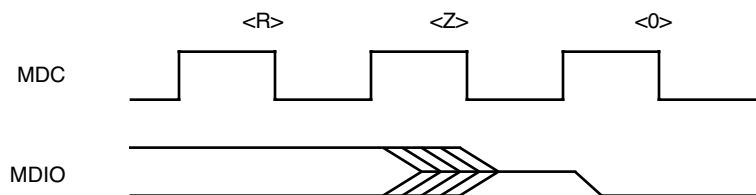


Figure 22–13—Behavior of MDIO during TA field of a read transaction

### 22.2.4.5.8 DATA (data)

The data field is 16 bits. The first data bit transmitted and received shall be bit 15 of the register being addressed.

## 22.3 Signal timing characteristics

All signal timing characteristics shall be measured using the techniques specified in Annex 22C. The signal threshold potentials  $V_{ih(min)}$  and  $V_{il(max)}$  are defined in 22.4.4.1.

The HIGH time of an MII signal is defined as the length of time that the potential of the signal is greater than or equal to  $V_{ih(min)}$ . The LOW time of an MII signal is defined as the length of time that the potential of the signal is less than or equal to  $V_{il(max)}$ .

The setup time of an MII signal relative to an MII clock edge is defined as the length of time between when the signal exits and remains out of the switching region and when the clock enters the switching region. The hold time of an MII signal relative to an MII clock edge is defined as the length of time between when the clock exits the switching region and when the signal enters the switching region.

The propagation delay from an MII clock edge to a valid MII signal is defined as the length of time between when the clock exits the switching region and when the signal exits and remains out of the switching region.

### 22.3.1 Signals that are synchronous to TX\_CLK

Figure 22–14 shows the timing relationship for the signals associated with the transmit data path at the MII connector. The clock to output delay shall be a minimum of 0 ns and a maximum of 25 ns.

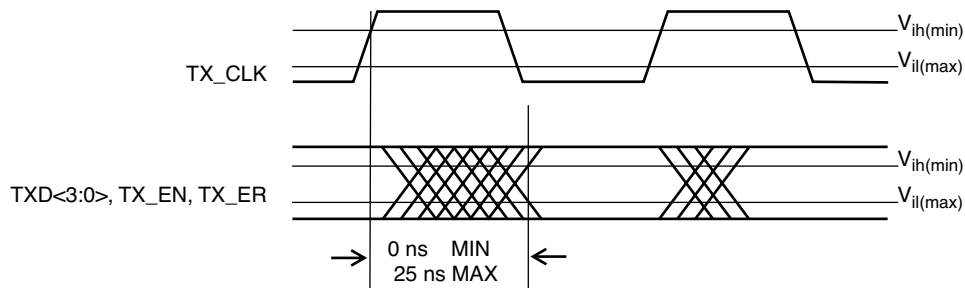


Figure 22–14—Transmit signal timing relationships at the MII

#### 22.3.1.1 TX\_EN

TX\_EN is transitioned by the Reconciliation sublayer synchronously with respect to the TX\_CLK rising edge with the timing as shown in Figure 22–14.

#### 22.3.1.2 TXD<3:0>

TXD<3:0> is transitioned by the Reconciliation sublayer synchronously with respect to the TX\_CLK rising edge with the timing as depicted in Figure 22–14.

#### 22.3.1.3 TX\_ER

TX\_ER is transitioned synchronously with respect to the rising edge of TX\_CLK as shown in Figure 22–14.

### 22.3.2 Signals that are synchronous to RX\_CLK

Figure 22–15 shows the timing relationship for the signals associated with the receive data path at the MII connector. The timing is referenced to the rising edge of the RX\_CLK. The input setup time shall be a minimum of 10 ns and the input hold time shall be a minimum of 10 ns.

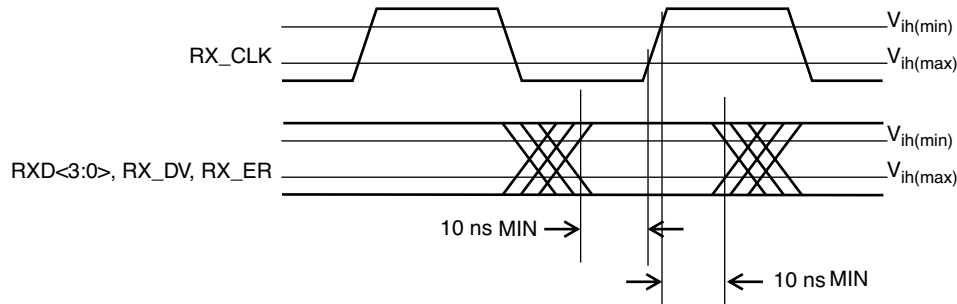


Figure 22–15—Receive signal timing relationships at the MII

#### 22.3.2.1 RX\_DV

RX\_DV is sampled by the Reconciliation sublayer synchronously with respect to the rising edge of RX\_CLK with the timing shown in Figure 22–15.

#### 22.3.2.2 RXD<3:0>

RXD<3:0> is sampled by the Reconciliation sublayer synchronously with respect to the rising edge of RX\_CLK as shown in Figure 22–15. The RXD<3:0> timing requirements must be met at all rising edges of RX\_CLK.

#### 22.3.2.3 RX\_ER

RX\_ER is sampled by the Reconciliation sublayer synchronously with respect to the rising edge of RX\_CLK as shown in Figure 22–15. The RX\_ER timing requirements must be met at all rising edges of RX\_CLK.

### 22.3.3 Signals that have no required clock relationship

#### 22.3.3.1 CRS

CRS is driven by the PHY. Transitions on CRS have no required relationship to either of the clock signals provided at the MII.

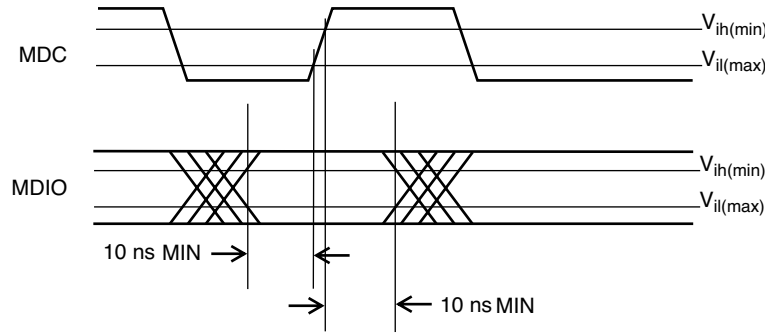
#### 22.3.3.2 COL

COL is driven by the PHY. Transitions on COL have no required relationship to either of the clock signals provided at the MII.

#### 22.3.4 MDIO timing relationship to MDC

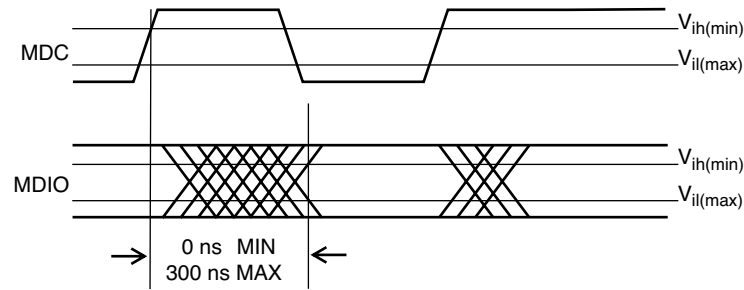
MDIO (Management Data Input/Output) is a bidirectional signal that can be sourced by the Station Management Entity (STA) or the PHY. When the STA sources the MDIO signal, the STA shall provide a minimum

of 10 ns of setup time and a minimum of 10 ns of hold time referenced to the rising edge of MDC, as shown in Figure 22–16, measured at the MII connector.



**Figure 22–16—MDIO sourced by STA**

When the MDIO signal is sourced by the PHY, it is sampled by the STA synchronously with respect to the rising edge of MDC. The clock to output delay from the PHY, as measured at the MII connector, shall be a minimum of 0 ns, and a maximum of 300 ns, as shown in Figure 22–17.



**Figure 22–17—MDIO sourced by PHY**

## 22.4 Electrical characteristics

The electrical characteristics of the MII are specified such that the three application environments described in 22.1 are accommodated. The electrical specifications are optimized for the integrated circuit to integrated circuit application environment, but integrated circuit drivers and receivers that are implemented in compliance with the specification will also support the mother board to daughter board and short cable application environments, provided those environments are constrained to the limits specified in this clause.

NOTE—The specifications for the driver and receiver characteristics can be met with TTL compatible input and output buffers implemented in a digital CMOS ASIC process.

### 22.4.1 Signal levels

The MII uses TTL signal levels, which are compatible with devices operating at a nominal supply voltage of either 5.0 or 3.3 V.

NOTE—Care should be taken to ensure that all MII receivers can tolerate dc input potentials from 0.00 V to 5.50 V, referenced to the COMMON signal, and transient input potentials as high as 7.3 V, or as low as –1.8 V, referenced to the COMMON signal, which can occur when MII signals change state. The transient duration will not exceed 15 ns. The dc source impedance will be no less than  $R_{oh(min)}$ . The transient source impedance will be no less than  $(68 \times 0.85 =) 57.8 \Omega$ .

### 22.4.2 Signal paths

MII signals can be divided into two groups: signals that go between the STA and the PHY, and signals that go between the Reconciliation sublayer and the PHY.

Signals between the STA and the PHY may connect to one or more PHYs. When a signal goes between the STA and a single PHY, the signal's path is a point-to-point transmission path. When a signal goes between the STA and multiple PHYs, the signal's transmission path has drivers and receivers attached in any order along the length of the path and is not considered a point-to-point transmission path.

Signals between the Reconciliation sublayer and the PHY may also connect to one or more PHYs. However, the transmission path of each of these signals shall be either a point-to-point transmission path or a sequence of point-to-point transmission paths connected in series.

All connections to a point-to-point transmission path are at the path ends. The simplest point-to-point transmission path has a driver at one end and a receiver at the other. Point-to-point transmission paths can also have more than one driver and more than one receiver if the drivers and receivers are lumped at the ends of the path, and if the maximum propagation delay between the drivers and receivers at a given end of the path is a very small fraction of the 10%–90% rise/fall time for signals driven onto the path.

The MII shall use unbalanced signal transmission paths. The characteristic impedance  $Z_o$  of transmission paths is not specified for electrically short paths where transmission line reflections can be safely ignored.

The characteristic impedance  $Z_o$  of electrically long transmission paths or path segments shall be  $68 \Omega \pm 15\%$ .

The output impedance of the driver shall be used to control transmission line reflections on all electrically long point-to-point signal paths.

NOTE—In the context of this clause, a transmission path whose round-trip propagation delay is less than half of the 10%–90% rise/fall time of signals driven onto the path is considered an electrically short transmission path.

### 22.4.3 Driver characteristics

The driver characteristics defined in this clause apply to all MII signal drivers. The driver characteristics are specified in terms of both their ac and dc characteristics.

NOTE—Rail-to-rail drivers that comply with the driver output V-I diagrams in Annex 22B will meet the following ac and dc characteristics.

#### 22.4.3.1 DC characteristics

The high (one) logic level output potential  $V_{oh}$  shall be no less than 2.40 V at an output current  $I_{oh}$  of –4.0 mA. The low (zero) logic level output potential  $V_{ol}$  shall not be greater than 0.40 V at an output current  $I_{ol}$  of 4.0 mA.

### 22.4.3.2 AC characteristics

Drivers must also meet certain ac specifications in order to ensure adequate signal quality for electrically long point-to-point transmission paths. The ac specifications shall guarantee the following performance requirements.

The initial incident potential change arriving at the receiving end of a point-to-point MII signal path plus its reflection from the receiving end of the path must switch the receiver input potential monotonically from a valid high (one) level to  $V_{il} \leq V_{il(max)} - 200$  mV, or from a valid low (zero) level to  $V_{ih} \geq V_{ih(min)} + 200$  mV.

Subsequent incident potential changes arriving at the receiving end of a point-to-point MII signal path plus their reflections from the receiving end of the path must not cause the receiver input potential to reenter the range  $V_{il(max)} - 200$  mV  $< V_i < V_{ih(min)} + 200$  mV except when switching from one valid logic level to the other. Such subsequent incident potential changes result from a mismatch between the characteristic impedance of the signal path and the driver output impedance.

### 22.4.4 Receiver characteristics

The receiver characteristics are specified in terms of the threshold levels for the logical high (one) and logical low (zero) states. In addition, receivers must meet the input current and capacitance limits.

#### 22.4.4.1 Voltage thresholds

An input potential  $V_i$  of 2.00 V or greater shall be interpreted by the receiver as a logical high (one). Thus,  $V_{ih(min)} = 2.00$  V. An input potential  $V_i$  of 0.80 V or less shall be interpreted by the receiver as a logical low (zero). Thus,  $V_{il(max)} = 0.80$  V. The switching region is defined as signal potentials greater than  $V_{il(max)}$  and less than  $V_{ih(min)}$ . When the input signal potential is in the switching region, the receiver output is undefined.

#### 22.4.4.2 Input current

The input current requirements shall be measured at the MII connector and shall be referenced to the +5 V supply and COMMON pins of the connector. The input current requirements shall be met across the full range of supply voltage specified in 22.5.1.

The bidirectional signal MDIO has two sets of input current requirements. The MDIO drivers must be disabled when the input current measurement is made.

The input current characteristics for all MII signals shall fall within the limits specified in Table 22–11.

NOTE—These limits for dc input current allow the use of weak resistive pull-ups or pull-downs on the input of each MII signal. They allow the use of weak resistive pull-downs on the signals other than COL, MDC, and MDIO. They allow the use of a weak resistive pull-up on the signal COL. They allow the use of a resistive pull-down of  $2 \text{ k}\Omega \pm 5\%$  on the MDIO signal in the STA. They require a resistive pull-up of  $1.5 \text{ k}\Omega \pm 5\%$  on the MDIO signal in a PHY that is attached to the MII via the mechanical interface specified in 22.6. The limits on MDC and MDIO allow the signals to be “bused” to several PHYs that are contained on the same printed circuit assembly, with a single PHY attached via the MII connector.

#### 22.4.4.3 Input capacitance

For all signals other than MDIO, the receiver input capacitance  $C_i$  shall not exceed 8 pF.

For the MDIO signal, the transceiver input capacitance shall not exceed 10 pF.



**Table 22–11 – Input current limits**

Symbol	Parameter	Condition	Signal(s)	Min ( $\mu\text{A}$ )	Max ( $\mu\text{A}$ )
$I_{ih}$	Input High Current	$V_i=5.25\text{ V}$	All except COL, MDC, MDIO <sup>a</sup>	—	200
			COL <sup>b</sup>	—	20
			MDC <sup>c</sup>	—	20
			MDIO <sup>d</sup>	—	3000
			MDIO <sup>e</sup>	—	20
$I_{il}$	Input Low Current	$V_i=0.00\text{ V}$	All except COL, MDC, MDIO <sup>a</sup>	–20	—
			COL <sup>b</sup>	–200	—
			MDC <sup>c</sup>	–20	—
			MDIO <sup>d</sup>	–180	—
			MDIO <sup>e</sup>	–3800	—
$I_{iq}$	Input Quiescent Current	$V_i=2.4\text{ V}$	MDIO <sup>d</sup>	—	1450
			MDIO <sup>e</sup>	–1450	—

<sup>a</sup>Measured at input of Reconciliation sublayer for CRS, RXD<3:0>, RX\_CLK, RX\_DV, RX\_ER, and TX\_CLK. Measured at inputs of PHY for TXD<3:0>, TX\_EN, and TX\_ER.

<sup>b</sup>Measured at input of Reconciliation sublayer.

<sup>c</sup>Measured at input of PHY.

<sup>d</sup>Measured at input of STA.

<sup>e</sup>Measured at input of PHY, which can be attached via the mechanical interface specified in 22.6.

### 22.4.5 Cable characteristics

The MII cable consists of a bundle of individual twisted pairs of conductors with an overall shield covering this bundle. Each twisted pair shall be composed of a conductor for an individual signal and a return path dedicated to that signal.

NOTE—It is recommended that the signals RX\_CLK and TX\_CLK be connected to pairs that are located in the center of the cable bundle.

#### 22.4.5.1 Conductor size

The specifications for dc resistance in 22.4.5.6 and characteristic impedance in 22.4.5.2 assume a conductor size of 0.32 mm (28 AWG).

#### 22.4.5.2 Characteristic impedance

The single-ended characteristic impedance of each twisted pair shall be  $68\ \Omega \pm 10\%$ . The characteristic impedance measurement shall be performed with the return conductor connected to the cable's overall shield at both ends of the cable.

### 22.4.5.3 Delay

The propagation delay for each twisted pair, measured from the MII connector to the PHY, shall not exceed 2.5 ns. The measurement shall be made with the return conductor of the pair connected to the cable's overall shield at both ends of the cable. The propagation delay shall be measured at a frequency of 25 MHz.

### 22.4.5.4 Delay variation

The variation in the propagation delay of the twisted pairs in a given cable bundle, measured from the MII connector to the PHY, shall not exceed 0.1 ns. The measurement shall be made with the return conductor of the pair connected to the cable's overall shield at both ends of the cable.

### 22.4.5.5 Shielding

The overall shield must provide sufficient shielding to meet the requirements of protection against electromagnetic interference.

The overall shield shall be terminated to the connector shell as defined in 22.6.2. A double shield, consisting of both braid and foil shielding, is strongly recommended.

### 22.4.5.6 DC resistance

The dc resistance of each conductor in the cable, including the contact resistance of the connector, shall not exceed 150 m $\Omega$  measured from the MII connector to the remote PHY.

## 22.4.6 Hot insertion and removal

The insertion or removal of a PHY from the MII with power applied (hot insertion or removal) shall not damage the devices on either side of the MII. In order to prevent contention between multiple output buffers driving the PHY output signals, a PHY that is attached to the MII via the mechanical interface defined in 22.6 shall ensure that its output buffers present a high impedance to the MII during the insertion process, and shall ensure that this condition persists until the output buffers are enabled via the Isolate control bit in the management interface basic register.

NOTE—The act of inserting or removing a PHY from an operational system may cause the loss of one or more packets or management frames that may be in transit across the MII or MDI.

## 22.5 Power supply

When the mechanical interface defined in 22.6 is used to interconnect printed circuit subassemblies, the Reconciliation sublayer shall provide a regulated power supply for use by the PHY.

The power supply shall use the following MII lines:

- +5 V: The plus voltage output to the PHY.
- COMMON: The return to the power supply.

### 22.5.1 Supply voltage

The regulated supply voltage to the PHY shall be 5 Vdc  $\pm$  5% at the MII connector with respect to the COMMON circuit at the MII over the range of load current from 0 to 750 mA. The method of over/under voltage protection is not specified; however, under no conditions of operation shall the source apply a voltage to the +5 V circuit of less than 0 V or greater than +5.25 Vdc.

Implementations that provide a conversion from the MII to the Attachment Unit Interface (AUI) to support connection to 10 Mb/s Medium Attachment Units (MAUs) will require a supplemental power source in order to meet the AUI power supply requirements specified in 7.5.2.5.

### 22.5.2 Load current

The sum of the currents carried on the +5 V lines shall not exceed 750 mA, measured at the MII connector. The surge current drawn by the PHY shall not exceed 5 A peak for a period of 10 ms. The PHY shall be capable of powering up from 750 mA current limited sources.

### 22.5.3 Short-circuit protection

Adequate provisions shall be made to ensure protection of the power supply from overload conditions, including a short circuit between the +5 V lines and the COMMON lines.

## 22.6 Mechanical characteristics

When the MII is used to interconnect two printed circuit assemblies via a short length of cable, the cable shall be connected to the circuit assembly that implements the Reconciliation sublayer by means of the mechanical interface defined in this clause.

### 22.6.1 Definition of mechanical interface

A 40-pole connector having the mechanical mateability dimensions as specified in IEC 61076-3-101: 1997 shall be used for the MII connector. The circuit assembly that contains the MAC sublayer and Reconciliation sublayer shall have a female connector with screw locks, and the mating cable shall have a male connector with jack screws.

No requirements are imposed on the mechanical interface used to connect the MII cable to the PHY circuit assembly when the MII cable is permanently attached to the PHY circuit assembly, as shown in Figure 22–2. If the cable is not permanently attached to the PHY circuit assembly, then a male connector with jack screws shall be used for the MII connector at the PHY circuit assembly.

NOTE—All MII conformance tests are performed at the mating surfaces of the MII connector at the Reconciliation sublayer end of the cable. If a PHY circuit assembly does not have a permanently attached cable, the vendor must ensure that all of the requirements of this clause are also met when a cable that meets the requirements of 22.4.5 is used to attach the PHY circuit assembly to the circuit assembly that contains the Reconciliation sublayer.

### 22.6.2 Shielding effectiveness and transfer impedance

The shells of these connectors shall be plated with conductive material to ensure the integrity of the current path from the cable shield to the chassis. The transfer impedance of this path shall not exceed the values listed in Table 22–12, after a minimum of 500 cycles of mating and unmating. The shield transfer impedance values listed in the table are measured in accordance with the procedure defined in Annex L of IEEE Std 1394-1995<sup>TM</sup> [B28]<sup>2</sup>.

All additions to provide for female shell to male shell conductivity shall be on the shell of the connector with male contacts. There should be multiple contact points around the sides of this shell to provide for shield continuity.

---

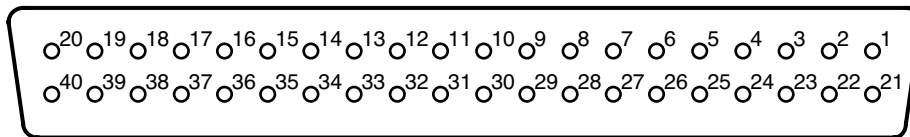
<sup>2</sup>The numbers in brackets correspond to those of the bibliography in Annex A in Part 1 of this standard.

**Table 22–12—Transfer impedance performance requirements**

Frequency	Value
30 MHz	–26 dBΩ
159 MHz	–13 dBΩ
500 MHz	–5 dBΩ

**22.6.3 Connector pin numbering**

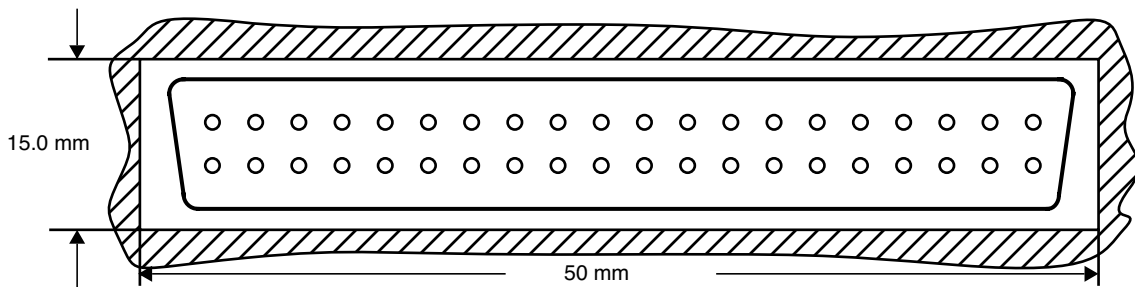
Figure 22–18 depicts the MII connector pin numbering, as seen looking into the contacts of a female connector from the mating side.



**Figure 22–18—MII connector pin numbering**

**22.6.4 Clearance dimensions**

The circuit assembly that contains the MAC sublayer and Reconciliation sublayer shall provide sufficient clearance around the MII connector to allow the attachment of cables that use die cast metal backshells and overmold assemblies. This requirement may be met by providing the clearance dimensions shown in Figure 22–19.



**Figure 22–19—MII connector clearance dimensions**

### 22.6.5 Contact assignments

Table 22–13 shows the assignment of circuits to connector contacts.

**Table 22–13—MII connector contact assignments**

Contact	Signal name	Contact	Signal name
1	+5 V	21	+5 V
2	MDIO	22	COMMON
3	MDC	23	COMMON
4	RXD<3>	24	COMMON
5	RXD<2>	25	COMMON
6	RXD<1>	26	COMMON
7	RXD<0>	27	COMMON
8	RX_DV	28	COMMON
9	RX_CLK	29	COMMON
10	RX_ER	30	COMMON
11	TX_ER	31	COMMON
12	TX_CLK	32	COMMON
13	TX_EN	33	COMMON
14	TXD<0>	34	COMMON
15	TXD<1>	35	COMMON
16	TXD<2>	36	COMMON
17	TXD<3>	37	COMMON
18	COL	38	COMMON
19	CRS	39	COMMON
20	+5 V	40	+5 V

## 22.7 Protocol Implementation Conformance Statement (PICS) proforma for Clause 22, Reconciliation Sublayer (RS) and Media Independent Interface (MII)<sup>3</sup>

### 22.7.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 22, Reconciliation Sublayer (RS) and Media Independent Interface (MII), shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

### 22.7.2 Identification

#### 22.7.2.1 Implementation identification

Supplier	
Contact point for enquiries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s)	
NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.	
NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

#### 22.7.2.2 Protocol summary

Identification of protocol standard	IEEE Std 802.3-2002 <sup>®</sup> , Clause 22, Reconciliation Sublayer (RS) and Media Independent Interface (MII)
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required? No <input type="checkbox"/> Yes <input type="checkbox"/> (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3-2002 <sup>®</sup> .)	

Date of Statement	
-------------------	--

#### 22.7.2.3 Major capabilities/options

Item	Feature	Subclause	Status	Support	Value/Comment
*GM	Implementation of GMII	22.2.4	O	Yes <input type="checkbox"/> No <input type="checkbox"/>	

<sup>3</sup>Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

### 22.7.3 PICS proforma tables for reconciliation sublayer and media independent interface

#### 22.7.3.1 Mapping of PLS service primitives

Item	Feature	Subclause	Status	Support	Value/Comment
PL1	Response to RX_ER	22.2.1.5	M		Must produce FrameCheckError at MAC

#### 22.7.3.2 MII signal functional specifications

Item	Feature	Subclause	Status	Support	Value/Comment
SF1	TX_CLK frequency	22.2.2.1	M		25% of transmitted data rate (25 MHz or 2.5 MHz)
SF2	TX_CLK duty cycle	22.2.2.1	M		35% to 65%
SF3	RX_CLK min high/low time	22.2.2.2	M		35% of nominal period
SF4	RX_CLK synchronous to recovered data	22.2.2.2	M		
SF5	RX_CLK frequency	22.2.2.2	M		25% of received data rate (25 MHz or 2.5 MHz)
SF6	RX_CLK duty cycle	22.2.2.2	M		35% to 65%
SF7	RX_CLK source due to loss of signal	22.2.2.2	M		Nominal clock reference (e.g., TX_CLK reference)
SF8	RX_CLK transitions only while RX_DV de-asserted	22.2.2.2	M		
SF9	RX_CLK max high/low time following de-assertion of RX_DV	22.2.2.2	M		max 2 times the nominal period
SF10	TX_EN assertion	22.2.2.3	M		On first nibble of preamble
SF11	TX_EN remains asserted	22.2.2.3	M		Stay asserted while all nibbles are transmitted over MII
SF12	TX_EN transitions	22.2.2.3	M		Synchronous with TX_CLK
SF13	TX_EN negation	22.2.2.3	M		Before first TX_CLK after final nibble of frame
SF14	TXD<3:0> transitions	22.2.2.4	M		Synchronous with TX_CLK
SF15	TXD<3:0> effect on PHY while TX_EN is de-asserted	22.2.2.4	M		No effect
SF16	TX_ER transitions	22.2.2.5	M		Synchronous with TX_CLK
SF17	TX_ER effect on PHY while TX_EN is asserted	22.2.2.5	M		Cause PHY to emit invalid symbol
SF18	TX_ER effect on PHY while operating at 10 Mb/s, or when TX_EN is de-asserted	22.2.2.5	M		No effect on PHY
SF19	TX_ER implementation	22.2.2.5	M		At MII of a PHY

**22.7.3.2 MII signal functional specifications (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
SF20	TX_ER pulled down if not actively driven	22.2.2.5	M		At MII of a repeater or MAC/RS only
SF21	RX_DV transitions	22.2.2.6	M		Synchronous with RX_CLK
SF22	RX_DV assertion	22.2.2.6	M		From first recovered nibble to final nibble of a frame per Figure 22-6
SF23	RX_DV negation	22.2.2.6	M		Before the first RX_CLK follows the final nibble per Figure 22-6
SF24	RXD<3:0> effect on Reconciliation sublayer while RX_DV is de-asserted	22.2.2.7	M		No effect
SF25	RX_ER assertion	22.2.2.8	M		By PHY to indicate error
SF26	RX_ER transitions	22.2.2.8	M		Synchronous with RX_CLK
SF27	RX_ER effect on Reconciliation sublayer while RX_DV is de-asserted	22.2.2.8	M		No effect
SF28	CRS assertion	22.2.2.9	M		By PHY when either transmit or receive is NON-IDLE
SF29	CRS de-assertion	22.2.2.9	M		By PHY when both transmit and receive are IDLE
SF30	CRS assertion during collision	22.2.2.9	M		Remain asserted throughout
SF31	COL assertion	22.2.2.10	M		By PHY upon detection of collision on medium
SF32	COL remains asserted while collision persists	22.2.2.10	M		
SF33	COL response to SQE	22.2.2.10	M		Assertion by PHY
SF34	MDC min high/low time	22.2.2.11	M		160 ns
SF35	MDC min period	22.2.2.11	M		400 ns
SF36	MDIO uses three-state drivers	22.2.2.12	M		
SF37	PHY pullup on MDIO	22.2.2.12	M		1.5 k $\Omega$ $\pm$ 5% (to +5 V)
SF38	STA pulldown on MDIO	22.2.2.12	M		2 k $\Omega$ $\pm$ 5% (to 0 V)



### 22.7.3.3 Frame structure

Item	Feature	Subclause	Status	Support	Value/Comment
FS1	Format of transmitted frames	22.2.3	M		Per Figure 22–10
FS2	Nibble transmission order	22.2.3	M		Per Figure 22–11
FS3	Preamble 7 octets long	22.2.3.2.1	M		10101010 10101010 10101010 10101010 10101010 10101010 10101010
FS4	Preamble and SFD transmission	22.2.3.2.1	M		Per Table 22–3
FS5	Preamble and SFD reception	22.2.3.2.2	M		Per Table 22–4, Table 22–5
FS6	N octets transmitted as 2N nibbles	22.2.3.3	M		Per Figure 22–11
FS7	Indication of excess nibbles	22.2.3.5	M		Frame contains non-integer number of octets is received

### 22.7.3.4 Management functions

Item	Feature	Subclause	Status	Support	Value/Comment
MF1	Incorporate of basic register set	22.2.4	M		Two 16-bit registers as Control register (Register 0) and Status register (Register 1)
MF2	Action on reset	22.2.4.1.1	M		Reset the entire PHY including Control and Status to default value and set bit 0.15 = 1
MF3	Return 1 until reset completed	22.2.4.1.1	M		Yes (when reset is done, 0.15 is self clearing)
MF4	Reset completes within 0.5 s	22.2.4.1.1	M		
MF5	Loopback mode	22.2.4.1.2	M		Whenever 0.14 is 1
MF6	Receive circuitry isolated from network in loopback mode	22.2.4.1.2	M		
MF7	Effect of assertion of TX_EN in loopback mode	22.2.4.1.2	M		No transmission
MF8	Propagation of data in loopback mode	22.2.4.1.2	M		PHY accepts transmit data and return it as receive data
MF9	Delay from TX_EN to RX_DV in loopback mode	22.2.4.1.2	M		Less than 512 BT
MF10	Behavior of COL in loopback mode	22.2.4.1.2	M		De-asserted (for 0.7 = 0)
MF11	Behavior of COL in loopback mode	22.2.4.1.2	M		If 0.7 = 1, see MF33 and MF34
MF12	Value of speed selection bits	22.2.4.1.3	M	Yes [ ]	Set to match a valid PHY speed

**22.7.3.4 Management functions (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
MF13	Ignore writes to speed selection bits for unsupported speed	22.2.4.1.3	M	Yes [ ]	
MF14	Auto-Negotiation enable	22.2.4.1.4	M		By setting 0.12 = 1
MF15	Duplex mode, speed selection have no effect when Auto-Negotiation is enabled	22.2.4.1.4	M	Yes [ ]	If 0.12=1, bits 0.13, 0.8 and 0.6 have no effect on link configuration
MF16	PHY without Auto-Negotiation returns value of zero	22.2.4.1.4	M		Yes (if 1.3=0, then 0.12=0)
MF17	PHY without Auto-Negotiation ignores writes to enable bit	22.2.4.1.4	M		Yes (if 1.3=0, 0.12 always = 0 and cannot be changed)
MF18	Response to management transactions in power down	22.2.4.1.5	M		Remains active
MF19	Spurious signals in power down	22.2.4.1.5	M		None (not allowed)
MF20	TX_CLK and RX_CLK stabilize within 0.5 s	22.2.4.1.5	M		Yes (after both bits 0.11 and 0.10 are cleared to zero)
MF21	PHY Response to input signals while isolated	22.2.4.1.6	M		NONE
MF22	High impedance on PHY output signals while isolated	22.2.4.1.6	M	Yes [ ]	Yes (TX_CLK, RX_CLK, RX_DV, RX_ER, RXD bundle, COL, and CRS)
MF23	Response to management transactions while isolated	22.2.4.1.6	M		Remains active
MF24	Default value of isolate	22.2.4.1.6	M		0.10 =1
MF25	PHY without Auto-Negotiation returns value of zero	22.2.4.1.7	M		0.9 = 0 if 1.3 = 0 or 0.12 = 0
MF26	PHY without Auto-Negotiation ignores writes to restart bit	22.2.4.1.7	M		0.9 = 0, cannot be changed if 1.3 = 0 or 0.12 = 0
MF27	Restart Auto-Negotiation	22.2.4.1.7	M		When 0.9 = 1 if 0.12 = 1 and 1.3 = 1
MF28	Return 1 until Auto-Negotiation initiated	22.2.4.1.7	M		0.9 is self clearing to 0
MF29	Auto-Negotiation not effected by clearing bit	22.2.4.1.7	M		
MF30	Value of duplex mode bit for PHYs with one duplex mode	22.2.4.1.8	M		Set 0.8 to match the correct PHY duplex mode
MF31	PHY with one duplex mode ignores writes to duplex bit	22.2.4.1.8	M		Yes (0.8 remains unchanged)
MF32	Loopback not affected by duplex mode	22.2.4.1.8	M		Yes (0.8 has no effect on PHY when 0.14 = 1)
MF33	Assertion of COL in collision test mode	22.2.4.1.9	M		Within 512 BT after TX_EN is asserted

**22.7.3.4 Management functions (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
MF34	De-assertion of COL in collision test mode	22.2.4.1.9	M	Yes [ ]	After TX_EN is de-asserted within: MII = 4 BT, GMII = 16 BT
MF35	Reserved bits written as zero	22.2.4.1.11	M		
MF36	Reserved bits ignored when read	22.2.4.1.11	M		
MF37	PHY returns 0 in reserved bits	22.2.4.1.11	M		
MF38	Effect of write on status register	22.2.4.2	M		No effect
MF39	Reserved bits ignored when read	22.2.4.2.8	M		
MF40	PHY returns 0 in reserved bits	22.2.4.2.8	M		
MF41	PHY returns 0 if Auto-Negotiation disabled	22.2.4.2.10	M		Yes (1.5 = 0 when 0.12 = 0)
MF42	PHY returns 0 if it lacks ability to perform Auto-Negotiation	22.2.4.2.10	M		Yes (1.5 = 0 when 1.3 = 0)
MF43	Remote fault has latching function	22.2.4.2.11	M		Yes (once set will remain set until cleared)
MF44	Remote fault cleared on read	22.2.4.2.11	M		Yes
MF45	Remote fault cleared on reset	22.2.4.2.11	M		Yes (when 0.15 = 1)
MF46	PHY without remote fault returns value of zero	22.2.4.2.11	M		Yes (1.4 always 0)
MF47	Link status has latching function	22.2.4.2.13	M		Yes (once cleared by link failure will remain cleared until read by MII)
MF48	Jabber detect has latching function	22.2.4.2.14	M		Yes (once set will remain set until cleared)
MF49	Jabber detect cleared on read	22.2.4.2.14	M		
MF50	Jabber detect cleared on reset	22.2.4.2.14	M		
MF51	All PHYs operating at rates of 100 Mb/s or above return 0 for jabber detect	22.2.4.2.14	M	Yes [ ]	Yes (1.1 always = 0 for all PHYs operating at rates of 100 Mb/s or above)
MF52	MDIO not driven if register read is unimplemented	22.2.4.3	M		Yes (MDIO remain high impedance)
MF53	Write has no effect if register written is unimplemented	22.2.4.3	M		
MF54	Registers 2 and 3 constitute unique identifier for PHY type	22.2.4.3.1	M		
MF55	MSB of PHY identifier is 2.15	22.2.4.3.1	M		
MF56	LSB of PHY identifier is 3.0	22.2.4.3.1	M		

**22.7.3.4 Management functions (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
MF57	Composition of PHY identifier	22.2.4.3.1	O	Yes [ ] No [ ]	22-bit OUI, 6-bit model, 4-bit version per Figure 22–12
MF58	Format of management frames	22.2.4.5	M		Per Table 22–9
MF59	Idle condition on MDIO	22.2.4.5.1	M		High impedance state
MF60	MDIO preamble sent by STA	22.2.4.5.2	M		32 contiguous logic one bits
MF61	MDIO preamble observed by PHY	22.2.4.5.2	M		32 contiguous logic one bits
MF62	Assignment of PHYAD 0	22.2.4.5.5	M		Address of PHY attached via Mechanical Interface
MF63	Assignment of REGAD 0	22.2.4.5.6	M		MII control register address
MF64	Assignment of REGAD 1	22.2.4.5.6	M		MII status register address
MF65	High impedance during first bit time of turnaround in read transaction	22.2.4.5.7	M		
MF66	PHY drives zero during second bit time of turnaround in read transaction	22.2.4.5.7	M		
MF67	STA drives MDIO during turnaround in write transaction	22.2.4.5.7	M		
MF68	First data bit transmitted	22.2.4.5.8	M		Bit 15 of the register being addressed
MF69	Incorporate Extended Status register	22.2.4	GM:M	Yes [ ] NA [ ]	16-bit register Extended Status register (Register 15)
MF70	Reserved bits written as zero	22.2.4.2.8	GM:M	Yes [ ] NA [ ]	
MF71	Extended Status	22.2.4.2.16	GM:M	Yes [ ] NA [ ]	Yes (1.8 always = 1 for 1000 Mb/s operation)
MF72	Write to Extended Status register	22.2.4.4	GM:M	Yes [ ] NA [ ]	No effect
MF73	Reserved bits written as zero	22.2.4.4.5	GM:M	Yes [ ] NA [ ]	
MF74	Reserved bits ignored when read	22.2.4.4.5	GM:M	Yes [ ] NA [ ]	
MF75	PHY returns 0 in reserved bits	22.2.4.4.5	GM:M	Yes [ ] NA [ ]	

### 22.7.3.5 Signal timing characteristics

Item	Feature	Subclause	Status	Support	Value/Comment
ST1	Timing characteristics measured in accordance with Annex 22C	22.3	M		
ST2	Transmit signal clock to output delay	22.3.1	M		Min = 0 ns; Max = 25 ns per Figure 22–14
ST3	Receive signal setup time	22.3.2	M		Min = 10 ns per Figure 22–15
ST4	Receive signal hold time	22.3.2	M		Min = 10 ns per Figure 22–15
ST5	MDIO setup and hold time	22.3.4	M		Setup min = 10 ns; Hold min = 10 ns per Figure 22–16
ST6	MDIO clock to output delay	22.3.4	M		Min = 0 ns; Max = 300 ns per Figure 22–17

### 22.7.3.6 Electrical characteristics

Item	Feature	Subclause	Status	Support	Value/Comment
EC1	Signal paths are either point to point, or a sequence of point-to-point transmission paths	22.4.2	M		
EC2	MII uses unbalanced signal transmission paths	22.4.2	M		
EC3	Characteristic impedance of electrically long paths	22.4.2	M		$68 \Omega \pm 15\%$
EC4	Output impedance of driver used to control reflections	22.4.2	M		On all electrically long point to point signal paths
EC5	$V_{oh}$	22.4.3.1	M		$\geq 2.4 \text{ V}$ ( $I_{oh} = -4 \text{ mA}$ )
EC6	$V_{ol}$	22.4.3.1	M		$\leq 0.4 \text{ V}$ ( $I_{ol} = 4 \text{ mA}$ )
EC7	Performance requirements to be guaranteed by ac specifications	22.4.3.2	M		Min switching potential change (including its reflection) $\geq 1.8 \text{ V}$
EC8	$V_{ih(min)}$	22.4.4.1	M		2 V
EC9	$V_{il(max)}$	22.4.4.1	M		0.8 V
EC10	Input current measurement point	22.4.4.2	M		At MII connector
EC11	Input current reference potentials	22.4.4.2	M		Reference to MII connector +5 V and COMMON pins
EC12	Input current reference potential range	22.4.4.2	M		0 V to 5.25 V
EC13	Input current limits	22.4.4.2	M		Per Table 22–10

**22.7.3.6 Electrical characteristics (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
EC14	Input capacitance for signals other than MDIO	22.4.4.3	M		$\leq 8$ pF
EC15	Input capacitance for MDIO	22.4.4.3	M		$\leq 10$ pF
EC16	Twisted-pair composition	22.4.5	M		Conductor for each signal with dedicated return path
EC17	Single-ended characteristic impedance	22.4.5.2	M		$68 \Omega \pm 10\%$
EC18	Characteristic impedance measurement method	22.4.5.2	M		With return conductor connected to cable shield
EC19	Twisted-pair propagation delay	22.4.5.3	M		$\leq 2.5$ ns
EC20	Twisted-pair propagation delay measurement method	22.4.5.3	M		With return conductor connected to cable shield
EC21	Twisted-pair propagation delay measurement frequency	22.4.5.3	M		25 MHz
EC22	Twisted-pair propagation delay variation	22.4.5.4	M		$\leq 0.1$ ns
EC23	Twisted-pair propagation delay variation measurement method	22.4.5.4	M		With return conductor connected to cable shield
EC24	Cable shield termination	22.4.5.5	M		To the connector shell
EC25	Cable conductor DC resistance	22.4.5.6	M		$\leq 150$ m $\Omega$
EC26	Effect of hot insertion/removal	22.4.6	M		Causes no damage
EC27	State of PHY output buffers during hot insertion	22.4.6	M		High impedance
EC28	State of PHY output buffers after hot insertion	22.4.6	M		High impedance until enabled via Isolate bit

**22.7.3.7 Power supply**

Item	Feature	Subclause	Status	Support	Value/Comment
PS1	Regulated power supply provided	22.5	M		To PHY by Reconciliation sublayer
PS2	Power supply lines	22.5	M		+5 V and COMMON (return of +5 V)
PS3	Regulated supply voltage limits	22.5.1	M		5 Vdc $\pm$ 5%
PS4	Over/under voltage limits	22.5.1	M		Over limit = 5.25 Vdc Under limit = 0 V
PS5	Load current limit	22.5.2	M		750 mA
PS6	Surge current limit	22.5.2	M		$\leq$ 5 A peak for 10 ms
PS7	PHY can power up from current limited source	22.5.2	M		From 750 mA current limited source
PS8	Short-circuit protection	22.5.2	M		When +5 V and COMMON are shorted

**22.7.3.8 Mechanical characteristics**

Item	Feature	Subclause	Status	Support	Value/Comment
*MC1	Use of Mechanical Interface	22.6	O		Optional
MC2	Connector reference standard	22.6.1	MC1:M		IEC 61076-3-101: 1997
MC3	Use of female connector	22.6.1	MC1:M		At MAC/RS side
MC4	Use of male connector	22.6.1	MC1:M		At PHY mating cable side
MC5	Connector shell plating	22.6.2	MC1:M		Use conductive material
MC6	Shield transfer impedance	22.6.2	MC1:M		After 500 cycles of mating/unmating, per Table 22-11
MC7	Additions to provide for female shell to male shell conductivity	22.6.2	MC1:M		On shell of conductor with male contacts
MC8	Clearance dimensions	22.6.4	MC1:M		15 mm $\times$ 50 mm, per Figure 22-19

## 23. Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) sublayer and baseband medium, type 100BASE-T4

NOTE—This PHY is not recommended for new installations.

### 23.1 Overview

The 100BASE-T4 PCS, PMA, and baseband medium specifications are aimed at users who want 100 Mb/s performance, but would like to retain the benefits of using voice-grade twisted-pair cable. 100BASE-T4 signaling requires four pairs of Category 3 or better cable, installed according to ISO/IEC 11801: 1995, as specified in 23.6. This type of cable, and the connectors used with it, are simple to install and reconfigure. 100BASE-T4 does not transmit a continuous signal between packets, which makes it useful in battery powered applications. The 100BASE-T4 PHY is one of the 100BASE-T family of high-speed CSMA/CD network specifications.

#### 23.1.1 Scope

This clause defines the type 100BASE-T4 Physical Coding Sublayer (PCS), type 100BASE-T4 Physical Medium Attachment (PMA) sublayer, and type 100BASE-T4 Medium Dependent Interface (MDI). Together, the PCS and PMA layers comprise a 100BASE-T4 Physical Layer (PHY). Provided in this document are full functional, electrical, and mechanical specifications for the type 100BASE-T4 PCS, PMA, and MDI. This clause also specifies the baseband medium used with 100BASE-T4.

#### 23.1.2 Objectives

The following are the objectives of 100BASE-T4:

- a) To support the CSMA/CD MAC in the half duplex mode of operation.
- b) To support the 100BASE-T MII, Repeater, and optional Auto-Negotiation.
- c) To provide 100 Mb/s data rate at the MII.
- d) To provide for operating over unshielded twisted pairs of Category 3, 4, or 5 cable, installed as horizontal runs in accordance with ISO/IEC 11801: 1995, as specified in 23.6, at distances up to 100 m (328 ft).
- e) To allow for a nominal network extent of 200 m, including:
  - 1) Unshielded twisted-pair links of 100 m.
  - 2) Two-repeater networks of approximately a 200 m span.
- f) To provide a communication channel with a mean ternary symbol error rate, at the PMA service interface, of less than one part in  $10^8$ .

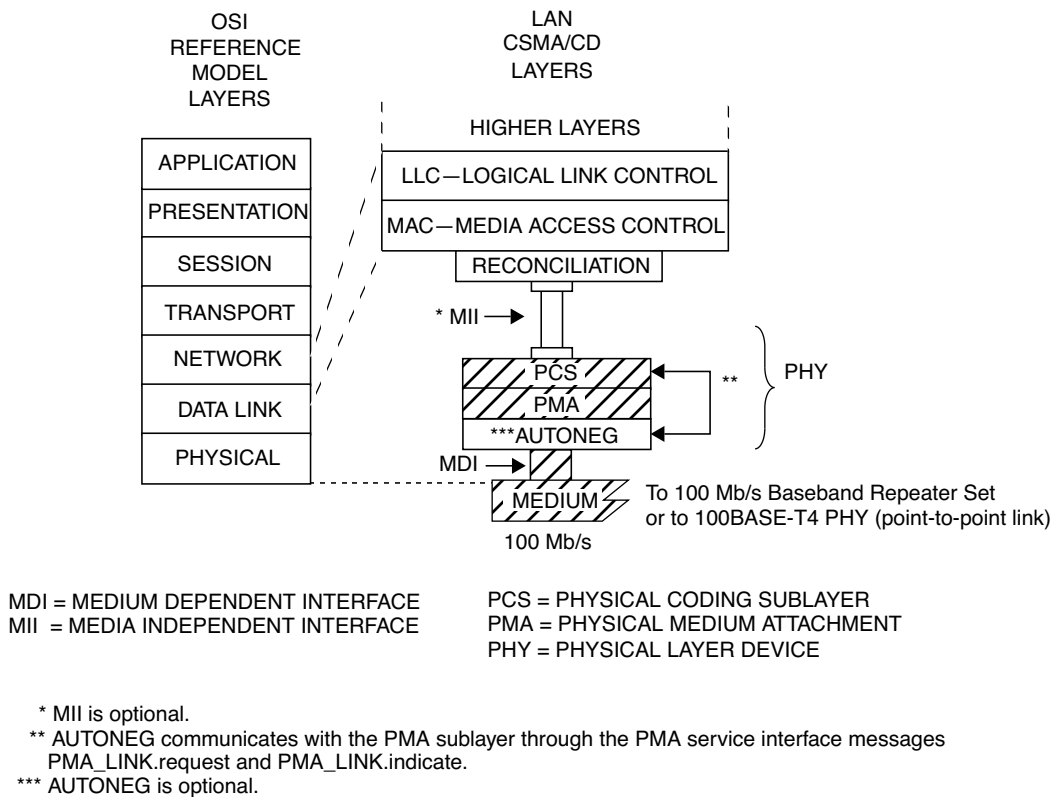
#### 23.1.3 Relation of 100BASE-T4 to other standards

Relations between the 100BASE-T4 PHY and the ISO/IEC Open Systems Interconnection (OSI) reference model and the IEEE 802.3<sup>®</sup> CSMA/CD LAN model are shown in Figure 23–1. The PHY Layers shown in Figure 23–1 connect one Clause 4 Media Access Control (MAC) layer to a Clause 27 repeater. This clause also discusses other variations of the basic configuration shown in Figure 23–1. This whole clause builds on Clauses 1 through 4 of this standard.

#### 23.1.4 Summary

The following paragraphs summarize the PCS and PMA clauses of this standard.





**Figure 23-1 – Type 100BASE-T4 PHY relationship to the ISO Open Systems Interconnection (OSI) reference model and the IEEE 802.3® CSMA/CD LAN model**

**23.1.4.1 Summary of Physical Coding Sublayer (PCS) specification**

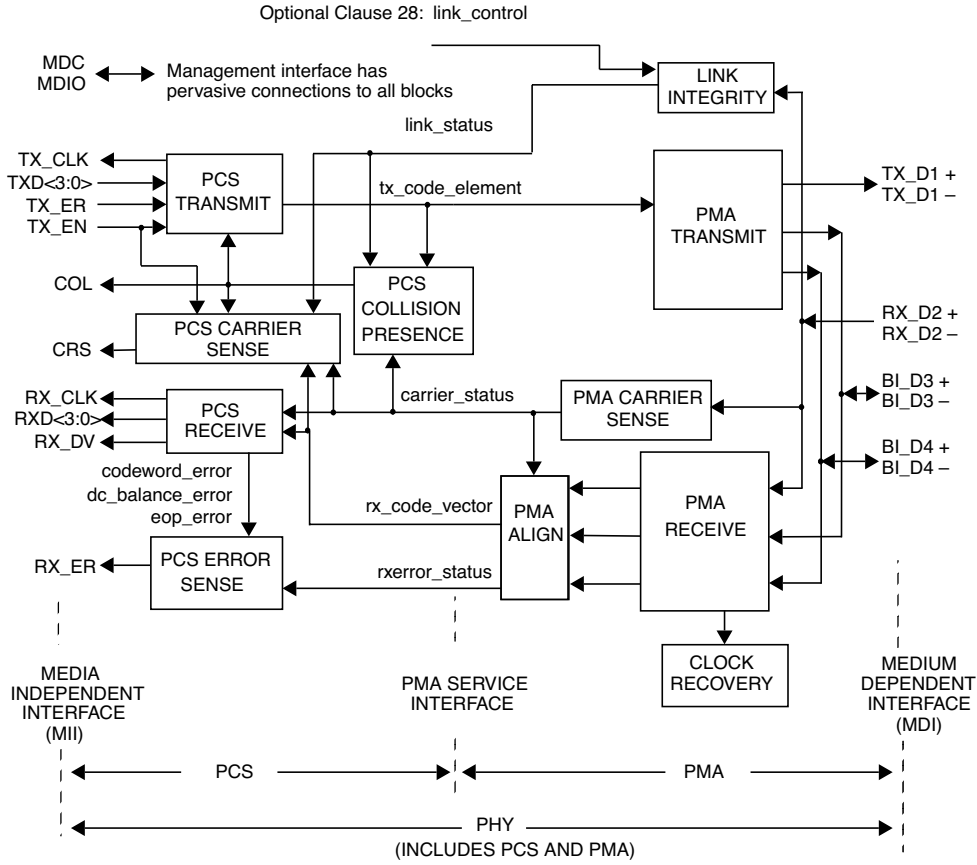
The 100BASE-T4 PCS couples a Media Independent Interface (MII), as described in Clause 22, to a Physical Medium Attachment sublayer (PMA).

The PCS Transmit function accepts data nibbles from the MII. The PCS Transmit function encodes these nibbles using an 8B6T coding scheme (to be described) and passes the resulting ternary symbols to the PMA. In the reverse direction, the PMA conveys received ternary symbols to the PCS Receive function. The PCS Receive function decodes them into octets, and then passes the octets one nibble at a time up to the MII. The PCS also contains a PCS Carrier Sense function, a PCS Error Sense function, a PCS Collision Presence function, and a management interface.

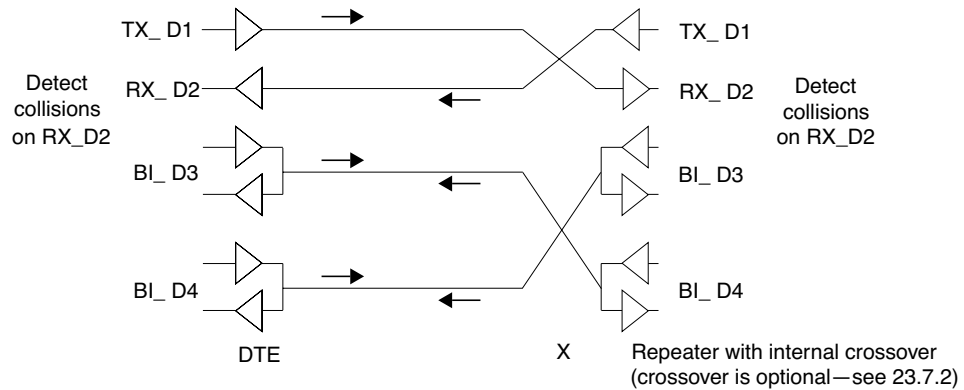
Figure 23-2 shows the division of responsibilities between the PCS, PMA, and MDI layers.

Physical level communication between PHY entities takes place over four twisted pairs. This specification permits the use of Category 3, 4, or 5 unshielded twisted pairs, installed according to ISO/IEC 11801: 1995, as specified in 23.6. Figure 23-3 shows how the PHY manages the four twisted pairs at its disposal.

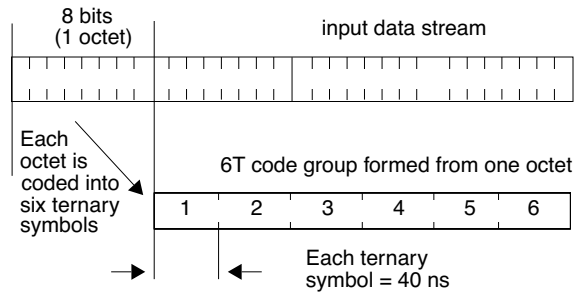
The 100BASE-T4 transmission algorithm always leaves one pair open for detecting carrier from the far end (see Figure 23-3). Leaving one pair open for carrier detection in each direction greatly simplifies media access control. All collision detection functions are accomplished using only the unidirectional pairs TX\_D1 and RX\_D2, in a manner similar to 10BASE-T. This collision detection strategy leaves three pairs in each direction free for data transmission, which uses an 8B6T block code, schematically represented in Figure 23-4.



**Figure 23-2—Division of responsibilities between 100BASE-T4 PCS and PMA**



**Figure 23-3—Use of wire pairs**



**Figure 23–4—8B6T coding**

8B6T coding, as used with 100BASE-T4 signaling, maps data octets into ternary symbols. Each octet is mapped to a pattern of 6 ternary symbols, called a 6T code group. The 6T code groups are fanned out to three independent serial channels. The effective data rate carried on each pair is one third of 100 Mb/s, which is 33.333... Mb/s. The ternary symbol transmission rate on each pair is 6/8 times 33.33 Mb/s, or precisely 25.000 MHz.

Refer to Annex 23A for a complete listing of 8B6T code words.

The PCS functions and state diagrams are specified in 23.2. The PCS electrical interface to the MII conforms to the interface requirements of Clause 21. The PCS interface to the PMA is an abstract message-passing interface specified in 23.3.

#### 23.1.4.2 Summary of physical medium attachment (PMA) specification

The PMA couples messages from the PMA service interface onto the twisted-pair physical medium. The PMA provides communications, at 100 Mb/s, over four pairs of twisted-pair wiring up to 100 m in length.

The PMA Transmit function, shown in Figure 23–2, comprises three independent ternary data transmitters. Upon receipt of a PMA\_UNITDATA.request message, the PMA synthesizes one ternary symbol on each of the three output channels (TX\_D1, BI\_D3, and BI\_D4). Each output driver has a *ternary* output, meaning that the output waveform can assume any of three values, corresponding to the transmission of ternary symbols CS0, CS1 or CS-1 (see 23.4.3.1) on each of the twisted pairs.

The PMA Receive function comprises three independent ternary data receivers. The receivers are responsible for acquiring clock, decoding the Start of Stream Delimiter (SSD) on each channel, and providing data to the PCS in the synchronous fashion defined by the PMA\_UNITDATA.indicate message. The PMA also contains functions for PMA Carrier Sense and Link Integrity.

PMA functions and state diagrams appear in 23.4. PMA electrical specifications appear in 23.5.

#### 23.1.5 Application of 100BASE-T4

##### 23.1.5.1 Compatibility considerations

All implementations of the twisted-pair link shall be compatible at the MDI. The PCS, PMA, and the medium are defined to provide compatibility among devices designed by different manufacturers. Designers are free to implement circuitry within the PCS and PMA (in an application-dependent manner) provided the MDI (and MII, when implemented) specifications are met.

### 23.1.5.2 Incorporating the 100BASE-T4 PHY into a DTE

The PCS is required when used with a DTE. The PCS provides functions necessary to the overall system operation (such as 8B6T coding) and cannot be omitted. Refer to Figure 23–1.

When the PHY is incorporated within the physical bounds of a DTE, conformance to the MII interface is optional, provided that the observable behavior of the resulting system is identical to a system with a full MII implementation. For example, an integrated PHY may incorporate an interface between PCS and MAC that is logically equivalent to the MII, but does not have the full output current drive capability called for in the MII specification.

### 23.1.5.3 Use of 100BASE-T4 PHY for point-to-point communication

The 100BASE-T4 PHY, in conjunction with the MAC specified in Clauses 1 through 4 (including parameterized values in 4.4.2.3 to support 100 Mb/s operation), may be used at both ends of a link for point-to-point applications between two DTEs. Such a configuration does not require a repeater. In this case each PHY may connect through an MII to its respective DTE. Optionally, either PHY (or both PHYs) may be incorporated into the DTEs without an exposed MII.

### 23.1.5.4 Support for Auto-Negotiation

The PMA service interface contains primitives used by the Auto-Negotiation algorithm (Clause 28) to automatically select operating modes when connected to a like device.

## 23.2 PCS functional specifications

The 100BASE-T4 PCS couples a Media Independent Interface (MII), as described in Clause 22, to a 100BASE-T4 Physical Medium Attachment sublayer (PMA).

At its interface with the MII, the PCS communicates via the electrical signals defined in Clause 22.

The interface between PCS and the next lower level (PMA) is an abstract message-passing interface described in 23.3. The physical realization of this interface is left to the implementor, provided the requirements of this standard, where applicable, are met.

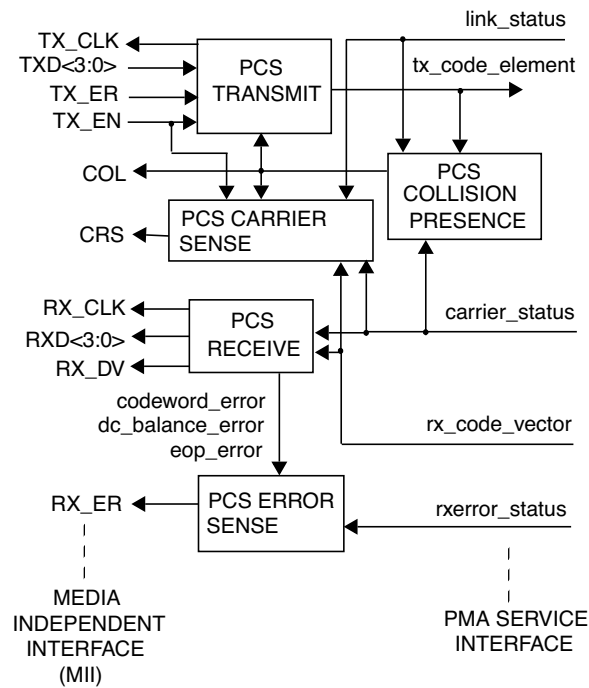
### 23.2.1 PCS functions

The PCS comprises one PCS Reset function and five simultaneous and asynchronous operating functions. The PCS operating functions are PCS Transmit, PCS Receive, PCS Error Sense, PCS Carrier Sense, and PCS Collision Presence. All operating functions start immediately after the successful completion of the PCS Reset function.

The PCS reference diagram, Figure 23–5, shows how the five operating functions relate to the messages of the PCS-PMA interface. Connections from the management interface (signals MDC and MDIO) to other layers are pervasive, and are not shown in Figure 23–5. The management functions are specified in Clause 30. See also Figure 23–6, which defines the structure of frames passed from PCS to PMA. See also Figure 23–7, which presents a reference model helpful for understanding the definitions of PCS Transmit function state variables *ohr1-4* and *tsr*.

#### 23.2.1.1 PCS Reset function

The PCS Reset function shall be executed any time either of two conditions occur. These two conditions are “power on” and the receipt of a reset request from the management entity. The PCS Reset function initializes



**Figure 23-5—PCS reference diagram**

all PCS functions. The PCS Reset function sets `pcs_reset=ON` for the duration of its reset function. All state diagrams take the open-ended `pcs_reset` branch upon execution of the PCS Reset function. The reference diagrams do not explicitly show the PCS Reset function.

### 23.2.1.2 PCS Transmit function

The PCS Transmit function shall conform to the PCS Transmit state diagram in Figure 23-8.

The PCS Transmit function receives nibbles from the TXD signals of the MII, assembles pairs of nibbles to form octets, converts the octets into 6T code groups according to the 8B6T code table, and passes the resulting ternary data to the PMA using the `PMA_UNITDATA.request` message. The state diagram of Figure 23-8 depicts the PCS Transmit function operation. Definitions of state variables `tsr`, `ohr`, `sosa`, `sosb`, `eop1-5`, and `tx_extend` used in that diagram, as well as in the following text, appear in 23.2.4.1. The physical structure represented in Figure 23-7 is not required; it merely serves to explain the meaning of the state diagram variables `ohr` and `tsr` in Figure 23-8. Implementors are free to construct any logical devices having functionality identical to that described by this functional description and the PCS Transmit state diagram, Figure 23-8.

PCS Transmit makes use of the `tsr` and `ohr` shift registers to manage nibble assembly and ternary symbol transmission. Nibbles from the MII go into `tsr`, which PCS Transmit reads as octets. PCS Transmit then encodes those octets and writes 6T code groups to the `ohr` registers. The `PMA_UNITDATA.request` message passes ternary symbols from the `ohr` registers to the PMA. In each state diagram block, the `ohr` loading operations are conducted first, then `tx_code_vector` is loaded and the state diagram waits 40 ns.

The first 5 octets assembled by the PCS Transmit function are encoded into the `sosa` code word and the next 3 octets assembled are encoded into the `sosb` code word. This guarantees that every packet begins with a valid preamble pattern. This is accomplished by the definition of `tsr`. In addition, the PCS Transmit state

diagram also specifies that at the start of a packet all three output holding registers ohr1, ohr3 and ohr4 will be loaded with the same value (sosa). This produces the ternary symbols labeled P3 and P4 in Figure 23–6

At the conclusion of the MAC frame, the PCS Transmit function appends eop1-5. This is accomplished by defining a variable tx\_extend to stretch the TX\_EN signal, and defining tsr during this time to be a sequence of constants that decodes to the proper eop code groups.

The encoding operation shall use the 8B6T code table listed in Annex 23A, and the dc balance encoding rules listed below. Encoding is performed separately for each transmit pair.

### 23.2.1.2.1 DC balance encoding rules

The encoding operation maintains dc balance on each transmit pair by keeping track of the cumulative weight of all 6T code groups (see *weight of 6T code group*, Annex 23A) transmitted on that pair. For each pair, it initiates the cumulative weight to 0 when the PCS Transmit function is in the AWAITING DATA TO TRANSMIT state. All 6T code groups in the code table have weight 0 or 1. The dc balance algorithm conditionally negates transmitted 6T code groups, so that the code weights transmitted on the line include 0, +1, and –1. This dc balance algorithm ensures that the cumulative weight on each pair at the conclusion of each 6T code group is always either 0 or 1, so only one bit per pair is needed to store the cumulative weight. As used below, the phrase “invert the cumulative weight bit” means “if the cumulative weight bit is zero then set it to one, otherwise set it to zero.”

After encoding any octet, except the constants sosa, sosb, eop1-5 or bad\_code, update the cumulative weight bit for the affected pair according to rules a) through c):

- a) If the 6T code group weight is 0, do not change the cumulative weight.
- b) If the 6T code group weight is 1, and the cumulative weight bit is 0, set the cumulative weight bit to 1.
- c) If the 6T code group weight is 1, and the cumulative weight bit is also 1, set the cumulative weight bit to 0, and then algebraically negate all the ternary symbol values in the 6T code group.

After encoding any of the constants sosa, sosb, or bad\_code, update the cumulative weight bit for the affected pair according to rule d):

- d) Do not change the cumulative weight. Never negate sosa, sosb or bad\_code.

After encoding any of the constants eop1-5, update the cumulative weight bit for the affected pair according to rules e) and f):

- e) If the cumulative weight is 0, do not change the cumulative weight; algebraically negate all the ternary symbol values in eop1-5.
- f) If the cumulative weight is 1, do not change the cumulative weight.

NOTE—The inversion rules for eop1-5 are opposite rule b). That makes eop1-5 look very unlike normal data, increasing the number of errors required to synthesize a false end-of-packet marker.

### 23.2.1.3 PCS Receive function

The PCS Receive function shall conform to the PCS Receive state diagram in Figure 23–9.

The PCS Receive function accepts ternary symbols from the PMA, communicated via the PMA\_UNITDATA.indicate message, converts them using 8B6T coding into a nibble-wide format and passes them up to the MII. This function also generates RX\_DV. The state diagram of Figure 23–9 depicts the PCS Receive function. Definitions of state variables ih2, ih3, and ih4 used in that diagram, as well as in the following text, appear in 23.2.4.1.

The last 6 values of the rx\_code\_vector are available to the decoder. PCS Receive makes use of these stored rx\_code\_vector values as well as the ih2-4 registers to manage the assembly of ternary symbols into 6T code groups, and the conversion of decoded data octets into nibbles. The last 6 ternary symbols for pair BI\_D3 (as extracted from the last 6 values of rx\_code\_vector) are referred to in the state diagram as BI\_D3[0:5]. Other pairs are referenced accordingly.

The PCS Receive state diagram starts the first time the PCS receives a PMA\_UNITDATA.indicate message with rx\_code\_vector=DATA (as opposed to IDLE or PREAMBLE). The contents of this first PMA\_UNITDATA.indicate (DATA) message are specified in 23.4.1.6.

After the sixth PMA\_UNITDATA.indicate (DATA) message (state DECODE CHANNEL 3), there is enough information to decode the first data octet. The decoded nibble is transmitted across the MII in two parts, a least significant nibble followed by a most significant nibble (see Clause 22).

During state COLLECT 4TH TERNARY SYMBOL the PCS Receive function raises RX\_DV and begins shifting out the nibbles of the 802.3<sup>®</sup> MAC SFD, least significant nibble first (SFD:LO). The most significant nibble of the 802.3<sup>®</sup> MAC SFD, called SFD:HI, is sent across the MII during the next state, COLLECT 5TH TERNARY SYMBOL.

Once eop is signaled by the decode operation, the state diagram de-asserts RX\_DV, preventing the end-of-packet bits from reaching the MII. At any time that RX\_DV is de-asserted, RXD<3:0> shall be all zeroes.

The decode operation shall use the 8B6T code table listed in Annex 23A, and the error-detecting rules listed in 23.2.1.3.1. Decoding and maintenance of the cumulative weight bit is performed separately for each receive pair.

### 23.2.1.3.1 Error-detecting rules

The decoding operation checks the dc balance on each receive pair by keeping track of the cumulative weight of all 6T code group received on that pair. For each pair, initialize the cumulative weight to 0 when the PCS Receive function is in the AWAITING INPUT state. As in the encoding operation, only one bit per pair is needed to store the cumulative weight.

Before decoding each octet, check the weight of the incoming code group and then apply rules a) through h) in sequence:

- a) If the received code group is eop1 (or its negation), set eop=ON. Then check the other pairs for conformance to the end-of-packet rules as follows: Check the last four ternary symbols of the next pair, and the last two ternary symbols from the third pair for exact conformance with the end-of-packet pattern specified by PCS Transmit, including the cumulative weight negation rules. If the received data does not conform, set the internal variable eop\_error=ON. Skip the other rules.
- b) If the received code group weight is greater than 1 or less than -1, set the internal variable dc\_balance\_error=ON. Decode to all zeros. Do not change the cumulative weight.
- c) If the received code group weight is zero, use the code table to decode. Do not change the cumulative weight.
- d) If the received code group weight is +1, and the cumulative weight bit is 0, use the code table to decode. Invert the cumulative weight bit.
- e) If the received code group weight is -1, and the cumulative weight bit is 1, algebraically negate each ternary symbol in the code group and then use the code table to decode. Invert the cumulative weight bit.
- f) If the received code group weight is +1 and the cumulative weight bit is 1, set the internal variable dc\_balance\_error=ON. Decode to all zeros. Do not change the cumulative weight.
- g) If the received code group weight is -1 and the cumulative weight bit is 0, set the internal variable dc\_balance\_error=ON. Decode to all zeros. Do not change the cumulative weight.

- h) If the (possibly negated) code group is not found in the code table, set `codeword_error=ON`. Decode to all zeros. Do not change the cumulative weight.

The variables `dc_balance_error`, `eop_error` and `codeword_error` shall remain OFF at all times other than those specified in the above error-detecting rules.

The `codeword_error=ON` indication for a (possibly negated) code group not found in the code table shall set `RX_ER` during the transfer of both affected data nibbles across the MII.

The `dc_balance_error=ON` indication for a code group shall set `RX_ER` during the transfer of both affected data nibbles across the MII.

The `eop_error=ON` indication shall set `RX_ER` during the transfer of the last decoded data nibble of the previous octet across the MII. That is at least one `RX_CLK` period earlier than the requirement for `codeword_error` and `dc_balance_error`.

These timing requirements imply consideration of implementation delays not specified in the PCS Receive state diagram.

`RX_DV` is asserted coincident with the transmission across the MII of valid packet data, including the Clause 4 MAC SFD, but not including the 100BASE-T4 end-of-packet delimiters `eop1-5`. When a packet is truncated due to early de-assertion of `carrier_status`, an `RX_ER` indication shall be generated and `RX_DV` shall be de-asserted, halting receive processing. The PCS Receive Function may use any of the existing signals `codeword_error`, `dc_balance_error`, or `eop_error` to accomplish this function.

#### 23.2.1.4 PCS Error Sense function

The PCS Error Sense function performs the task of sending `RX_ER` to the MII whenever `rxerror_status=ERROR` is received from the PMA sublayer or when any of the PCS decoding error conditions occur. The PCS Error Sense function shall conform to the PCS Error Sense state diagram in Figure 23–10.

Upon detection of any error, the error sense process shall report `RX_ER` to the MII before the last nibble of the Clause 4 MAC frame has been passed across the MII. Errors attributable to a particular octet are reported to the MII coincident with the octet in which they occurred.

The timing of `rxerror_status` shall cause `RX_ER` to appear on the MII no later than the last nibble of the first data octet in the frame.

#### 23.2.1.5 PCS Carrier Sense function

The PCS Carrier Sense function shall perform the function of controlling the MII signal `CRS` according to the rules presented in this clause.

While `link_status = OK`, `CRS` is asserted whenever `rx_crs=ON` or `TX_EN=1`, with timing as specified in 23.11.2, and Table 23-6.

#### 23.2.1.6 PCS Collision Presence function

A PCS collision is defined as the simultaneous occurrence of `tx_code_vector≠IDLE` and the assertion of `carrier_status=ON` while `link_status=OK`. While a PCS collision is detected, the MII signal `COL` shall be asserted, with timing as specified in 23.11.2 and Table 23–7.

At other times `COL` shall remain de-asserted.



## 23.2.2 PCS interfaces

### 23.2.2.1 PCS–MII interface signals

The following signals are formally defined in 22.2.2. Jabber detection as specified in 22.2.4.2.12 is not required by this standard.

**Table 23–1 – MII interface signals**

Signal name	Meaning
TX_CLK	Transmit Clock
TXD<3:0>	Transmit Data
TX_ER	Forces transmission of illegal code
TX_EN	Frames Transmit Data
COL	Collision Indication
CRS	Non-Idle Medium Indication
RX_CLK	Receive Clock
RXD<3:0>	Receive Data
RX_DV	Frames Receive SFD and DATA
RX_ER	Receive Error Indication
MDC	Management Data Clock
MDIO	Management Data

#### 23.2.2.2 PCS–Management entity signals

The management interface has pervasive connections to all functions. Operation of the management control lines MDC and MDIO, and requirements for managed objects inside the PCS and PMA, are specified in Clauses 22 and 30, respectively.

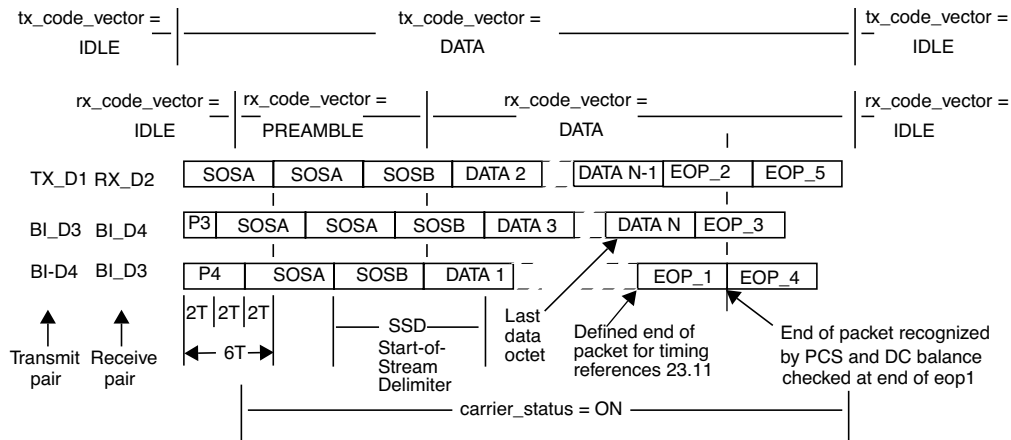
The loopback mode of operation shall be implemented in accordance with 22.2.4.1.2. The loopback mode of operation loops back transmit data to receive data, thus providing a way to check for the presence of a PHY.

No spurious signals shall be emitted onto the MDI when the PHY is held in power-down mode as defined in 22.2.4.1.5 (even if TX\_EN is ON) or when released from power-down mode, or when external power is first applied to the PHY.

#### 23.2.3 Frame structure

Frames passed from the PCS sublayer to the PMA sublayer shall have the structure shown in Figure 23–6. This figure shows how ternary symbols on the various pairs are synchronized as they are passed by the PMA\_UNITDATA.indicate and PMA\_UNITDATA.request messages. Time proceeds from left to right in the figure.

In the frame structure example, the last 6T code group, DATA N, happens to appear on transmit pair BI\_D3. It could have appeared on any of the three transmit pairs, with the five words eop1 through eop5 appended afterward as the next five octets in sequence. The end of packet as recognized by the PCS is defined as the end of the last ternary symbol of eop1. At this point a receiver has gathered enough information to locate the last word in the packet and check the dc balance on each pair.



**Figure 23-6—PCS sublayer to PMA sublayer frame structure**

If the PMA service interface is exposed, data carried between PCS and PMA by the PMA\_UNITDATA.indicate and PMA\_UNITDATA.request messages shall have a clock in each direction. Details of the clock implementation are left to the implementor. The choice of binary encoding for each ternary symbol is left to the implementor.

The following frame elements appear in Figure 23-6 (ternary symbols are transmitted leftmost first):

- SOSA The succession of six ternary symbols: [ 1 -1 1 -1 1 -1 ], which is the result of encoding the constant sosa.
- SOSB The succession of six ternary symbols: [ 1 -1 1 -1 -1 1 ], which is the result of encoding the constant sosb.
- P3 The succession of two ternary symbols: [ 1 -1 ].
- P4 The succession of four ternary symbols: [ 1 -1 1 -1 ].
- DATA A 6T code group that is the result of encoding a data octet in a packet that is not part of the Clause 4 MAC preamble or SFD.
- EOP1-5 A 6T code group that is the result of encoding one of the end-of-packet patterns eop1-5.

### 23.2.4 PCS state diagrams

The notation used in the state diagrams follows the conventions of 21.5. Transitions shown without source states are evaluated continuously and take immediate precedence over all other conditions.

#### 23.2.4.1 PCS state diagram constants

Register `tsr` may take on any of the nine constant values listed below (sosa through eop5, bad\_code, and zero\_code). These values are used to describe the functional operation of the coding process.

NOTE—Implementors are under no obligation to implement these constants in any particular way. For example, some implementors may choose to implement these codes as special flag bits attached to MII TXD nibble registers. Other implementors may choose to implement insertion of these codes on the downstream side of the coder function, using precoded 6T sequences.

All 6T code words are sent leftmost ternary symbol first.

sosa	A constant that encodes to:	[ 1 -1 1 -1 1 -1].
sosb	A constant that encodes to:	[ 1 -1 1 -1 -1 1].
eop1	A constant that encodes to:	[ 1 1 1 1 1 1].
eop2	A constant that encodes to:	[ 1 1 1 1 -1 -1].
eop3	A constant that encodes to:	[ 1 1 -1 -1 0 0].
eop4	A constant that encodes to:	[ -1 -1 -1 -1 -1 -1].
eop5	A constant that encodes to:	[ -1 -1 0 0 0 0].
bad_code	A constant that encodes to:	[ -1 -1 -1 1 1 1].
zero_code	A constant that encodes to:	[ 0 0 0 0 0 0].

### 23.2.4.2 PCS state diagram variables

codeword\_error

Indicates reception of invalid 6T code group.

Values: ON and OFF

Set by: PCS Receive; error-detecting rules

dc\_balance\_error

Indicates reception of dc coding violation.

Values: ON and OFF

Set by: PCS Receive; error-detecting rules

eop

Indicates reception of eop1.

A state variable set by the decoding operation. Reset to OFF when in PCS Receive state AWAITING INPUT. When the decoder detects eop1 on any pair, it sets this flag ON. The timing of eop shall be adjusted such that the last nibble of the last decoded data octet in a packet is the last nibble sent across the MII by the PMA Receive state diagram with RX\_DV set ON.

Values: ON and OFF

Set by: PCS Receive; error-detecting rules

eop\_error

Indicates reception of data with improper end-of-packet coding.

Values: ON and OFF

Set by: PCS Receive; error-detecting rules

ih2, ih4, and ih3 (input holding registers)

A set of holding registers used for the purpose of holding decoded data octets in preparation for sending across the MII one nibble at a time. One register is provided for each of the three receive pairs RX\_D2, BI\_D4, and BI\_D3, respectively.

Value: octet

Set by: PCS Receive

Each time the PCS Receive function decodes a 6T code group, it loads the result (an octet) into one of the ih2-4 registers. These three registers are loaded in round-robin fashion, one register being loaded every two ternary symbol times.

The PCS Receive state diagram reads nibbles as needed from the ih2-4 registers and stuffs them

into RXD.

ohr1, ohr3, and ohr4 (output holding registers)

(See Figure 23–7.) A set of shift registers used for the purpose of transferring coded 6T ternary symbol groups one ternary symbol at a time into the PMA. One register is provided for each of the three transmit pairs TX\_D1, BI\_D3, and BI\_D4, respectively.

Value: 6T code group. Each of the six cells holds one ternary symbol (i.e., –1, 0, or 1).

Set by: PCS Transmit

Each time the PCS Transmit function encodes a data octet, it loads the result (a 6T code group) into one of the ohr registers. Three registers are loaded in round-robin fashion, one register being loaded every two ternary symbol times. The PCS shall transmit octets on the three transmit pairs in round-robin fashion, in the order TX\_D1, BI\_D3, and BI\_D4, starting with TX\_D1.

The PMA\_UNITDATA.request (DATA) message picks the least significant (rightmost) ternary symbol from each ohr register and sends it to the PMA, as shown below. (Note that 6T code words in Annex 23A are listed with lsb on the left, not the right.)

tx\_code\_vector[TX\_D1] = the LSB of ohr1, also called ohr1[0]

tx\_code\_vector[BI\_D3] = the LSB of ohr3, also called ohr3[0]

tx\_code\_vector[BI\_D4] = the LSB of ohr4, also called ohr4[0]

After each PMA\_UNITDATA.request message, all three ohr registers shift right by one ternary symbol, shifting in zero from the left. The PCS Transmit function loads a new 6T code group into each ohr immediately after the last ternary symbol of the previous group is shifted out.

At the beginning of a preamble, the PCS Transmit function loads the same value (sosa) into all three output holding registers, which causes alternating transitions to immediately appear on all three output pairs. The result on pairs BI\_D3 and BI\_D4 is depicted by code words P3 and P4 in Figure 23–6.

pcs\_reset

Causes reset of all PCS functions when ON.

Values: ON and OFF

Set by: PCS Reset

rx\_crs

A latched asynchronous variable. Timing for the MII signal CRS is derived from rx\_crs.

Values: ON and OFF

Set ON when: carrier\_status changes to ON

Set OFF when either of two events occurs:  
carrier\_status changes to OFF, or  
detection of eop1, properly framed, on any of the lines RX\_D2, BI\_D4, or BI\_D3

Additionally, if, 20 ternary symbol times after rx\_crs falls, carrier\_status remains set to ON then set rx\_crs=ON.

NOTE—A special circuit for the detection of eop1 and subsequent de-assertion of rx\_crs, faster than the full 8B6T decoding circuits, is generally required to meet the timing requirements for CRS listed in 23.11.

## tsr (transmit shift register)

(See Figure 23–7.) A shift register defined for the purpose of assembling nibbles from the MII TXD into octets.

Values:	The variable tsr always contains both the current nibble of TXD and the previous nibble of TXD. Valid values for tsr therefore include all octets. Register tsr may also take on any of the nine constant values listed in 23.2.4.1.
Nibble order:	When encoding the tsr octet, the previous TXD nibble is considered the least significant nibble.
Set by:	PCS Transmit

During the first 16 TX\_CLK cycles after TX\_EN is asserted, tsr shall assume the following values in sequence regardless of TXD: sosa, sosa, sosa, sosa, sosa, sosa, sosa, sosa, sosa, sosa, sosa, sosa, sosb, sosb, sosb, sosb, sosb. This action substitutes the 100BASE-T4 preamble for the Clause 4 MAC preamble. The PCS Transmit state diagram samples the tsr only every other clock, which reduces the number of sosa and sosb constants actually coded to 5 and 3, respectively.

During the first 10 TX\_CLK cycles after TX\_EN is de-asserted, tsr shall assume the following values in sequence, regardless of TXD: eop1, eop1, eop2, eop2, eop3, eop3, eop4, eop4, eop5, eop5. This action appends the 100BASE-T4 end-of-packet delimiter to each pair. The PCS Transmit state diagram samples the tsr only every other clock, which reduces the number of eop1-5 constants actually coded to 1 each.

Except for the first 16 TX\_CLK cycles after TX\_EN is asserted, any time TX\_ER and TX\_EN are asserted, tsr shall assume the value bad\_code with such timing as to cause both nibbles of the affected octet to be encoded as bad\_code. If TX\_ER is asserted at any time during the first 16 TX\_CLK cycles after TX\_EN is asserted, tsr shall during the 17th and 18th clock cycles assume the value bad\_code.

If TX\_EN is de-asserted on an odd nibble boundary, the PCS shall extend TX\_EN by one TX\_CLK cycle, and behave as if TX\_ER were asserted during that additional cycle.

Except for the first 10 TX\_CLK cycles after TX\_EN is de-asserted, any time TX\_EN is not asserted, tsr shall assume the value zero\_code.

## tx\_extend

A latched, asynchronous state variable used to extend the TX\_EN signal long enough to ensure complete transmission of all nonzero ternary symbols in eop1-5.

Values:	ON and OFF
Set ON upon:	rising edge of TX_EN
Set OFF upon:	either of two conditions: <ol style="list-style-type: none"> <li>In the event of a collision (COL is asserted at any time during transmission) set tx_extend=OFF when TX_EN de-asserts.</li> <li>In the event of no collision (COL remains de-asserted throughout transmission) set tx_extend=OFF upon completion of transmission of last ternary symbol in eop4.</li> </ol>

NOTE 1—The 6T code group eop5 has four zeroes at the end. The 6T code group eop4 contains the last nonzero ternary symbol to be transmitted.

NOTE 2—The effect of a collision, if present, is to truncate the frame at the original boundary determined by TX\_EN. Noncolliding frames are extended, while colliding frames are not.

### 23.2.4.3 PCS state diagram timer

tw1\_timer

A continuous free-running timer.

Values: The condition tw1\_timer\_done goes true when the timer expires.

Restart when: Immediately after expiration (restarting the timer resets condition tw1\_timer\_done).

Duration: 40 ns nominal.

TX\_CLK shall be generated synchronous to tw1\_timer (see tolerance required for TX\_CLK in 23.5.1.2.10).

On every occurrence of tw1\_timer\_done, the state diagram advances by one block. The message PMA\_UNITDATA.request is issued concurrent with tw1\_timer\_done.

### 23.2.4.4 PCS state diagram functions

encode()

The encode operation of 23.2.1.2.

Argument: octet

Returns: 6T code group

decode()

The decode operation of 23.2.1.3.

Argument: 6T code group

Returns: octet

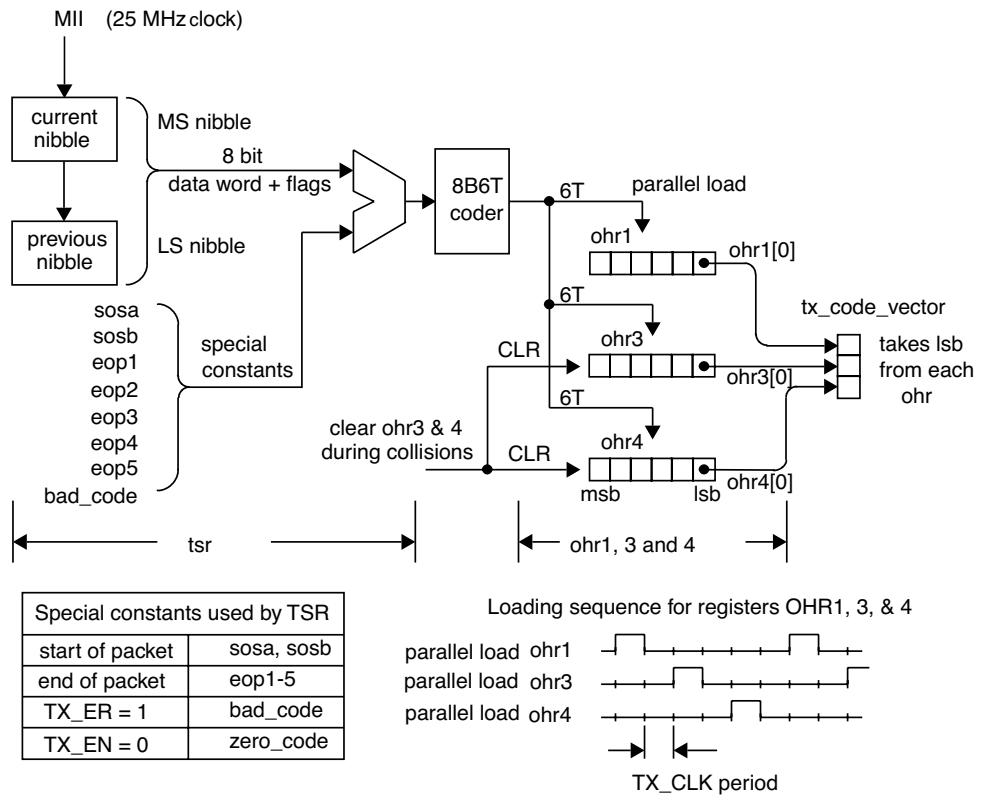


Figure 23-7—PCS Transmit reference diagram

23.2.4.5 PCS state diagrams

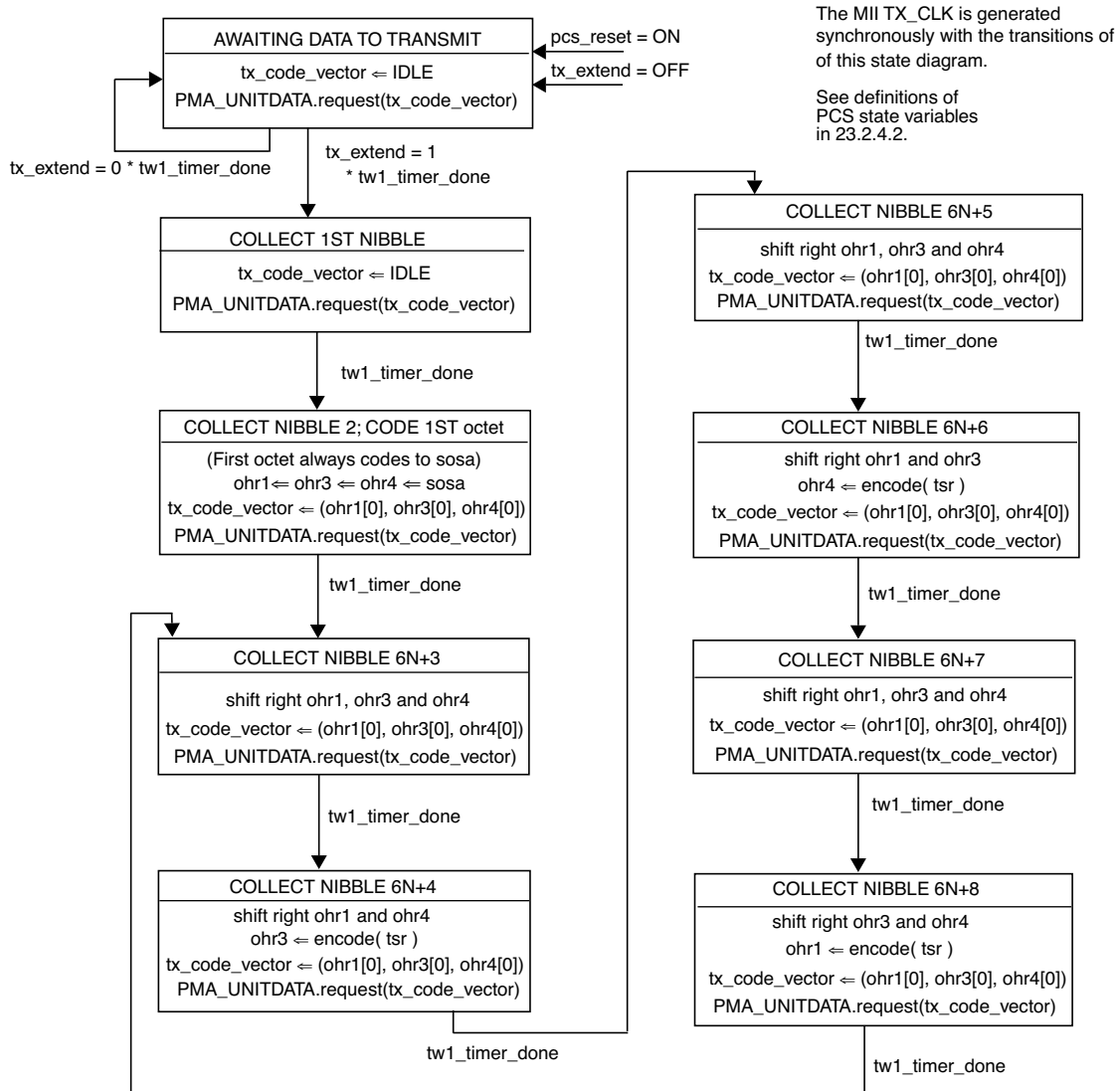


Figure 23–8—PCS Transmit state diagram



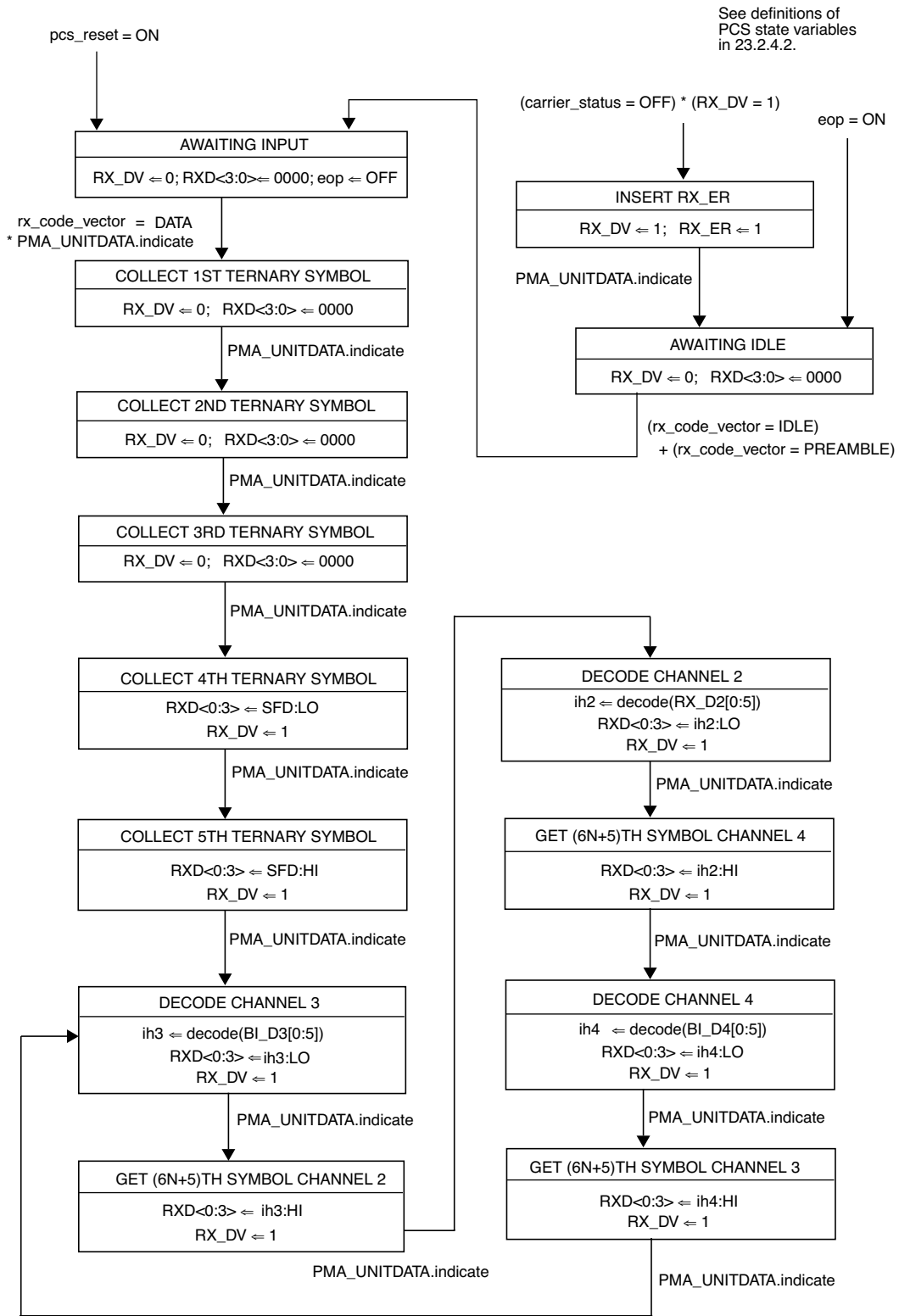
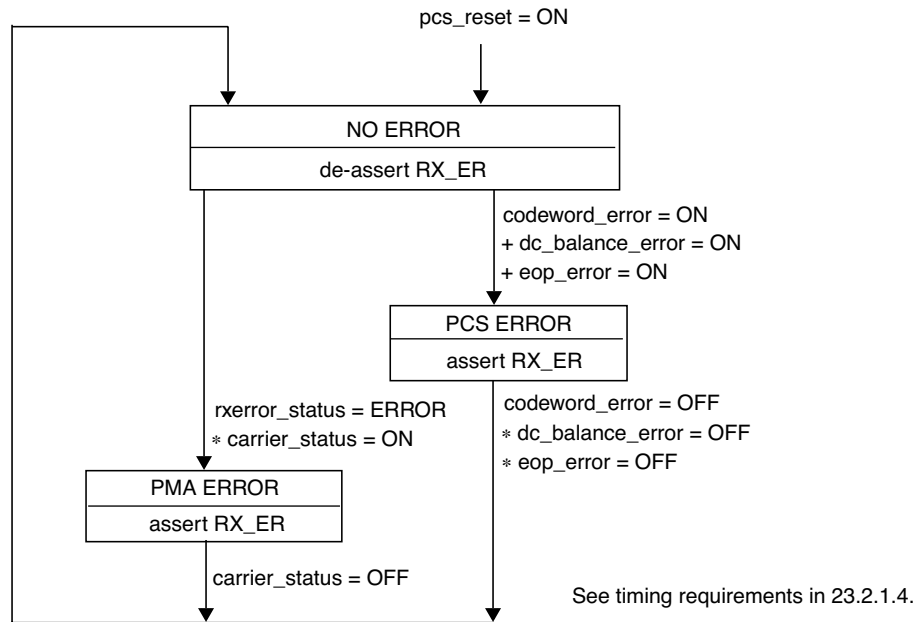


Figure 23-9—PCS Receive state diagram



**Figure 23–10—PCS Error Sense state diagram**

### 23.2.5 PCS electrical specifications

The interface between PCS and PMA is an abstract message-passing interface, having no specified electrical properties.

Electrical characteristics of the signals passing between the PCS and MII may be found in Clause 22.

### 23.3 PMA service interface

This clause specifies the services provided by the PMA to either the PCS or a Repeater client. These services are described in an abstract manner and do not imply any particular implementation.

The PMA Service Interface supports the exchange of code vectors between the PMA and its client (either the PCS or a Repeater). The PMA also generates status indications for use by the client.

The following primitives are defined:

- PMA\_TYPE.indicate
- PMA\_UNITDATA.request
- PMA\_UNITDATA.indicate
- PMA\_CARRIER.indicate
- PMA\_LINK.indicate
- PMA\_LINK.request
- PMA\_RXERROR.indicate

### 23.3.1 PMA\_TYPE.indicate

This primitive is generated by the PMA to indicate the nature of the PMA instantiation. The purpose of this primitive is to allow clients to support connections to the various types of 100BASE-T PMA entities in a generalized manner.

#### 23.3.1.1 Semantics of the service primitive

PMA\_TYPE.indicate (pma\_type)

The pma\_type parameter for use with the 100BASE-T4 PMA is T4.

#### 23.3.1.2 When generated

The PMA shall continuously generate this primitive to indicate the value of pma\_type.

#### 23.3.1.3 Effect of receipt

The client uses the value of pma\_type to define the semantics of the PMA\_UNITDATA.request and PMA\_UNITDATA.indicate primitives.

### 23.3.2 PMA\_UNITDATA.request

This primitive defines the transfer of data (in the form of tx\_code\_vector parameters) from the PCS or repeater to the PMA.

#### 23.3.2.1 Semantics of the service primitive

PMA\_UNITDATA.request (tx\_code\_vector)

When transmitting data using 100BASE-T4 signaling, the PMA\_UNITDATA.request conveys to the PMA simultaneously the logical output value for each of the three transmit pairs TX\_D1, BI\_D3, and BI\_D4. The value of tx\_code\_vector during data transmission is therefore a three-element vector, with one element corresponding to each output pair. Each of the three elements of the tx\_code\_vector may take on one of three logical values: 1, 0, or -1, corresponding to the three ternary possibilities +, 0, and - listed for each ternary symbol in the 8B6T code table (see Annex 23A).

Between packets, the 100BASE-T4 PMA layer sends the 100BASE-T4 idle signal, TP\_IDL\_100. The PCS informs the PMA layer that it is between packets, thus enabling the PMA idle signal, by setting the tx\_code\_vector parameter to IDLE.

For pma\_type 100BASE-T4, the tx\_code\_vector parameter can take on either of two forms:

IDLE	A single value indicating to the PMA that there is no data to convey. The PMA generates link integrity pulses during the time that tx_code_vector = IDLE.
DATA	A vector of three ternary symbols, one for each of the three transmit pairs TX_D1, BI_D3, and BI_D4. The ternary symbol for each pair may take on one of three values, 1, 0, or -1.

The ternary symbols comprising tx\_code\_vector, when they are conveyed using the DATA format, are called, according to the pair on which each will be transmitted, tx\_code\_vector[BI\_D4], tx\_code\_vector[TX\_D1], and tx\_code\_vector[BI\_D3].

### 23.3.2.2 When generated

The PCS or Repeater client generates PMA\_UNITDATA.request synchronous with every MII TX\_CLK.

For the purposes of state diagram descriptions, it may be assumed that at the time PMA\_UNITDATA.request is generated, the MII signals TX\_EN, and TX\_ER, and TXD instantly become valid and that they retain their values until the next PMA\_UNITDATA.request.

In the state diagrams, PMA\_UNITDATA.request is assumed to occur at the conclusion of each tw1 wait function.

### 23.3.2.3 Effect of receipt

Upon receipt of this primitive, the PMA transmits the indicated ternary symbols on the MDI.

## 23.3.3 PMA\_UNITDATA.indicate

This primitive defines the transfer of data (in the form of rx\_code\_vector parameters) from the PMA to the PCS or repeater during the time that link\_status=OK.

### 23.3.3.1 Semantics of the service primitive

PMA\_UNITDATA.indicate (rx\_code\_vector)

When receiving data using 100BASE-T4 signaling, the PMA\_UNITDATA.indicate conveys to the PCS simultaneously the logical input value for each of the three receive pairs RX\_D2, BI\_D4, and BI\_D3. The value of rx\_code\_vector during data reception is therefore a three-element vector, with one element corresponding to each input pair. Each of the three elements of the rx\_code\_vector may take on one of three logical values: 1, 0, or -1, corresponding to the three ternary possibilities +, 0, and - listed for each ternary symbol in the 8B6T code table (see Annex 23A).

Between packets, the rx\_code\_vector is set by the PMA to the value IDLE.

From the time the PMA asserts carrier\_status=ON until the PMA recognizes the SSD pattern (not all of the pattern need be received in order for the PMA to recognize the pattern), the PMA sets rx\_code\_vector to the value PREAMBLE.

For pma\_type 100BASE-T4, the rx\_code\_vector parameter can take on any of three forms:

IDLE	A single value indicating that the PMA has no data to convey.
PREAMBLE	A single value indicating that the PMA has detected carrier, but has not received a valid SSD.
DATA	A vector of three ternary symbols, one for each of the three receive pairs RX_D2, BI_D3, and BI_D4. The ternary symbol for each pair may take on one of three values, 1, 0, or -1.

The ternary symbols comprising rx\_code\_vector, when they are conveyed using the DATA format, are called, according to the pair upon which each symbol was received, rx\_code\_vector[BI\_D3], rx\_code\_vector[RX\_D2], and rx\_code\_vector[BI\_D4].

### 23.3.3.2 When generated

The PMA shall generate PMA\_UNITDATA.indicate (DATA) messages synchronous with data received at the MDI.

**23.3.3.3 Effect of receipt**

The effect of receipt of this primitive is unspecified.

**23.3.4 PMA\_CARRIER.indicate**

This primitive is generated by the PMA to indicate the status of the signal being received from the MDI. The purpose of this primitive is to give the PCS or repeater client the earliest reliable indication of activity on the underlying medium.

**23.3.4.1 Semantics of the service primitive**

PMA\_CARRIER.indicate (carrier\_status)

The carrier\_status parameter can take on one of two values: OFF or ON, indicating whether the incoming signal should be interpreted as being between packets (OFF) or as a packet in progress (ON).

**23.3.4.2 When generated**

The PMA shall generate this primitive to indicate the value of carrier\_status.

**23.3.4.3 Effect of receipt**

The effect of receipt of this primitive is unspecified.

**23.3.5 PMA\_LINK.indicate**

This primitive is generated by the PMA to indicate the status of the underlying medium. The purpose of this primitive is to give the PCS or repeater client or Auto-Negotiation algorithm a means of determining the validity of received code elements.

**23.3.5.1 Semantics of the service primitive**

PMA\_LINK.indicate (link\_status)

The link\_status parameter can take on one of three values: FAIL, READY, or OK:

FAIL	The link integrity function does not detect a valid 100BASE-T4 link.
READY	The link integrity function detects a valid 100BASE-T4 link, but has not been enabled by Auto-Negotiation.
OK	The 100BASE-T4 link integrity function detects a valid 100BASE-T4 link, and has been enabled by Auto-Negotiation.

**23.3.5.2 When generated**

The PMA shall generate this primitive to indicate the value of link\_status.

**23.3.5.3 Effect of receipt**

The effect of receipt of this primitive is unspecified.

### 23.3.6 PMA\_LINK.request

This primitive is generated by the Auto-Negotiation algorithm. The purpose of this primitive is to allow the Auto-Negotiation algorithm to enable and disable operation of the PHY.

#### 23.3.6.1 Semantics of the service primitive

PMA\_LINK.request (link\_control)

The link\_control parameter can take on one of three values: SCAN\_FOR\_CARRIER, DISABLE, or ENABLE.

SCAN_FOR_CARRIER	Used by the Auto-Negotiation algorithm prior to receiving any fast link pulses. During this mode the PHY reports link_status=READY if it recognizes 100BASE-T4 carrier from the far end, but no other actions are enabled.
DISABLE	Used by the Auto-Negotiation algorithm to disable PHY processing in the event fast link pulses are detected. This gives the Auto-Negotiation algorithm a chance to determine how to configure the link.
ENABLE	Used by Auto-Negotiation to turn control over to the PHY for data processing functions. This is the default mode if Auto-Negotiation is not present.

#### 23.3.6.2 Default value of parameter link\_control

Upon power-on, reset, or release from power-down, the link\_control parameter shall revert to ENABLE. If the optional Auto-Negotiation algorithm is not implemented, no PMA\_LINK.request message will arrive and the PHY will operate indefinitely with link\_control=ENABLE.

#### 23.3.6.3 When generated

The Auto-Negotiation algorithm generates this primitive to indicate to the PHY how to behave.

Upon power-on, reset, or release from power down, the Auto-Negotiation algorithm, if present, issues the message PMA\_LINK.request (SCAN\_FOR\_CARRIER).

#### 23.3.6.4 Effect of receipt

Whenever link\_control=SCAN\_FOR\_CARRIER, the PHY shall enable the Link Integrity state diagram, but block passage into the state LINK\_PASS, while holding rcv=DISABLE, and xmit=DISABLE. While link\_control=SCAN\_FOR\_CARRIER, the PHY shall report link\_status=READY if it recognizes 100BASE-T4 link integrity pulses coming from the far end, otherwise it reports link\_status=FAIL.

Whenever link\_control=DISABLE, the PHY shall report link\_status=FAIL and hold the Link Integrity state diagram in the RESET state, while holding rcv=disable and xmit=DISABLE.

While link\_control=ENABLE, the PHY shall allow the Link Integrity function to determine if the link is available and, if so, set rcv=ENABLE and xmit=ENABLE.

### 23.3.7 PMA\_RXERROR.indicate

The primitive is generated in the PMA by the PMA Align function to indicate the status of the signal being received from the MDI. The purpose of this primitive is to give the PCS or repeater client an indication of a PMA detectable receive error.

#### 23.3.7.1 Semantics of the service primitive

PMA\_RXERROR.indicate (rxerror\_status)

The rxerror\_status parameter can take on one of two values: ERROR or NO\_ERROR, indicating whether the incoming signal contains a detectable error (ERROR) or not (NO\_ERROR).

#### 23.3.7.2 When generated

The PMA shall generate this primitive to indicate whether or not each incoming packet contains a PMA detectable error (23.2.1.4).

#### 23.3.7.3 Effect of receipt

The effect of receipt of this primitive is unspecified.

## 23.4 PMA functional specifications

The PMA couples messages from a PMA service interface (23.3) to the 100BASE-T4 baseband medium (23.6).

The interface between PCS and the baseband medium is the Medium Dependent Interface (MDI), specified in 23.7.

### 23.4.1 PMA functions

The PMA sublayer comprises one PMA Reset function and six simultaneous and asynchronous operating functions. The PMA operating functions are PMA Transmit, PMA Receive, PMA Carrier Sense, Link Integrity, PMA Align, and Clock Recovery. All operating functions are started immediately after the successful completion of the PMA Reset function. When the PMA is used in conjunction with a PCS, the RESET function may be shared between layers.

The PMA reference diagram, Figure 23–11, shows how the operating functions relate to the messages of the PMA Service interface and the signals of the MDI. Connections from the management interface, comprising the signals MDC and MDIO, to other layers are pervasive, and are not shown in Figure 23–11. The Management Interface and its functions are specified in Clause 22.

#### 23.4.1.1 PMA Reset function

The PMA Reset function shall be executed any time either of two conditions occur. These two conditions are power-on and the receipt of a reset request from the management entity. The PMA Reset function initializes all PMA functions. The PMA Reset function sets pma\_reset=ON for the duration of its reset function. All state diagrams take the open-ended pma\_reset branch upon execution of the PMA Reset function. The reference diagrams do not explicitly show the PMA Reset function.

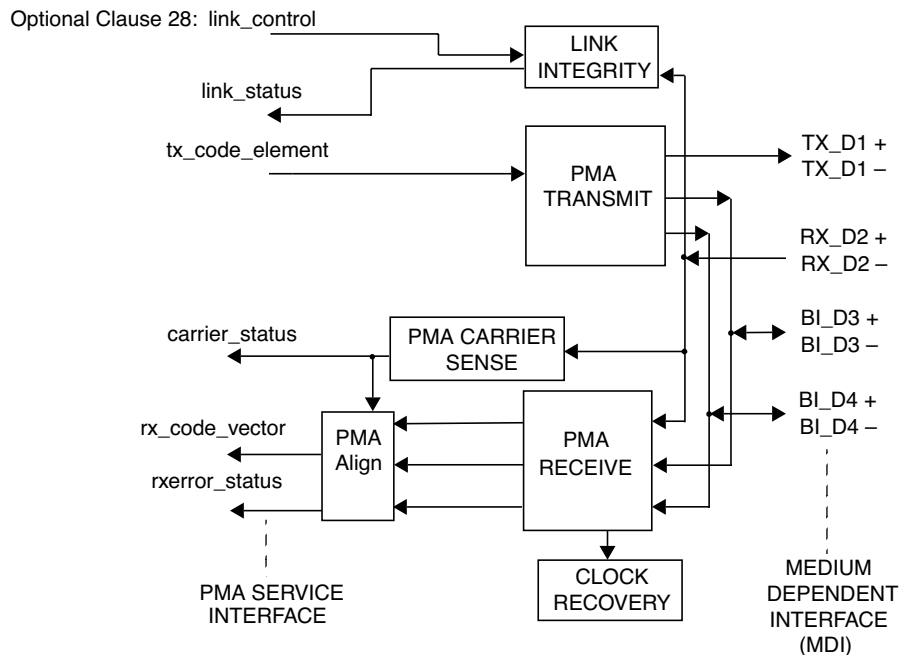


Figure 23-11—PMA reference diagram

### 23.4.1.2 PMA Transmit function

Except as provided for in the next paragraph, whenever  $(tx\_code\_vector=DATA) \times (pma\_carrier=OFF)$ , the PMA shall transmit onto the MDI ternary symbols on pairs TX\_D1, BI\_D3, and BI\_D4 equal to  $tx\_code\_vector[TX\_D1]$ ,  $tx\_code\_vector[BI\_D3]$ , and  $tx\_code\_vector[BI\_D4]$ , respectively.

Whenever  $(tx\_code\_vector=DATA) \times (pma\_carrier=ON)$ , the PMA shall transmit onto the MDI ternary symbols on pairs TX\_D1, BI\_D3, and BI\_D4 equal to  $tx\_code\_vector[TX\_D1]$ , CS0, and CS0, respectively, and continue doing so until  $tx\_code\_vector=IDLE$ .

NOTE—This shuts off the transmitters on channels BI\_D3 and BI\_D4, and keeps them off, in the event of a collision. Shutting off the transmitters prevents overload and saturation of the transmitters, and also reduces the amount of near-end crosstalk present while monitoring for the end of carrier.

Whenever  $tx\_code\_vector=IDLE$ , an idle signal shall be transmitted on pair TX\_D1 and silence on pairs BI\_D3 and BI\_D4. The idle signal consists of periods of silence (times where the differential output voltage remains at  $0\text{ mV} \pm 50\text{ mV}$ ) broken by the transmission of link integrity test pulses.

The 100BASE-T4 idle signal is similar to the 10BASE-T idle signal, but with 100BASE-T4 ternary signal levels and a faster repetition rate. The 100BASE-T4 idle signal is called TP\_IDL\_100. The TP\_IDL\_100 signal shall be a repeating sequence formed from one  $1.2\text{ ms} \pm 0.6\text{ ms}$  period of silence (the time where the differential voltage remains at  $0\text{ mV} \pm 50\text{ mV}$ ) and one link test pulse. Each link test pulse shall be a succession of two ternary symbols having logical values of  $-1$  and  $1$  transmitted on pair TX\_D1 using CS-1 and CS1 as defined in 23.4.3.1. Following a packet, the TP\_IDL\_100 shall start with a period of silence.

Transmission of TP\_IDL\_100 may be terminated at any time with respect to the link test pulse. It shall be terminated such that ternary symbols of the subsequent packet are not corrupted, and are not delayed any more than is specified in 23.11.



For any link test pulse occurring within 20 ternary symbol times of the beginning of a preamble, the zero crossing jitter (as defined in 23.5.1.2.5) of the link test pulse when measured along with the zero crossings of the preamble shall be less than 4 ns p-p.

NOTE—The above condition allows clock recovery implementations that optionally begin fast-lock sequences on part of a link integrity pulse to properly acquire lock on a subsequent preamble sequence.

Regardless of other considerations, when the transmitter is disabled (xmit=DISABLE), the PMA Transmit function shall transmit the TP\_IDL\_100 signal.

### 23.4.1.3 PMA Receive function

PMA Receive contains the circuits necessary to convert physically encoded ternary symbols from the physical MDI receive pairs (RX\_D2, BI\_D3 and BI\_D4) into a logical format suitable for the PMA Align function. Each receive pair has its own dedicated PMA Receive circuitry.

The PHY shall receive the signals on the receive pairs (RX\_D2, BI\_D3, and BI\_D4) and translate them into one of the PMA\_UNITDATA.indicate parameters IDLE, PREAMBLE, or DATA with a ternary symbol error rate of less than one part in  $10^8$ .

If both pma\_carrier=ON and tx\_code\_vector=DATA, the value of rx\_code\_vector is unspecified until pma\_carrier=OFF.

### 23.4.1.4 PMA Carrier Sense function

The PMA Carrier Sense function shall set pma\_carrier=ON upon reception of the following pattern on pair RX\_D2 at the receiving MDI, as measured using a 100BASE-T4 transmit test filter (23.5.1.2.3):

Any signal greater than 467 mV, followed by any signal less than –225 mV, followed by any signal greater than 467 mV, all three events occurring within 2 ternary symbol times.

The operation of carrier sense is undefined for signal amplitudes greater than 4.5 V.

See 23.5.1.3.2 for a list of signals defined *not* to set pma\_carrier=ON.

After asserting pma\_carrier=ON, PMA Carrier Sense shall set pma\_carrier=OFF upon receiving either of these conditions:

- a) Seven consecutive ternary symbols of value CS0 on pair RX\_D2.
- b) (tx\_code\_vector=DATA) has not been true at any time since pma\_carrier was asserted, *and* the 6T code group eop1 has been received, properly framed, on any of the lines RX\_D2, BI\_D4, or BI\_D3, *and* enough time has passed to assure passage of all ternary symbols of eop4 across the PMA service interface.

NOTE—Designers may wish to take advantage of the fact that the minimum received packet fragment will include at least 24 ternary symbols of data on pair RX\_D2. Therefore, once carrier is activated, it is not necessary to begin searching for seven consecutive zeroes until after the 24th ternary symbol has been received. During the time that the first 24 ternary symbols are being received, the near-end crosstalk from pairs BI\_D3 and BI\_D4, which are switched off during collisions, decays substantially.

While rcv=ENABLE, the PMA CARRIER function shall set carrier\_status = pma\_carrier.

While rcv≠ENABLE, the PMA CARRIER function shall set carrier\_status = OFF.

This function operates independently of the Link Integrity function.

### 23.4.1.5 Link Integrity function

Link Integrity provides the ability to protect the network from the consequences of failure of the simplex link attached to RX\_D2. While such a failure is present, transfer of data by the Transmit and Receive functions is disabled.

Link Integrity observes the incoming wire pair, RX\_D2, to determine whether the device connected to the far end is of type 100BASE-T4. Based on its observations, Link Integrity sets two important internal variables:

- a) `pma_type` variable is set to 100BASE-T4.
- b) `link_status` variable is a parameter sent across the PMA Service interface.

The Link Integrity function shall comply with the state diagram of Figure 23–12.

Four conditions gate the progression of states toward LINK\_PASS: (1) reception of at least 31 link integrity test pulses; (2) reception of at least 96 more link integrity test pulses, or reception of carrier; (3) cessation of carrier, if it was present; (4) detection of equals link\_control ENABLE.

While the PMA is not in the LINK\_PASS state, the Link Integrity function sets `rcv=DISABLE` and `xmit=DISABLE`, thus disabling the bit transfer of the Transmit and Receive functions.

If a visible indicator is provided on the PHY to indicate the link status, it is recommended that the color be green and that the indicator be labeled appropriately. It is further recommended that the indicator be on when the PHY is in the LINK\_PASS state and off otherwise.

### 23.4.1.6 PMA Align function

The PMA Align function accepts received ternary symbols from the PMA Receive function, along with `pma_carrier`. PMA Align is responsible for realigning the received ternary symbols to eliminate the effects of unequal pair propagation time, commonly called pair skew. PMA Align also looks for the SSD pattern to determine the proper alignment of 6T code groups, and then forwards PMA\_UNITDATA.indicate (DATA) messages to the PCS. The SSD pattern includes referencing patterns on each of the three receive lines that may be used to establish the proper relationship of received ternary symbols (see Figure 23–6).

NOTE—The skew between lines is not expected to change measurably from packet to packet.

At the beginning of each received frame, the PMA Carrier Sense function asserts `pma_carrier=ON`. During the preamble, the Clock Recovery function begins synchronizing its receive clock. Until clock is synchronized, data coming from the low-level PMA Receive function is meaningless. The PMA Align function is responsible for waiting for the receiver clock to stabilize and then properly recognizing the 100BASE-T4 coded SSD pattern. The PMA Align function shall send PMA\_UNITDATA.indicate (PREAMBLE) messages to the PCS from the time `pma_carrier=ON` is asserted until the PMA is ready to transfer the first PMA\_UNITDATA.indicate (DATA) message. Once the PMA Align function locates a SSD pattern, it begins forwarding PMA\_UNITDATA.indicate (DATA) messages to the PCS, starting with the first ternary symbol of the first data word on pair BI\_D3, as defined in Figure 23–6. This first PMA\_UNITDATA.indicate (DATA) message shall transfer the following ternary symbols, as specified in the frame structure diagram, Figure 23–6:

- `rx_code_vector[BI_D3]` first ternary symbol of first data code group
- `rx_code_vector[RX_D2]` second ternary symbol prior to start of second data code group
- `rx_code_vector[BI_D4]` fourth ternary symbol prior to start of third data code group

PMA Align shall continue sending PMA\_UNITDATA.indicate (DATA) messages until pma\_carrier=OFF. While pma\_carrier=OFF, PMA Align shall emit PMA\_UNITDATA.indicate (IDLE) messages.

If no valid SSD pattern is recognized within 22 ternary symbol times of the assertion of pma\_carrier=ON, the PMA Align function shall set rxerror\_status=ERROR. The PMA Align function is permitted to begin sending PMA\_UNITDATA.indicate (DATA) messages upon receipt of a partially recognized SSD pattern, but it is required to set rxerror\_status=ERROR if the complete SSD does not match perfectly the expected ternary symbol sequence. Rxerror\_status shall be reset to NO\_ERROR when pma\_carrier=OFF.

The PMA Align function is permitted to use the first received packet of at least minimum size after RESET or the transition to LINK\_PASS to learn the nominal skew between pairs, adjust its equalizer, or perform any other initiation functions. During this first packet, the PMA Align function shall emit PMA\_UNITDATA.indicate (PREAMBLE) messages, but may optionally choose to never begin sending PMA\_UNITDATA.indicate (DATA) messages.

The PMA Align function shall tolerate a maximum skew between any two pairs of 60 ns in either direction without error.

To protect the network against the consequences of mistaken packet framing, the PMA Align function shall detect the following error and report it by setting rxerror\_status=ERROR (optionally, those error patterns already detected by codeword\_error, dc\_balance\_error, or eop\_error do not also have to be detected by rxerror\_status): *In a series of good packets, any one packet that has been corrupted with three or fewer ternary symbols in error causing its sosb 6T code groups on one or more pairs to appear in the wrong location.*

Several approaches are available for meeting this requirement, including, but not limited to, a) comparing the relative positions of sosb 6T code groups on successive packets; b) measuring the time between the first preamble pulse and reception of sosb on each pair; c) counting the number of zero crossings from the beginning of the preamble until sosb; and d) monitoring for exception strings like “11” and “-1-1-1” in conjunction with one or more of the above techniques.

Regardless of other considerations, when the receive function is disabled (rcv=DISABLE), the PMA Align function shall emit PMA\_UNITDATA.indicate (IDLE) messages and no others.

#### **23.4.1.7 Clock Recovery function**

The Clock Recovery function couples to all three receive pairs. It provides a synchronous clock for sampling each pair. While it may not drive the MII directly, the Clock Recovery function is the underlying root source of RX\_CLK.

The Clock Recovery function shall provide a clock suitable for synchronously decoding ternary symbols on each line within the bit error tolerance provided in 23.4.1.3. During each preamble, in order to properly recognize the frame delimiting pattern formed by code word sosb on each pair, the received clock signal must be stable and ready for use in time to decode the following ternary symbols: the 16th ternary symbol of pair RX\_D2, the 18th ternary symbol of pair BI\_D4, and the 14th ternary symbol of pair BI\_D3.

#### **23.4.2 PMA interface messages**

The messages between the PMA and PCS are defined above in 23.3, PMA Service Interface. Communication between a repeater unit and PMA also uses the PMA Service Interface. Communication through the MDI is summarized in Tables 23-2 and 23-3.

TP\_IDLE\_100 is defined in 23.4.1.2. The waveforms used to convey CS1, CS0, and CS-1 are defined in 23.5.1.2.

**Table 23–2—MDI signals transmitted by the PHY**

Signal	Allowed pair	Meaning
CS1	TX_D1, BI_D3 BI_D4	A waveform that conveys the ternary symbol 1. Nominal voltage level +3.5 V.
CS0	TX_D1, BI_D3 BI_D4	A waveform that conveys the ternary symbol 0. Nominal voltage level 0 V.
CS-1	TX_D1, BI_D3 BI_D4	A waveform that conveys the ternary symbol –1. Nominal voltage level –3.5 V.
TP_IDL_100	TX_D1	Idle signal. Indicates transmitter is currently operating at 100 Mb/s.

**Table 23–3—Signals received at the MDI**

Signal	Allowed pair	Meaning
CS1	RX_D2, BI_D3 BI_D4	A waveform that conveys the ternary symbol 1. Nominal transmitted voltage level +3.5 V.
CS0	RX_D2, BI_D3 BI_D4	A waveform that conveys the ternary symbol 0. Nominal transmitted voltage level 0 V.
CS-1	RX_D2, BI_D3 BI_D4	A waveform that conveys the ternary symbol –1. Nominal transmitted voltage level –3.5 V.
TP_IDL_100	RX_D2	Idle signal. Indicates transmitter is currently operating at 100 Mb/s.

TP\_IDL\_100 is defined in 23.4.1.2. The encodings for CS1, CS0, and CS-1 are defined in 23.5.1.2.

Re-timing of CS1, CS0, and CS-1 signals within the PMA is required.

### 23.4.3 PMA state diagrams

The notation used in the state diagrams follows the conventions of 21.5. Transitions shown without source states are evaluated continuously and take immediate precedence over all other conditions.

#### 23.4.3.1 PMA constants

##### CS0

A waveform that conveys the ternary symbol 0.

Value: CS0 has a nominal voltage of 0 V. See 23.5.1.2.

##### CS1

A waveform that conveys the ternary symbol 1.

Value: CS1 has a nominal peak voltage of +3.5 V. See 23.5.1.2.

##### CS-1

A waveform that conveys the ternary symbol –1.

Value: CS-1 has a nominal peak voltage of –3.5 V. See 23.5.1.2.

**link\_100\_max**

A constant.

Value: Greater than 5.0 ms and less than 7.0 ms.

Used by link\_max\_timer to detect the absence of 100BASE-T4 link test pulses on pair RX\_D2.

**link\_100\_min**

A constant.

Value: Greater than 0.15 ms and less than 0.45 ms.

Used by cnt\_link to detect link test pulses on pair RX\_D2 that are too close together to be valid 100BASE-T4 link test pulses.

**23.4.3.2 State diagram variables****pma\_reset**

Causes reset of all PCS functions.

Values: ON and OFF

Set by: PMA Reset

**pma\_carrier**

A version of carrier\_status used internally by the PMA sublayer. The variable pma\_carrier always functions regardless of the link status. The value of pma\_carrier is passed on through the PMA service interface as carrier\_status when rcv=ENABLE. At other times, the passage of pma\_carrier information to the PMA service interface is blocked.

Values: ON, OFF

Set by: PMA CARRIER

**rcv**

Controls the flow of data from the PMA to PCS through the PMA\_UNITDATA.indicate message.

Values: ENABLE (receive is enabled)  
DISABLE (the PMA always sends PMA\_UNITDATA.indicate (IDLE), and carrier\_status is set to OFF)

**xmit**

Controls the flow of data from PCS to PMA through the PMA\_UNITDATA.request message.

Values: ENABLE (transmit is enabled)  
DISABLE (the PMA interprets all PMA\_UNITDATA.request messages as PMA\_UNITDATA.request (IDLE). The PMA transmits no data, but continues sending TP\_IDL\_100).

**23.4.3.3 State diagram timers****link\_max\_timer**

A re-triggerable timer.

Values: The condition link\_max\_timer\_done goes true when the timer expires.

Restart when: Timer is restarted for its full duration by every occurrence of either a link test pulse on pair RX\_D2 or the assertion of pma\_carrier=ON (restarting the timer resets the condition link\_max\_timer\_done).

Duration: link\_100\_max

Used by Link Integrity to detect the absence of 100BASE-T4 link test pulses on pair RX\_D2.

**23.4.3.4 State diagram counters**

cnt\_link

Counts number of 100BASE-T4 link test pulses (see 23.5.1.3.1) received on pair RX\_D2.

Values: nonnegative integers

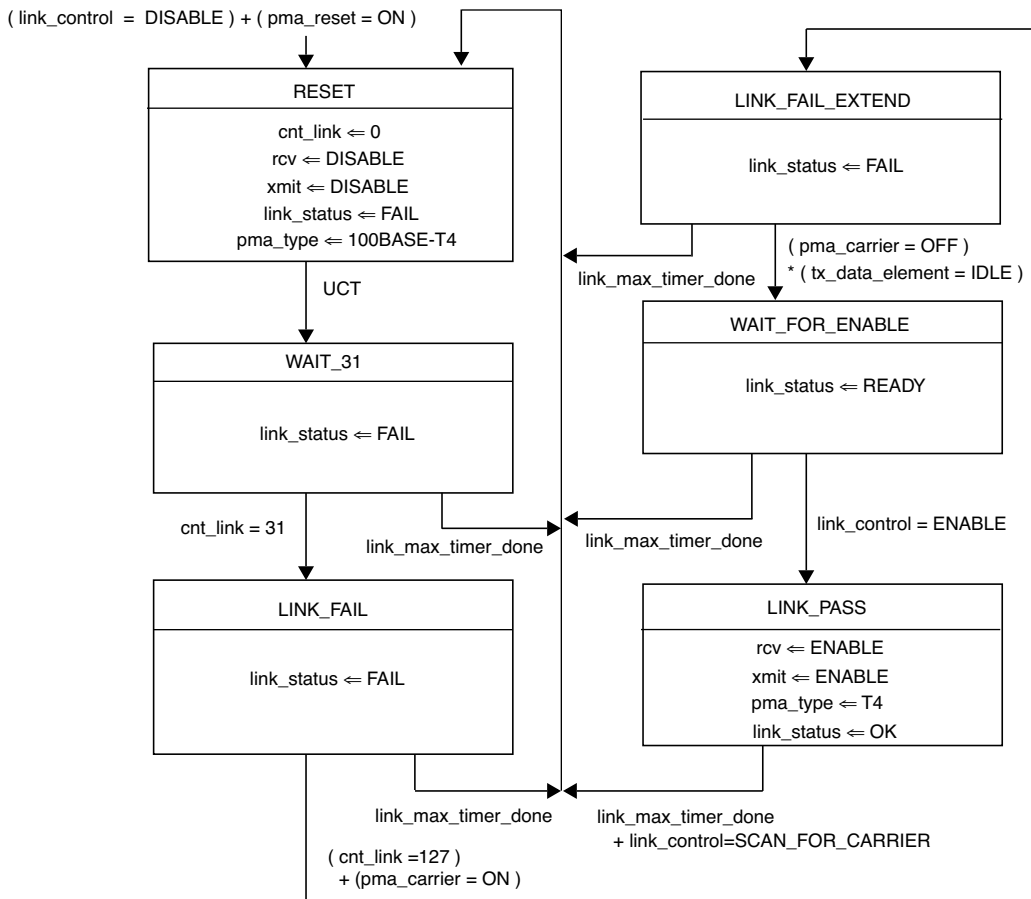
Reset to zero: On either of two conditions:

- a) While in any state other than LINK\_PASS, reset counter to zero if successive link test pulses are received within link\_100\_min.
- b) While in any state, reset to zero if link\_max\_timer expires.

While in the LINK\_PASS state, ignore pulses received within link\_100\_min (i.e., do not count them).

**23.4.3.5 Link Integrity state diagram**

The Link Integrity state diagram is shown in Figure 23–12.



NOTE—The variables link\_control and link\_status are designated as link\_control\_[T4] and link\_status\_[T4], respectively, by the Auto-Negotiation Arbitration state diagram (Figure 28–16).

**Figure 23–12—Link Integrity state diagram**

## 23.5 PMA electrical specifications

This clause defines the electrical characteristics of the PHY at the MDI.

The ground reference point for all common-mode tests is the MII ground circuit. Implementations without an MII use the chassis ground. The values of all components in test circuits shall be accurate to within  $\pm 1\%$  unless otherwise stated.

### 23.5.1 PMA-to-MDI interface characteristics

#### 23.5.1.1 Isolation requirement

The PHY shall provide electrical isolation between the DTE, or repeater circuits including frame ground, and all MDI leads. This electrical separation shall withstand at least one of the following electrical strength tests:

- a) 1500 V rms at 50 Hz to 60 Hz for 60 s, applied as specified in subclause 5.3.2 of IEC 60950: 1991.
- b) 2250 Vdc for 60 s, applied as specified in subclause 5.3.2 of IEC 60950: 1991.
- c) A sequence of ten 2400 V impulses of alternating polarity, applied at intervals of not less than 1 s. The shape of the impulses shall be 1.2/50  $\mu$ s (1.2  $\mu$ s virtual front time, 50  $\mu$ s virtual time or half value), as defined in IEC 60060.

There shall be no insulation breakdown, as defined in subclause 5.3.2 of IEC 60950: 1991, during the test. The resistance after the test shall be at least 2 M $\Omega$ , measured at 500 Vdc.

#### 23.5.1.2 Transmitter specifications

The PMA shall provide the Transmit function specified in 23.4.1.2 in accordance with the electrical specifications of this clause.

Where a load is not specified, the transmitter shall meet requirements of this clause when each transmit output is connected to a differentially connected 100  $\Omega$  resistive load.

##### 23.5.1.2.1 Peak differential output voltage

While repetitively transmitting the ternary sequence [0 0 1 0 0 0 0 0 -1 0 0 0] (leftmost ternary symbol first), and while observing the differential transmitted output at the MDI, for any pair, with no intervening cable, the absolute value of both positive and negative peaks shall fall within the range of 3.15 V to 3.85 V (3.5 V  $\pm$  10%).

##### 23.5.1.2.2 Differential output templates

While repetitively transmitting the ternary sequence [0 0 1 0 0 0 0 0 -1 0 0 0], and while observing the transmitted output at the MDI, the observed waveform shall fall within the normalized transmit template listed in Table 23-4. Portions of this table are represented graphically in Figure 23-13. The entire normalized transmit template shall be scaled by a single factor between 3.15 and 3.85. It is a functional requirement that linear interpolation be used between points. The template time axis may be shifted horizontally to attain the most favorable match. In addition to this simple test pattern, all other pulses, including link integrity pulses and also including the first pulse of each packet preamble, should meet this same normalized transmit template, with appropriate shifting and linear superposition of the CS1 and CS-1 template limits. Transmitters are allowed to insert additional delay in the transmit path in order to meet the first pulse requirement, subject to the overall timing limitations listed in 23.11, Timing summary.

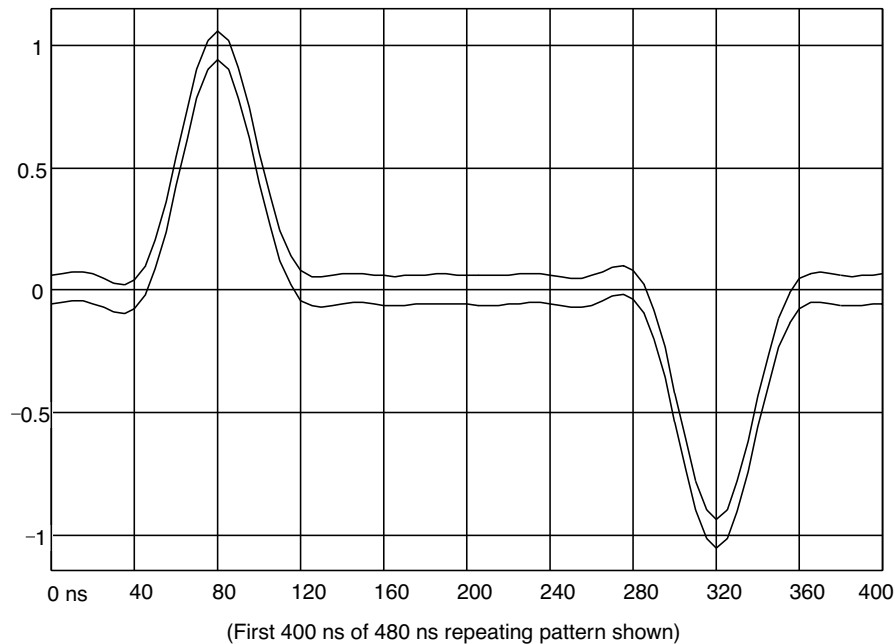
While transmitting the TP\_IDL\_100 signal, and while observing the transmitted output at the MDI, the observed waveform shall fall within the normalized link pulse template listed in Table 23–4. Portions of this table are represented graphically in Figure 23–14. The entire template shall be scaled by the same factor used for the normalized transmit template test. It is a functional requirement that linear interpolation be used between template points. The template time axis may be shifted horizontally to attain the most favorable match.

After transmitting seven or more consecutive CS0 waveforms during the TP\_IDL\_100 signal, each pair, as observed using the 100BASE-T4 Transmit Test Filter (23.5.1.2.3) connected to the MDI, shall attain a state within 50 mV of zero.

When the TX\_D1, BI\_D3, or BI\_D4 pair is driven with a repeating pattern (1 -1 1 -1 ...) any harmonic measured at the MDI output shall be at least 27 dB below the fundamental at 12.5 MHz.

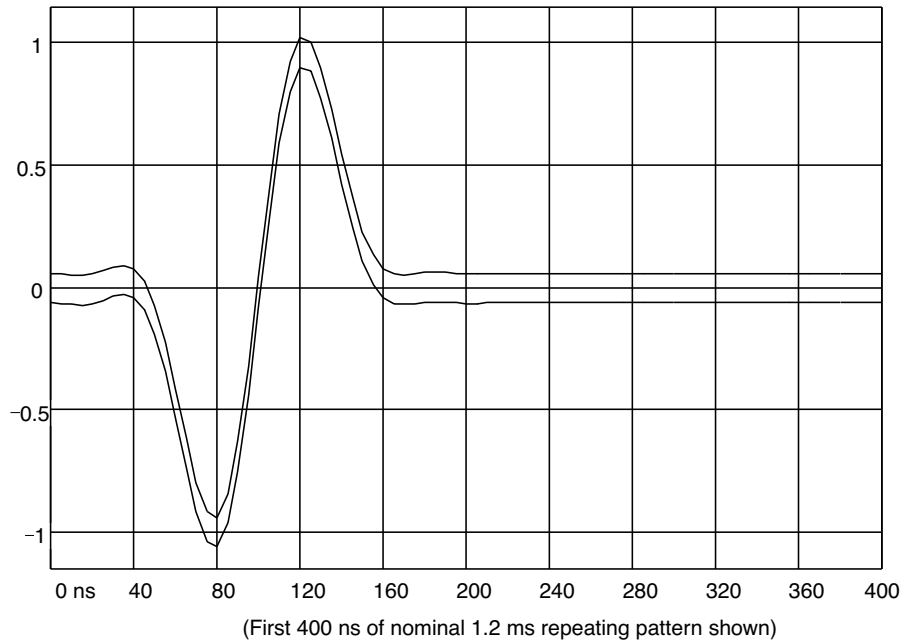
NOTE 1—The specification on maximum spectral components is not intended to ensure compliance with regulations concerning RF emissions. The implementor should consider any applicable local, national, or international regulations. Additional filtering of spectral components may therefore be necessary.

NOTE 2—The repetitive pattern [0 0 1 0 0 0 0 -1 0 0 0] (leftmost ternary symbol first) may be synthesized using the 8B6T coding rules from a string of repeating data octets with value 73 hex. The repetitive pattern [ 1 -1 1 -1 1 -1] (leftmost ternary symbol first) may be synthesized using the 8B6T coding rules from a string of repeating data octets with value 92 hex.



**Figure 23–13—Normalized transmit template as measured at MD**





**Figure 23–14—Normalized link pulse template as measured at MDI**

The ideal template values may be automatically generated from the following equations:

$$\begin{array}{l} \text{Laplace transform of} \\ \text{Ideal transmit response} \end{array} \quad \text{IdealResponse}(s) = \frac{\text{Ideal}(s)}{\text{LPF}(s)}$$

Where  $\text{Ideal}(s)$  is a 100% raised cosine system response

Where  $\text{LPF}(s)$  is a 3-pole Butterworth low pass filter response with  $-3$  dB point at 25 MHz

Convert  $\text{IdealResponse}(s)$  from frequency domain to time domain

Use at least 8 samples per ternary symbol for the conversion

Superimpose alternating positive and negative copies of the ideal time response, separated by 6 ternary symbol times, to form the ideal transmit voltage waveform.

The template limits are formed by offsetting the ideal transmit voltage waveform by plus and minus 6% of its peak.

### 23.5.1.2.3 Differential output ISI (intersymbol interference)

While observing a pseudo-random 8B6T coded data sequence (with every 6T code group represented at least once) preceded by at least 128 octets and followed by at least 128 octets of data, and while observing the transmitted output through a 100BASE-T4 Transmit Test Filter (one implementation of which is depicted in Figure 23–16), the ISI shall be less than 9%. The ISI for this test is defined by first finding the largest of the three peak-to-peak ISI error voltages marked in Figure 23–15 as TOP ISI, MIDDLE ISI, and BOTTOM ISI.

**Table 23–4—Normalized voltage templates as measured at the MDI**

Time, ns	Normalized transmit template, pos. limit	Normalized transmit template, neg. limit	Normalized link template, pos. limit	Normalized link template, neg. limit
0	0.060	-0.061	0.061	-0.060
5	0.067	-0.054	0.056	-0.065
10	0.072	-0.049	0.052	-0.069
15	0.072	-0.049	0.052	-0.069
20	0.063	-0.058	0.058	-0.063
25	0.047	-0.074	0.071	-0.050
30	0.030	-0.091	0.086	-0.035
35	0.023	-0.098	0.094	-0.027
40	0.041	-0.080	0.080	-0.041
45	0.099	-0.022	0.027	-0.094
50	0.206	0.085	-0.076	-0.197
55	0.358	0.237	-0.231	-0.352
60	0.544	0.423	-0.428	-0.549
65	0.736	0.615	-0.640	-0.761
70	0.905	0.784	-0.829	-0.950
75	1.020	0.899	-0.954	-1.075
80	1.060	0.940	-0.977	-1.098
85	1.020	0.899	-0.876	-0.997
90	0.907	0.786	-0.653	-0.774
95	0.744	0.623	-0.332	-0.453
100	0.560	0.439	0.044	-0.077
105	0.384	0.263	0.419	0.298
110	0.239	0.118	0.738	0.617
115	0.137	0.016	0.959	0.838
120	0.077	-0.044	1.060	0.940
125	0.053	-0.068	1.044	0.923
130	0.050	-0.071	0.932	0.811
135	0.057	-0.064	0.759	0.638
140	0.064	-0.057	0.565	0.444
145	0.067	-0.054	0.383	0.262
150	0.065	-0.056	0.238	0.117
155	0.061	-0.060	0.138	0.017
160	0.057	-0.064	0.081	-0.040
165	0.055	-0.066	0.057	-0.064

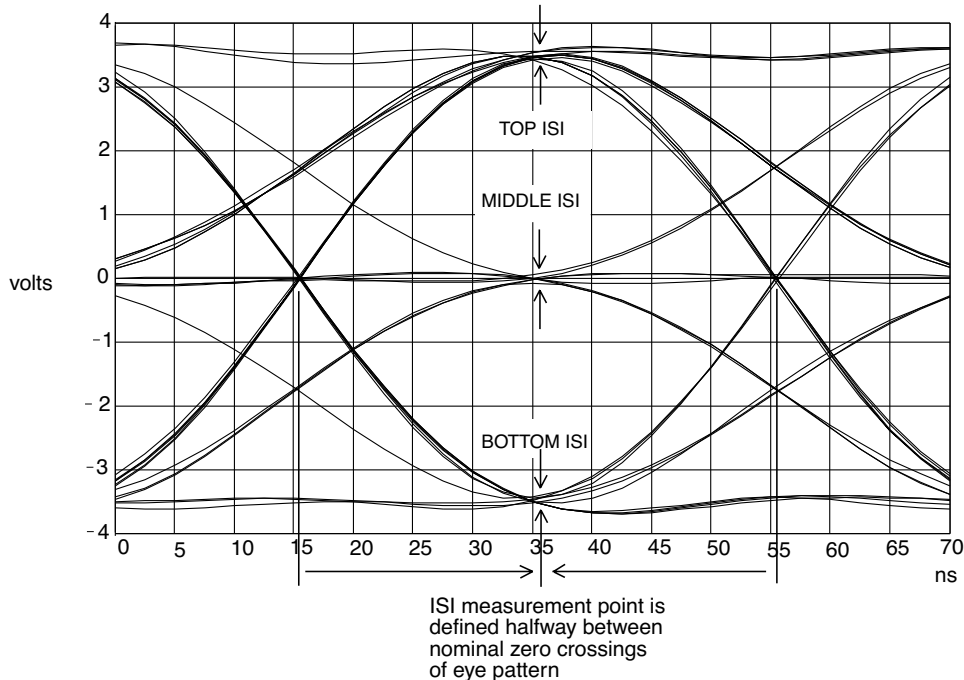
**Table 23–4—Normalized voltage templates as measured at the MDI (continued)**

Time, ns	Normalized transmit template, pos. limit	Normalized transmit template, neg. limit	Normalized link template, pos. limit	Normalized link template, neg. limit
170	0.056	-0.065	0.054	-0.067
175	0.059	-0.062	0.058	-0.063
180	0.062	-0.059	0.063	-0.058
185	0.064	-0.057	0.064	-0.057
190	0.064	-0.057	0.063	-0.058
195	0.062	-0.059	0.060	-0.061
200	0.060	-0.061	0.058	-0.063
205	0.057	-0.064	0.058	-0.063
210	0.056	-0.065	0.059	-0.062
215	0.058	-0.063	0.060	-0.061
220	0.061	-0.060	0.062	-0.059
225	0.064	-0.057	0.062	-0.059
230	0.066	-0.055	0.062	-0.059
235	0.065	-0.056	0.061	-0.060
240	0.061	-0.060	0.060	-0.061
245	0.054	-0.067	0.060	-0.061
250	0.049	-0.072	0.060	-0.061
255	0.049	-0.072	0.060	-0.061
260	0.058	-0.063	0.061	-0.060
265	0.074	-0.047	0.061	-0.060
270	0.091	-0.030	0.061	-0.060
275	0.099	-0.022	0.061	-0.060
280	0.080	-0.041	0.060	-0.061
285	0.022	-0.099	0.060	-0.061
290	-0.085	-0.206	0.060	-0.061
295	-0.238	-0.359	0.060	-0.061
300	-0.423	-0.544	0.061	-0.060
305	-0.615	-0.736	0.061	-0.060
310	-0.783	-0.904	0.061	-0.060
315	-0.899	-1.020	0.061	-0.060
320	-0.940	-1.061	0.060	-0.061
325	-0.899	-1.020	0.060	-0.061
330	-0.786	-0.907	0.060	-0.061
335	-0.623	-0.744	0.060	-0.061

**Table 23–4—Normalized voltage templates as measured at the MDI (continued)**

Time, ns	Normalized transmit template, pos. limit	Normalized transmit template, neg. limit	Normalized link template, pos. limit	Normalized link template, neg. limit
340	−0.439	−0.560	0.061	−0.060
345	−0.263	−0.384	0.061	−0.060
350	−0.118	−0.239	0.061	−0.060
355	−0.016	−0.137	0.061	−0.060
360	0.044	−0.077	0.060	−0.061
365	0.068	−0.053	0.060	−0.061
370	0.070	−0.051	0.060	−0.061
375	0.064	−0.057	0.060	−0.061
380	0.057	−0.064	0.061	−0.060
385	0.054	−0.067	0.061	−0.060
390	0.056	−0.065	0.061	−0.060
395	0.060	−0.061	0.061	−0.060
400	0.064	−0.057	0.060	−0.061
405	0.065	−0.056	0.060	−0.061
410	0.064	−0.057	0.060	−0.061
415	0.061	−0.060	0.060	−0.061
420	0.059	−0.062	0.061	−0.060
425	0.058	−0.063	0.061	−0.060
430	0.059	−0.062	0.061	−0.060
435	0.060	−0.061	0.061	−0.060
440	0.061	−0.060	0.060	−0.061
445	0.062	−0.059	0.060	−0.061
450	0.062	−0.059	0.060	−0.061
455	0.061	−0.060	0.060	−0.061
460	0.060	−0.061	0.061	−0.060
465	0.059	−0.062	0.061	−0.060
470	0.060	−0.061	0.061	−0.060
475	0.060	−0.061	0.061	−0.060
480	0.061	−0.060	0.060	−0.061

The largest of these peak-to-peak ISI error voltages is then divided by the overall peak-to-peak signal voltage. (The technique of limiting the ratio of worst ISI to overall peak-to-peak voltage at 9% accomplishes the same end as limiting the ratio of worst ISI to nominal peak-to-peak at 10%.)



**Figure 23-15—Definition of sampling points for ISI measurement**

It is a mandatory requirement that the peak-to-peak ISI, and the overall peak-to-peak signal voltage, be measured at a point in time halfway between the nominal zero crossings of the observed eye pattern.

It is a mandatory requirement that the 100BASE-T4 Transmit Test Filter perform the function of a third-order Butterworth filter with its  $-3$  dB point at 25.0 MHz.

One acceptable implementation of a 100BASE-T4 Transmit Test Filter appears in Figure 23-16. That implementation uses the 100BASE-T4 Transmit Test Filter as a line termination. The output of the filter is terminated in  $100\ \Omega$ . It is a mandatory requirement that such implementations of the 100BASE-T4 Transmit Test Filter be designed such that the reflection loss of the filter, when driven by a  $100\ \Omega$  source, exceeds 17 dB across the frequency range 2 to 12.5 MHz.

Equivalent circuits that implement the same overall transfer function are also acceptable. For example, the 100BASE-T4 Transmit Test Filter may be tapped onto a line in parallel with an existing termination. It is a mandatory requirement that such implementations of the 100BASE-T4 Transmit Test Filter be designed with an input impedance sufficiently high that the reflection loss of the parallel combination of filter and  $100\ \Omega$  termination, when driven by  $100\ \Omega$ , exceeds 17 dB across the frequency range 2 to 12.5 MHz.

#### 23.5.1.2.4 Transmitter differential output impedance

The differential output impedance as measured at the MDI for each transmit pair shall be such that any reflection due to differential signals incident upon the MDI from a balanced cable having an impedance of  $100\ \Omega$  is at least 17 dB below the incident signal, over the frequency range of 2.0 MHz to 12.5 MHz. This return loss shall be maintained at all times when the PHY is fully powered.

With every transmitter connected as in Figure 23-17, and while transmitting a repeating sequence of packets as specified in Table 23-5, the amount of droop on any transmit pair as defined in Figure 23-18 during the transmission of eop1 and eop4 shall not exceed 6.0%.

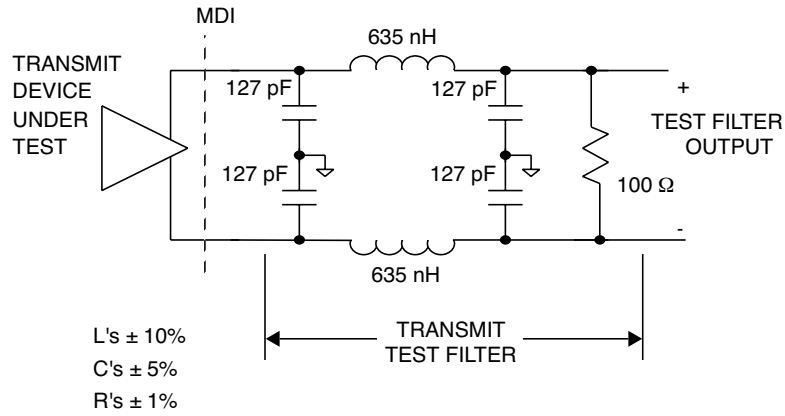


Figure 23–16—Acceptable implementation of transmit test filter

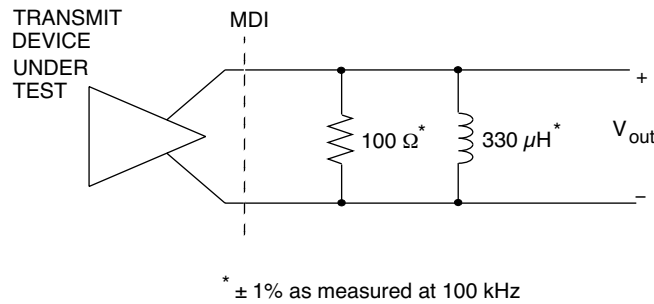


Figure 23–17—Output impedance test setup

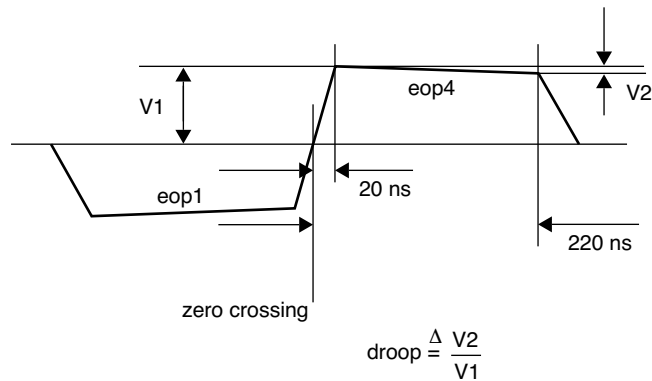
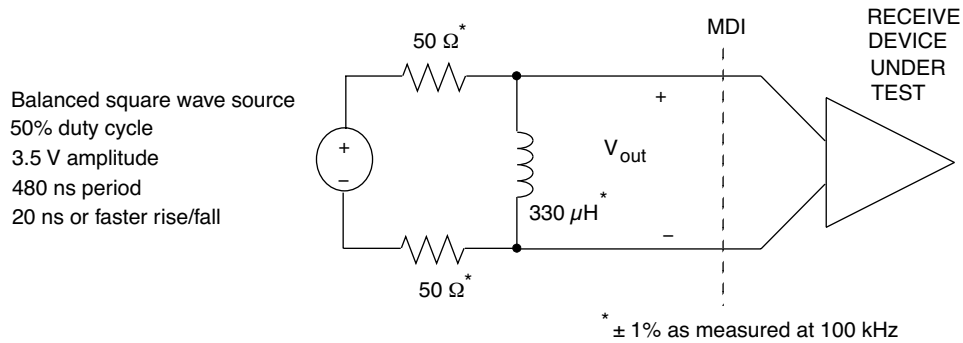


Figure 23–18—Measurement of output droop

Table 23–5—Sequence of packets for droop test

Packet sequence (Transmit this sequence of packets in a repetitive loop)	Packet length (Number of data octets)	Data, hex (All octets in each packet are the same)
first packet	64	AA
second packet	65	AA
third packet	66	AA



**Figure 23-19—Input impedance test setup**

### 23.5.1.2.5 Output timing jitter

While repetitively transmitting a random sequence of valid 8B6T code words, and while observing the output of a 100BASE-T4 Transmit Test Filter connected at the MDI to any of the transmit pairs as specified in 23.5.1.2.3, the measured jitter shall be no more than 4 ns p-p. For the duration of the test, each of the other transmit pairs shall be connected to either a 100BASE-T4 Transmit Test Filter or a 100 Ω resistive load.

NOTE 1—Jitter is the difference between the actual zero crossing point in time and the ideal time. For various ternary transitions, the zero crossing time is defined differently. For transitions between +1 and –1 or vice versa, the zero crossing point is defined as that point in time when the voltage waveform crosses zero. For transitions between zero and the other values, or from some other value to zero, the zero crossing time is defined as that point in time when the voltage waveform crosses the boundary between logical voltage levels, halfway between zero volts and the logical +1 or logical –1 ideal level.

NOTE 2—The ideal zero crossing times are contained in a set of points  $\{t_n\}$  where  $t_n = t_0 + n/f$ , where  $n$  is an integer, and  $f$  is in the range 25.000 MHz  $\pm$  0.01%. A collection of zero crossing times satisfies the jitter requirement if there exists a pair  $(t_0, f)$  such that each zero crossing time is separated from some member of  $\{t_n\}$  by no more than 4 ns.

### 23.5.1.2.6 Transmitter impedance balance

The common-mode to differential-mode impedance balance of each transmit output shall exceed

$$29 - 17 \log \left( \frac{f}{10} \right) \text{ dB}$$

where  $f$  is the frequency (in MHz) over the frequency range 2.0 MHz to 12.5 MHz. The balance is defined as

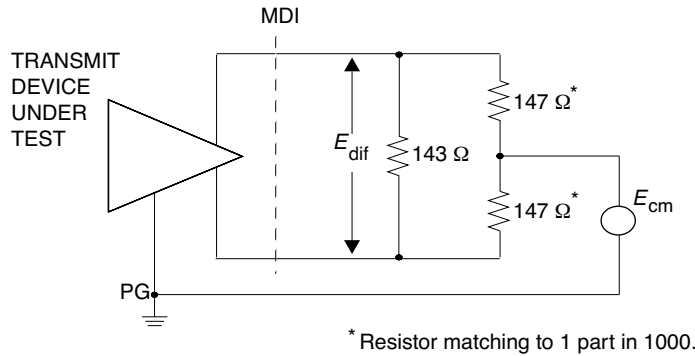
$$20 \log \left( \frac{E_{\text{cm}}}{E_{\text{dif}}} \right)$$

where  $E_{\text{cm}}$  is an externally applied sine-wave voltage as shown in Figure 23-20.

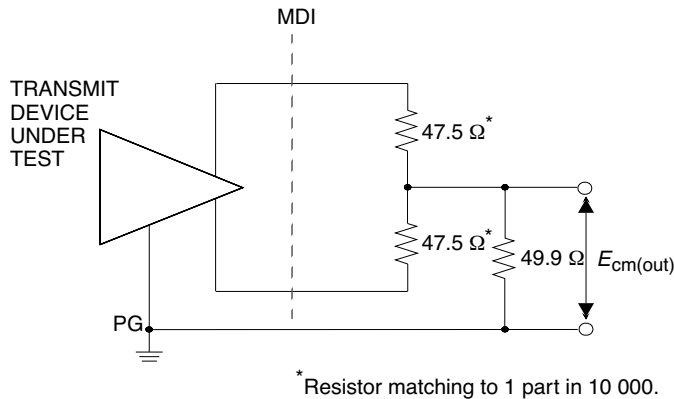
NOTE—The balance of the test equipment (such as the matching of the test resistors) must be insignificant relative to the balance requirements.

### 23.5.1.2.7 Common-mode output voltage

The implementor should consider any applicable local, national, or international regulations. Driving unshielded twisted pairs with high-frequency, common-mode voltages may result in interference to other equipment. FCC conducted and radiated emissions tests may require that, while transmitting data, the magnitude of the total common-mode output voltage,  $E_{\text{cm(out)}}$ , on any transmit circuit, be less than a few millivolts when measured as shown in Figure 23-21.



**Figure 23-20—Transmitter impedance balance and common-mode rejection test circuit**



**Figure 23-21—Common-mode output voltage test circuit**

**23.5.1.2.8 Transmitter common-mode rejection**

The application of  $E_{cm}$  as shown in Figure 23-20 shall not change the differential voltage at any transmit output,  $E_{dif}$ , by more than 100 mV for all data sequences while the transmitter is sending data. Additionally, the edge jitter added by the application of  $E_{cm}$  shall be no more than 1.0 ns.  $E_{cm}$  shall be a 15 V peak 10.1 MHz sine wave.

**23.5.1.2.9 Transmitter fault tolerance**

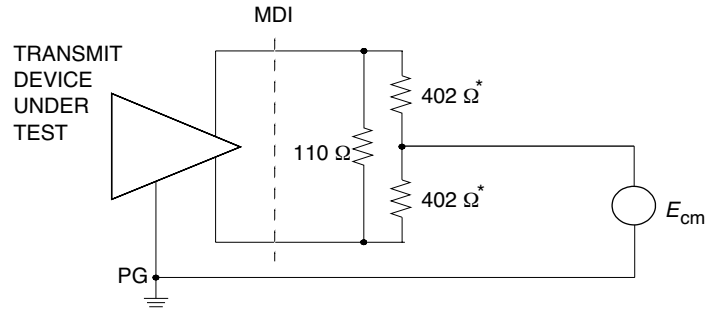
Transmitters, when either idle or nonidle, shall withstand without damage the application of short circuits across any transmit output for an indefinite period of time and shall resume normal operation after such faults are removed. The magnitude of the current through such a short circuit shall not exceed 420 mA.

Transmitters, when either idle or nonidle, shall withstand without damage a 1000 V common-mode impulse applied at  $E_{cm}$  of either polarity (as indicated in Figure 23-22). The shape of the impulse shall be 0.3/50  $\mu$ s (300 ns virtual front time, 50  $\mu$ s virtual time of half value), as defined in IEC 60060.

**23.5.1.2.10 Transmit clock frequency**

The ternary symbol transmission rate on each pair shall be 25.000 MHz  $\pm$  0.01%.





\* Resistor matching to 1 part in 100.

**Figure 23-22—Transmitter fault tolerance test circuit**

### 23.5.1.3 Receiver specifications

The PMA shall provide the Receive function specified in 23.4.1.3 in accordance with the electrical specifications of this clause. The patch cables and interconnecting hardware used in test configurations shall meet Category 5 specifications as in ISO/IEC 11801: 1995.

The term *worst-case UTP model*, as used in this clause, refers to lumped-element cable model shown in Figure 23-23 that has been developed to simulate the attenuation and group delay characteristics of 100 m of worst-case Category 3 PVC UTP cable.

This constant resistance filter structure has been optimized to best match the following amplitude and group delay characteristics, where the argument  $f$  is in hertz, and the argument  $x$  is the cable length in meters. For the worst-case UTP model, argument  $x$  was set to 100 m, and the component values determined for a best least mean squared fit of both real and imaginary parts of  $H(f, x)$  over the frequency range 2 to 15 MHz.

NOTE—This group delay model is relative and does not include the fixed delay associated with 100 m of Category 3 cable. An additional 570 ns of fixed delay should be added in order to obtain the absolute group delay.

$$PropagationImag(f, x) = j(-10) \sqrt{\frac{f}{10^7}} \left( \frac{x}{100} \right)$$

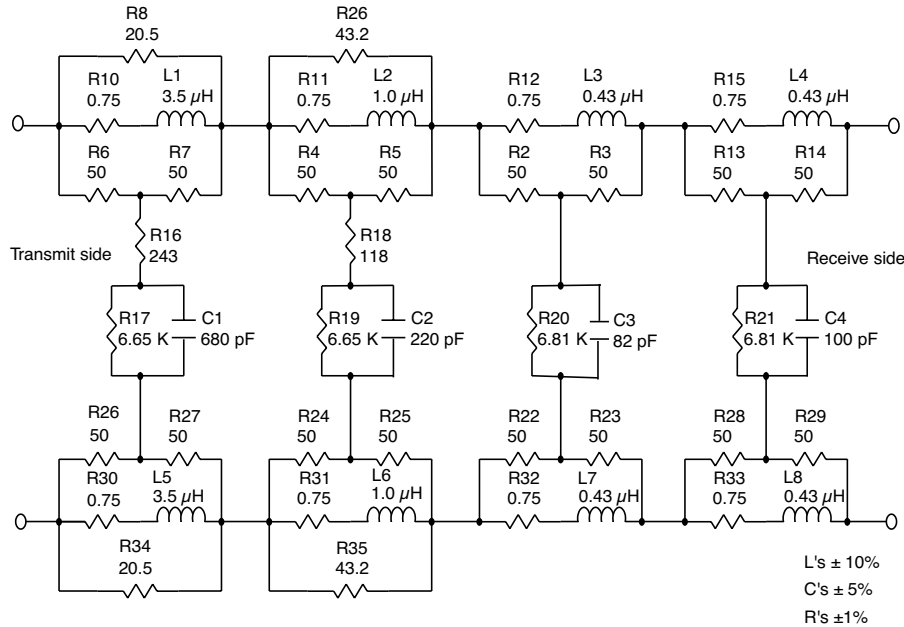
$$PropagationReal(f, x) = - \left( 7.1 \sqrt{\frac{f}{10^6}} + 0.70 \frac{f}{10^6} \right) \left( \frac{x}{305} \right)$$

$$\frac{PropagationImag(f, x) + PropagationReal(f, x)}{20}$$

$$H(f, x) = 10$$

#### 23.5.1.3.1 Receiver differential input signals

Differential signals received on the receive inputs that were transmitted within the constraints of 23.5.1.2, and have then passed through a worst-case UTP model, shall be correctly translated into one of the PMA\_UNITDATA.indicate messages and sent to the PCS. In addition, the receiver, when presented with a link test pulse generated according to the requirements of 23.4.1.2 and followed by at least 3T of silence on pair RX\_D2, shall accept it as a link test pulse.



**Figure 23–23—Worst-case UTP model**

Both data and link test pulse receive features shall be tested in at least two configurations: using the worst-case UTP model, and with a connection less than one meter in length between transmitter and receiver.

A receiver is allowed to discard the first received packet after the transition into state LINK\_PASS, using that packet for the purpose of fine-tuning its receiver equalization and clock recovery circuits.

NOTE—Implementors may find it practically impossible to meet the requirements of this subclause without using some form of adaptive equalization.

### 23.5.1.3.2 Receiver differential noise immunity

The PMA, when presented with 8B6T encoded data meeting the requirements of 23.5.1.3.1, shall translate this data into PMA\_UNITDATA.indicate (DATA) messages with a bit loss of no more than that specified in 23.4.1.3.

The PMA Carrier Sense function shall *not* set pma\_carrier=ON upon receiving any of the following signals on pair RX\_D2 at the receiving MDI, as measured using a 100BASE-T4 transmit test filter (23.5.1.2.3):

- All signals having a peak magnitude less than 325 mV.
- All continuous sinusoidal signals of amplitude less than 8.7 V peak-to-peak and frequency less than 1.7 MHz.
- All sine waves of single cycle or less duration, starting with phase 0° or 180°, and of amplitude less than 8.7 V peak-to-peak, where the frequency is between 1.7 MHz and 15 MHz. For a period of 7 BT before and after this single cycle, the signal shall be less than 325 mV.
- Fast link pulse burst (FLP burst), as defined in Clause 28.
- The link integrity test pulse signal TP\_IDL\_100.

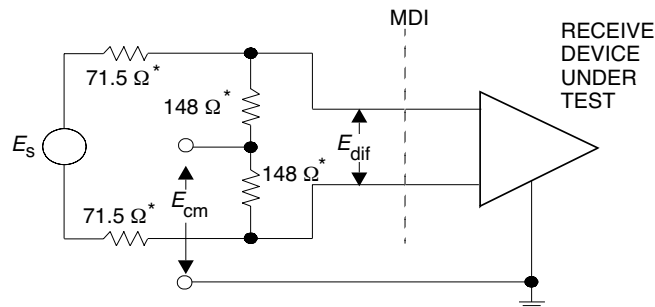
### 23.5.1.3.3 Receiver differential input impedance

The differential input impedance as measured at the MDI for each receive input shall be such that any reflection due to differential signals incident upon each receive input from a balanced cable having an impedance of  $100\ \Omega$  is at least 17 dB below the incident signal, over the frequency range of 2.0 MHz to 12.5 MHz. This return loss shall be maintained at all times when the PHY is fully powered.

With each receiver connected as in Figure 23–19, and with the source adjusted to simulate eop1 and eop4 (50% duty cycle square wave with 3.5 V amplitude, period of 480 ns, and risetime of 20 ns or faster), the amount of droop on each receive pair as defined in Figure 23–18 shall not exceed 6.0%.

### 23.5.1.3.4 Common-mode rejection

While receiving packets from a compliant 100BASE-T4 transmitter connected to all MDI pins, a receiver shall send the proper PMA\_UNITDATA.indicate messages to the PCS for any differential input signal  $E_s$  that results in a signal  $E_{dif}$  that meets 23.5.1.3.1 even in the presence of common-mode voltages  $E_{cm}$  (applied as shown in Figure 23–24).  $E_{cm}$  shall be a 25 V peak-to-peak square wave, 500 kHz or lower in frequency, with edges no slower than 4 ns (20%–80%), connected to each of the receive pairs RX\_D2, BI\_D3, and BI\_D4.



\* Resistor matching to 1 part in 1000.

Figure 23–24—Receiver common-mode rejection test circuit

### 23.5.1.3.5 Receiver fault tolerance

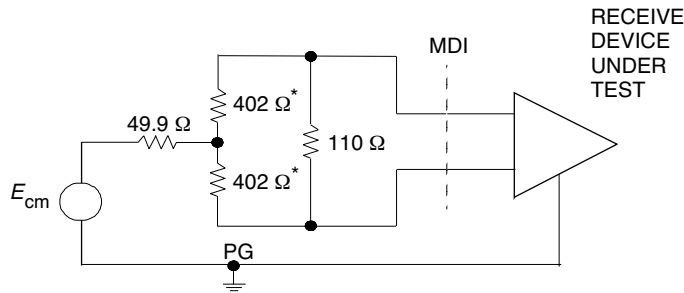
The receiver shall tolerate the application of short circuits between the leads of any receive input for an indefinite period of time without damage and shall resume normal operation after such faults are removed. Receivers shall withstand without damage a 1000 V common-mode impulse of either polarity ( $E_{cm}$  as indicated in Figure 23–25). The shape of the impulse shall be 0.3/50  $\mu s$  (300 ns virtual front time, 50  $\mu s$  virtual time of half value), as defined in IEC 60060.

### 23.5.1.3.6 Receiver frequency tolerance

The receive feature shall properly receive incoming data with a ternary symbol rate within the range  $25.000\ \text{MHz} \pm 0.01\%$ .

## 23.5.2 Power consumption

After 100 ms following PowerOn, the current drawn by the PHY shall not exceed 0.75 A when powered through the MII.



\* Resistor matching to 1 part in 100.

**Figure 23–25— Common-mode impulse test circuit**

The PHY shall be capable of operating from all voltage sources allowed by Clause 22, including those current limited to 0.75 A, as supplied by the DTE or repeater through the resistance of all permissible MII cables.

The PHY shall not introduce extraneous signals on the MII control circuits during normal power-up and power-down.

While in power-down mode the PHY is not required to meet any of the 100BASE-T4 performance requirements.

## 23.6 Link segment characteristics

### 23.6.1 Cabling

Cabling and installation practices generally suitable for use with this standard appear in ISO/IEC 11801: 1995. Exceptions, notes, and additional requirements are as listed below.

- a) 100BASE-T4 uses a star topology. Horizontal cabling is used to connect PHY entities.
- b) 100BASE-T4 is an ISO/IEC 11801: 1995 class C application, with additional installation requirements and transmission parameters specified in 23.6.2 through 23.6.4. The highest fundamental frequency transmitted by 8B6T coding is 12.5 MHz. The aggregate data rate for three pairs using 8B6T coding is 100 Mb/s.
- c) 100BASE-T4 shall use four pairs of balanced cabling, Category 3 or better, with a nominal characteristic impedance of 100 Ω.
- d) When using Category 3 cable for the link segment, Clause 23 recommends, but does not require, the use of Category 4 or better connecting hardware, patch cords and jumpers. The use of Category 4 or better connecting hardware increases the link segment composite NEXT loss, composite ELFEXT loss and reduces the link segment insertion loss. This lowers the link segment crosstalk noise, which in turn decreases the probability of errors.
- e) The use of shielded cable is outside the scope of this standard.

### 23.6.2 Link transmission parameters

Unless otherwise specified, link segment testing shall be conducted using source and load impedances of 100 Ω.

### 23.6.2.1 Insertion loss

The insertion loss of a simplex link segment shall be no more than 12 dB at all frequencies between 2 and 12.5 MHz. This consists of the attenuation of the twisted pairs, connector losses, and reflection losses due to impedance mismatches between the various components of the simplex link segment. The insertion loss specification shall be met when the simplex link segment is terminated in source and load impedances that satisfy 23.5.1.2.4 and 23.5.1.3.3.

NOTE—The loss of PVC-insulated cable exhibits significant temperature dependence. At temperatures greater than 40 °C, it may be necessary to use a less temperature-dependent cable, such as many Fluorinated Ethylene Propylene (FEP), Polytetrafluoroethylene (PTFE), or Perfluoroalkoxy (PFA) plenum-rated cables.

### 23.6.2.2 Differential characteristic impedance

The magnitude of the differential characteristic impedance of a 3 m length of twisted pair used in a simplex link shall be between 85 Ω and 115 Ω for all frequencies between 2 MHz and 12.5 MHz.

### 23.6.2.3 Coupling parameters

In order to limit the noise coupled into a simplex link segment from adjacent simplex link segments, Near-End Crosstalk (NEXT) loss and Equal Level Far-End Crosstalk (ELFEXT) loss are specified for each simplex link segment. In addition, since three simplex links (TX\_D1, BI\_D3, and BI\_D4) are used to send data between PHYs and one simplex link (RX\_D2) is used to carry collision information as specified in 23.1.4, Multiple-Disturber NEXT loss and Multiple-Disturber ELFEXT loss are also specified.

#### 23.6.2.3.1 Differential Near-End Crosstalk (NEXT) loss

The differential Near-End Crosstalk (NEXT) loss between two simplex link segments is specified in order to ensure that collision information can be reliably received by the PHY receiver. The NEXT loss between each of the three data carrying simplex link segments and the collision sensing simplex link segment shall be at least  $24.5 - 15 \times \log_{10}(f/12.5)$  (where  $f$  is the frequency in MHz) over the frequency range 2.0 MHz to 12.5 MHz.

#### 23.6.2.3.2 Multiple-disturber NEXT (MDNEXT) loss

Since three simplex links are used to send data between PHYs and one simplex link is used to carry collision information, the NEXT noise that is coupled into the collision, sensing simplex link segment is from multiple (three) signal sources, or disturbers. The MDNEXT loss between the three data carrying simplex link segments and the collision sensing simplex link segment shall be at least  $21.4 - 15 \times \log_{10}(f/12.5)$  dB (where  $f$  is the frequency in MHz) over the frequency range 2.0 to 12.5 MHz. Refer to 12.7.3.2 and Annex B.3 Example Crosstalk Computation for Multiple Disturbers, for a tutorial and method for estimating the MDNEXT loss for an n-pair cable.

#### 23.6.2.3.3 Equal Level Far-End Crosstalk (ELFEXT) loss

Equal Level Far-End Crosstalk (ELFEXT) loss is specified in order to limit the crosstalk noise at the far end of a simplex link segment to meet the BER objective specified in 23.1.2 and the noise specifications of 23.6.3. Far-End Crosstalk (FEXT) noise is the crosstalk noise that appears at the far end of a simplex link segment which is coupled from an adjacent simplex link segment with the noise source (transmitters) at the near end. ELFEXT loss is the ratio of the data signal to FEXT noise at the output of a simplex link segment (receiver input). To limit the FEXT noise from adjacent simplex link segments, the ELFEXT loss between two data carrying simplex link segments shall be greater than  $23.1 - 20 \times \log_{10}(f/12.5)$  dB (where  $f$  is the frequency in MHz) over the frequency range 2.0 MHz to 12.5 MHz. ELFEXT loss at frequency  $f$  and distance  $l$  is defined as

$$\text{ELFEXT\_Loss}(f,l) = 20 \times \log_{10} \left( \frac{V_{\text{pds}}}{V_{\text{pcn}}} \right) - \text{SLS\_Loss} \text{ (dB)}$$

where

$V_{pds}$  is the peak voltage of disturbing signal (near-end transmitter)  
 $V_{pcn}$  is the peak crosstalk noise at the far end of disturbed simplex link segment  
 $SLS\_Loss$  is the insertion loss of the disturbing simplex link segment

#### 23.6.2.3.4 Multiple-disturber ELFEXT (MDELFEEXT) loss

Since three simplex links are used to transfer data between PHYs, the FEXT noise that is coupled into an data carrying simplex link segment is from multiple (two) signal sources, or disturbers. The MDELFEEXT loss between a data carrying simplex link segment and the other two data carrying simplex link segments shall be greater than  $20.9 - 20 \times \log_{10}(f/12.5)$  (where  $f$  is the frequency in MHz) over the frequency range 2.0 MHz to 12.5 MHz. Refer to 12.7.3.2 and Appendix A3, Example Crosstalk Computation for Multiple Disturbers, for a tutorial and method for estimating the MDELFEEXT loss for an n-pair cable.

#### 23.6.2.4 Delay

Since T4 sends information over three simplex link segments in parallel, the absolute delay of each and the differential delay are specified to comply with network round-trip delay limits and ensure the proper decoding by receivers, respectively.

##### 23.6.2.4.1 Maximum link delay

The propagation delay of a simplex link segment shall not exceed 570 ns at all frequencies between 2.0 MHz and 12.5 MHz.

##### 23.6.2.4.2 Maximum link delay per meter

The propagation delay per meter of a simplex link segment shall not exceed 5.7 ns/m at all frequencies between 2.0 MHz and 12.5 MHz.

##### 23.6.2.4.3 Difference in link delays

The difference in propagation delay, or skew, under all conditions, between the fastest and the slowest simplex link segment in a link segment shall not exceed 50 ns at all frequencies between 2.0 MHz and 12.5 MHz. It is a further functional requirement that, once installed, the skew between all pair combinations due to environmental conditions shall not vary more than  $\pm 10$  ns, within the above requirement.

#### 23.6.3 Noise

The noise level on the link segments shall be such that the objective error rate is met. The noise environment consists generally of two primary contributors: self-induced near-end crosstalk, which affects the ability to detect collisions, and far-end crosstalk, which affects the signal-to-noise ratio during packet reception.

##### 23.6.3.1 Near-End Crosstalk

The MDNEXT (Multiple-Disturber Near-End Crosstalk) noise on a link segment depends on the level of the disturbing signals on pairs TX\_D1, BI\_D3, and BI\_D4, and the crosstalk loss between those pairs and the disturbed pair, RX\_D2.

The MDNEXT noise on a link segment shall not exceed 325 mVp.

This standard is compatible with the following assumptions:

- a) Three disturbing pairs with 99th percentile pair-to-pair NEXT loss greater than 24.5 dB at 12.5 MHz (i.e., Category 3 cable).
- b) Six additional disturbers (2 per simplex link) representing connectors at the near end of the link segment with 99th percentile NEXT loss greater than 40 dB at 12.5 MHz (i.e., Category 3 connectors installed in accordance with 23.6.4.1).
- c) All disturbers combined according to the MDNEXT Monte Carlo procedure outlined in Appendix A3, Example Crosstalk Computation for Multiple Disturbers.

The MDNEXT noise is defined using three maximum level 100BASE-T4 transmitters sending uncorrelated continuous data sequences while attached to the simplex link segments TX\_D1, BI\_D3, and BI\_D4 (disturbing links), and the noise measured at the output of a filter connected to the simplex link segment RX\_D2 (disturbed link). Each continuous data sequence is a pseudo-random bit pattern having a length of at least 2047 bits that has been coded according to the 8B6T coding rules in 23.2.1.2. The filter is the 100BASE-T4 Transmit Test Filter specified in 23.5.1.2.3.

### 23.6.3.2 Far-End Crosstalk

The MDFEXT (Multiple-Disturber Far-End Crosstalk) noise on a link segment depends on the level of the disturbing signals on pairs TX\_D1, BI\_D3, and BI\_D4, and the various crosstalk losses between those pairs.

The MDFEXT noise on a link segment shall not exceed 87 mVp.

This standard is compatible with the following assumptions:

- a) Two disturbing pairs with 99th percentile ELFEXT (Equal Level Far-End Crosstalk) loss greater than 23 dB at 12.5 MHz.
- b) Nine additional disturbers (three per simplex link) representing connectors in the link segment with 99th percentile NEXT loss greater than 40 dB at 12.5 MHz.
- c) All disturbers combined according to the MDNEXT Monte Carlo procedure outlined in Appendix A3, Example Crosstalk Computation for Multiple Disturbers.

The MDFEXT noise is defined using two maximum level 100BASE-T4 transmitters sending uncorrelated continuous data sequences while attached to two simplex link segments (disturbing links) and the noise measured at the output of a filter connected to the far end of a third simplex link segment (disturbed link). Each continuous data sequence is a pseudo-random bit pattern having a length of at least 2047 bits that has been coded according to the 8B6T coding rules in 23.2.1.2. The filter is the 100BASE-T4 Transmit Test Filter specified in 23.5.1.2.3.

## 23.6.4 Installation practice

### 23.6.4.1 Connector installation practices

The amount of untwisting in a pair as a result of termination to connecting hardware should be no greater than 25 mm (1.0 in) for Category 3 cables. This is the same value recommended in ISO/IEC 11801: 1995 for Category 4 connectors.

### 23.6.4.2 Disallow use of Category 3 cable with more than four pairs

Jumper cables, or horizontal runs, made from more than four pairs of Category 3 cable are not allowed.

### 23.6.4.3 Allow use of Category 5 jumpers with up to 25 pairs

Jumper cables made from up to 25 pairs of Category 5 cable, for the purpose of mass-terminating port connections at a hub, are allowed. Such jumper cables, if used, shall be limited in length to no more than 10 m total.

## 23.7 MDI specification

This clause defines the MDI. The link topology requires a crossover function between PMAs. Implementation and location of this crossover are also defined in this clause.

### 23.7.1 MDI connectors

Eight-pin connectors meeting the requirements of section 3 and figures 1 through 5 of IEC 60603-7: 1990 shall be used as the mechanical interface to the balanced cabling. The plug connector shall be used on the balanced cabling and the jack on the PHY. These connectors are depicted (for informational use only) in Figures 23–26 and 23–27. Table 23-6 shows the assignment of PMA signals to connector contacts for PHYs with and without an internal crossover.

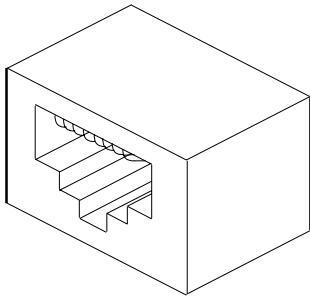


Figure 23–26—MDI connector

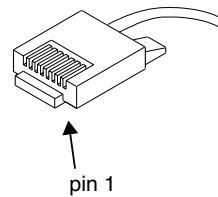


Figure 23–27—Balanced cabling connector

Table 23-6—MDI connection and labeling requirements

Contact	PHY without internal crossover (recommended for DTE) internal PMA signals	PHY with internal crossover (recommended for repeater) internal PMA signals	MDI labeling requirement
1	TX_D1+	RX_D2+	TX_D1+
2	TX_D1–	RX_D2–	TX_D1–
3	RX_D2+	TX_D1+	RX_D2+
4	BI_D3+	BI_D4+	BI_D3+
5	BI_D3–	BI_D4–	BI_D3–
6	RX_D2–	TX_D1–	RX_D2–
7	BI_D4+	BI_D3+	BI_D4+
8	BI_D4–	BI_D3–	BI_D4–



### 23.7.2 Crossover function

It is a functional requirement that a crossover function be implemented in every link segment. The crossover function connects the transmitters of one PHY to the receivers of the PHY at the other end of the link segment. Crossover functions may be implemented internally to a PHY or elsewhere in the link segment. For a PHY that does not implement the crossover function, the MDI labels in the last column of Table 23–4 refer to its own internal circuits (second column). For PHYs that do implement the internal crossover, the MDI labels in the last column of Table 23–4 refer to the internal circuits of the remote PHY of the link segment. Additionally, the MDI connector for a PHY that implements the crossover function shall be marked with the graphical symbol “X”. Internal and external crossover functions are shown in Figure 23–28. The crossover function specified here for pairs TX\_D1 and RX\_D2 is compatible with the crossover function specified in 14.5.2 for pairs TD and RD.

When a link segment connects a DTE to a repeater, it is recommended the crossover be implemented in the PHY local to the repeater. If both PHYs of a link segment contain internal crossover functions, an additional external crossover is necessary. It is recommended that the crossover be visible to an installer from one of the PHYs. When both PHYs contain internal crossovers, it is further recommended in networks in which the topology identifies either a central backbone segment or a central repeater that the PHY furthest from the central element be assigned the external crossover to maintain consistency.

Implicit implementation of the crossover function within a twisted-pair cable, or at a wiring panel, while not expressly forbidden, is beyond the scope of this standard.

### 23.8 System considerations

The repeater unit specified in Clause 27 forms the central unit for interconnecting 100BASE-T4 twisted-pair links in networks of more than two nodes. It also provides the means for connecting 100BASE-T4 twisted-pair links to other 100 Mb/s baseband segments. The proper operation of a CSMA/CD network requires that network size be limited to control round-trip propagation delay as specified in Clause 29.

### 23.9 Environmental specifications

#### 23.9.1 General safety

All equipment meeting this standard shall conform to IEC 60950: 1991.

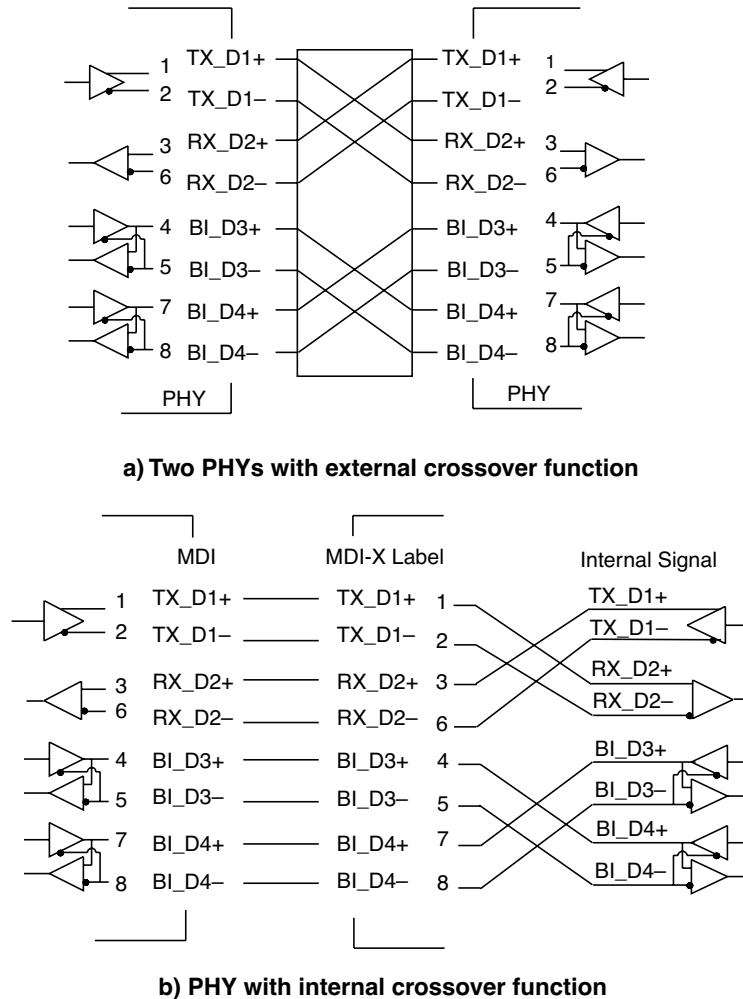
#### 23.9.2 Network safety

This clause sets forth a number of recommendations and guidelines related to safety concerns; the list is neither complete nor does it address all possible safety issues. The designer is urged to consult the relevant local, national, and international safety regulations to ensure compliance with the appropriate requirements.

LAN cable systems described in this clause are subject to at least four direct electrical safety hazards during their installation and use. These hazards are as follows:

- a) Direct contact between LAN components and power, lighting, or communications circuits
- b) Static charge buildup on LAN cables and components
- c) High-energy transients coupled onto the LAN cable system
- d) Voltage potential differences between safety grounds to which various LAN components are connected

Such electrical safety hazards must be avoided or appropriately protected against for proper network installation and performance. In addition to provisions for proper handling of these conditions in an operational



**Figure 23-28—Crossover function**

system, special measures must be taken to ensure that the intended safety features are not negated during installation of a new network or during modification or maintenance of an existing network.

### 23.9.2.1 Installation

It is a mandatory functional requirement that sound installation practice, as defined by applicable local codes and regulations, be followed in every instance in which such practice is applicable.

### 23.9.2.2 Grounding

Any safety grounding path for an externally connected PHY shall be provided through the circuit ground of the MII connection.

#### **WARNING**

It is assumed that the equipment to which the PHY is attached is properly grounded, and not left floating nor serviced by a “doubly insulated, ac power distribution system.” The use of floating or insulated equipment, and the consequent implications for safety, are beyond the scope of this standard.

### 23.9.2.3 Installation and maintenance guidelines

It is a mandatory functional requirement that, during installation and maintenance of the cable plant, care be taken to ensure that noninsulated network cable conductors do not make electrical contact with unintended conductors or ground.

### 23.9.2.4 Telephony voltages

The use of building wiring brings with it the possibility of wiring errors that may connect telephony voltages to 100BASE-T4 equipment. Other than voice signals (which are low voltage), the primary voltages that may be encountered are the “battery” and ringing voltages. Although there is no universal standard, the following maximums generally apply.

Battery voltage to a telephone line is generally 56 Vdc applied to the line through a balanced 400  $\Omega$  source impedance.

Ringing voltage is a composite signal consisting of an ac component and a dc component. The ac component is up to 175 V peak at 20 Hz to 60 Hz with a 100  $\Omega$  source resistance. The dc component is 56 Vdc with a 300  $\Omega$  to 600  $\Omega$  source resistance. Large reactive transients can occur at the start and end of each ring interval.

Although 100BASE-T4 equipment is not required to survive such wiring hazards without damage, application of any of the above voltages shall not result in any safety hazard.

NOTE—Wiring errors may impose telephony voltages differentially across 100BASE-T4 transmitters or receivers. Because the termination resistance likely to be present across a receiver’s input is of substantially lower impedance than an off-hook telephone instrument, receivers will generally appear to the telephone system as off-hook telephones. Therefore, full-ring voltages will be applied for only short periods. Transmitters that are coupled using transformers will similarly appear like off-hook telephones (though perhaps a bit more slowly) due to the low resistance of the transformer coil.

## 23.9.3 Environment

### 23.9.3.1 Electromagnetic emission

The twisted-pair link shall comply with applicable local and national codes for the limitation of electromagnetic interference.

### 23.9.3.2 Temperature and humidity

The twisted-pair link is expected to operate over a reasonable range of environmental conditions related to temperature, humidity, and physical handling (such as shock and vibration). Specific requirements and values for these parameters are considered to be beyond the scope of this standard.

It is recommended that manufacturers indicate in the literature associated with the PHY the operating environmental conditions to facilitate selection, installation, and maintenance.

## 23.10 PHY labeling

It is recommended that each PHY (and supporting documentation) be labeled in a manner visible to the user with at least these parameters:

- a) Data rate capability in Mb/s
- b) Power level in terms of maximum current drain (for external PHYs)
- c) Any applicable safety warnings

See also 23.7.2.

## 23.11 Timing summary

### 23.11.1 Timing references

All MII signals are defined (or corrected to) the DTE end of a zero length MII cable.

NOTE—With a finite length MII cable, TX\_CLK appears in the PHY one cable propagation delay *earlier* than at the MII. This advances the transmit timing. Receive timing is retarded by the same amount.

The phrase *adjusted for pair skew*, when applied to a timing reference on a particular pair, means that the designated timing reference has been adjusted by adding to it the difference between the time of arrival of preamble on the latest of the three receive pairs and the time of arrival of preamble on that particular pair.

#### PMA\_UNITDATA.request

Figures 23–29, 23–30, 23–31, and 23–32. The implementation of this abstract message is not specified. Conceptually, this is the time at which the PMA has been given full knowledge and use of the ternary symbols to be transmitted.

#### PMA\_UNITDATA.indicate

Figure 23–33. The implementation of this abstract message is not specified. Conceptually, this is the time at which the PCS has been given full knowledge and use of the ternary symbols received.

#### WAVEFORM

Figure 23–29. Point in time at which output waveform has moved 1/2 way from previous nominal output level to present nominal output level.

#### TX\_EN

Figure 23–30. First rising edge of TX\_CLK following the rising edge of TX\_EN.

#### NOT\_TX\_EN

Figures 23–31 and 23–32. First rising edge of TX\_CLK following the falling edge of TX\_EN.

#### CRS

Figure 23–33. Rising edge of CRS.

#### CARRIER\_STATUS

Figure 23–33. Rising edge of carrier\_status.

#### NOT\_CARRIER\_STATUS

Figure 23–34. Falling edge of carrier\_status.

#### RX\_DV

No figure. First rising edge of RX\_CLK following rising edge of RX\_DV.

#### COL

No figure. Rising edge of COL signal at MII.

#### NOT\_COL

No figure. Falling edge of COL signal at MII.

#### PMA\_ERROR

No figure. Time at which rxerror\_status changes to ERROR.

### 23.11.2 Definitions of controlled parameters

#### PMA\_OUT

Figure 23–29. Time between PMA\_UNITDATA.request (tx\_code\_vector) and the WAVEFORM timing reference for each of the three transmit channels TX\_D1, BI\_D3, or BI\_D4.

#### TEN\_PMA

Figures 23–30, 23–31, and 23–32. Time between TX\_EN timing reference and MA\_UNITDATA.request (tx\_code\_vector).

#### TEN\_CRIS

Figure 23–30. Time between TX\_EN timing reference and the loopback of TX\_EN to CRS as measured at the CRS timing reference point.

#### NOT\_TEN\_CRIS

Figures 23–31 and 23–32. Time between NOT\_TX\_EN timing reference and the loopback of TX\_EN to CRS as measured at the NOT\_CRIS timing reference point. In the event of a collision (COL is raised at any point during a packet) the minimum time for NOT\_TEN\_CRIS may optionally be as short as 0.

#### RX\_PMA\_CARRIER

Figure 23–33. Time between the WAVEFORM timing reference, adjusted for pair skew, of first pulse of a normal preamble (or first pulse of a preamble preceded by a link test pulse or a partial link test pulse) and the CARRIER\_STATUS timing reference.

#### RX\_CRIS

Figure 23–33. Time between the WAVEFORM timing reference, adjusted for pair skew, of first pulse of a normal preamble (or first pulse of a preamble preceded by a link test pulse or a partial link test pulse) and the CRS timing reference.

NOTE—The input waveform used for this test is an ordinary T4 preamble, generated by a compliant T4 transmitter. As such, the delay between the first and third pulses of the preamble (which are used by the carrier sense logic) is very nearly 80 ns.

#### RX\_NOT\_CRIS

For a data packet, the time between the WAVEFORM timing reference, adjusted for pair skew, of the first pulse of eop1, and the de-assertion of CRS. For a collision fragment, the time between the WAVEFORM timing reference, adjusted for pair skew, of the ternary symbol on pair TX\_D2, which follows the last ternary data symbol received on pair RX\_D2, and the de-assertion of CRS.

Both are limited to the same value. For a data packet, detection of the six ternary symbols of eop1 is accomplished in the PCS layer. For a collision fragment, detection of the concluding seven ternary zeroes is accomplished in the PMA layer, and passed to the PCS in the form of the carrier\_status indication.

#### FAIRNESS

The difference between RX\_NOT\_CRIS at the conclusion of one packet and RX\_CRIS on a subsequent packet. The packets used in this test may arrive with an IPG anywhere in the range of 80 to 160.

#### RX\_PMA\_DATA

Figure 23–33. Time between the WAVEFORM timing reference, adjusted for pair skew, of first pulse of a normal preamble (or first pulse of a preamble preceded by a link test pulse or a partial link test pulse) and the particular PMA\_UNITDATA.indicate that transfers to the PCS the first ternary symbol of the first 6T code group from receive pair BI\_D3.

**EOP\_CARRIER\_STATUS**

Figure 23–34. For a data packet, the time between the WAVEFORM timing reference, adjusted for pair skew, of first pulse of eop1 and the NOT\_CARRIER\_STATUS timing reference.

**EOC\_CARRIER\_STATUS**

Figure 23–35. In the case of a colliding packet, the time between the WAVEFORM timing reference, adjusted for pair skew, of the ternary symbol on pair RX\_D2, which follows the last ternary data symbol received on pair RX\_D2 and the NOT\_CARRIER\_STATUS timing reference.

**RX\_RXDV**

No figure. Time between WAVEFORM timing reference, adjusted for pair skew, of first pulse of a normal preamble (or first pulse of a preamble preceded by a link test pulse or a partial link test pulse) and the RX\_DV timing reference.

**RX\_PMA\_ERROR**

No figure. In the event of a preamble in error, the time between the WAVEFORM timing reference adjusted for pair skew, of first pulse of that preamble (or first pulse of the preamble preceded by a link test pulse or a partial link test pulse), and the PMA\_ERROR timing reference.

**RX\_COL**

No figure. In the event of a collision, the time between the WAVEFORM timing reference adjusted for pair skew, of first pulse of a normal preamble (or first pulse of a preamble preceded by a link test pulse or a partial link test pulse), and the COL timing reference.

**RX\_NOT\_COL**

No figure. In the event of a collision in which the receive signal stops before the locally transmitted signal, the time between the WAVEFORM timing reference adjusted for pair skew, of the ternary symbol on pair RX\_D2, which follows the last ternary data symbol received on pair RX\_D2 and the NOT\_COL timing reference point.

**TX\_NOT\_COL**

No figure. In the event of a collision in which the locally transmitted signal stops before the received signal, the time between the NOT\_TX\_EN timing reference and the loopback of TX\_EN to COL as measured at the NOT\_COL timing reference point.

**TX\_SKEW**

Greatest absolute difference between a) the waveform timing reference of the first pulse of a preamble as measured on output pair TX\_D1; b) the waveform timing reference of the first pulse of a preamble as measured on output pair BI\_D3; and c) the waveform timing reference of the first pulse of a preamble as measured on output pair BI\_D4. Link test pulses, if present during the measurement, must be separated from the preamble by at least 100 ternary symbols.

**CRS\_PMA\_DATA**

Time between the timing reference for CARRIER STATUS and the transferral, via PMA\_UNITDATA.indicate, of the first ternary symbol of the 6T code group marked DATA1 in Figure 23–6.

**COL\_to\_BI\_D3/D4\_OFF**

No figure. In the case of a colliding packet, the time between the WAVEFORM timing reference, adjusted for pair skew, of the first pulse of preamble (or the first pulse of the preamble preceded by a link test pulse or a partial link test pulse) on RX\_D2, and the first ternary zero transmitted on BI\_D3 and on BI\_D4.

NOTE—Subclause 23.4.1.2 mandates that transmission on pairs BI\_D3 and BI\_D4 be halted in the event of a collision.

### 23.11.3 Table of required timing values

While in the LINK\_PASS state, each PHY timing parameter shall fall within the Low and High limits listed in Table 23–7. All units are in bit times. A bit time equals 10 ns.

**Table 23–7—Required timing values**

Controlled parameter	Low limit (bits)	High limit (bits)	Comment
PMA_OUT	1	9.5	
TEN_PMA + PMA_OUT	7	17.5	
TEN_CRIS	0	+4	
NOT_TEN_CRIS	0	36	
RX_PMA_CARRIER	0	15.5	
RX_CRIS	0	27.5	
RX_NOT_CRIS	0	51.5	
FAIRNESS	0	28	
RX_PMA_DATA	67	90.5	
EOP_CARRIER_STATUS	51	74.5	
EOC_CARRIER_STATUS	3	50.5	
RX_RXDV	81	114.5	
RX_PMA_ERROR	RX_PMA_DATA	RX_PMA_DATA + 20	Allowed limits equal the actual RX_PMA_DATA time for the device under test plus from 0 to 20 BT
RX_COL	0	27.5	SAME AS RX_CRIS
RX_NOT_COL	0	51.5	SAME AS RX_NOT_CRIS
TX_NOT_COL	0	36	
TX_SKEW	0	0.5	
CRS_PMA_DATA	0	78.5	
COL_to_BI_D3/D4_OFF	0	40	

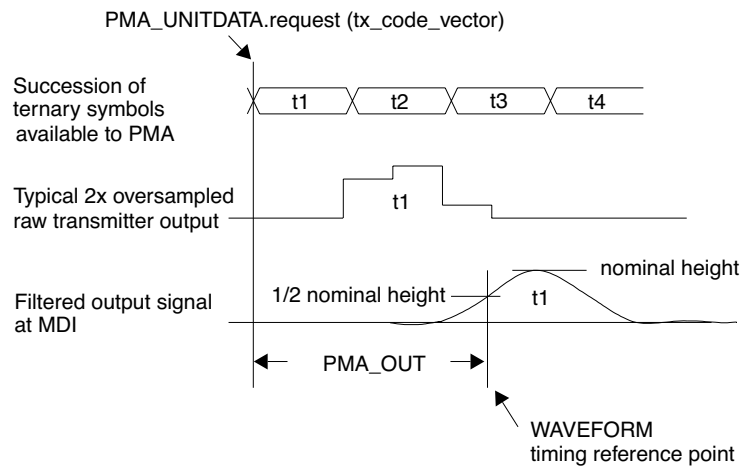


Figure 23–29—PMA TRANSMIT timing while tx\_code\_vector = DATA

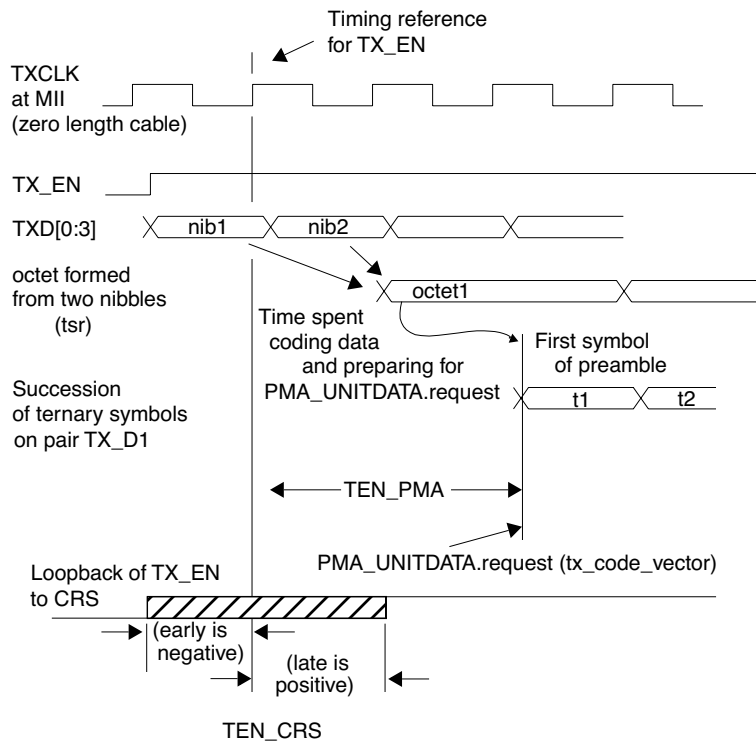


Figure 23–30—PCS TRANSMIT timing at start of packet



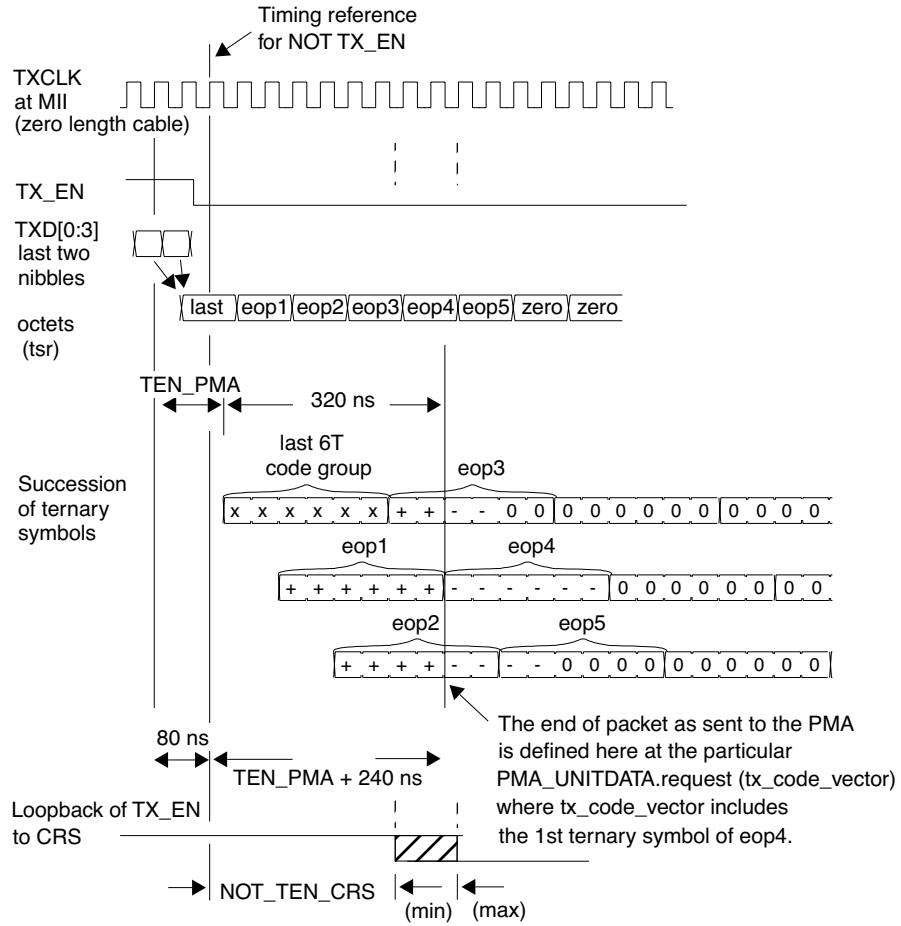


Figure 23-31 – PCS TRANSMIT timing end of normal packet

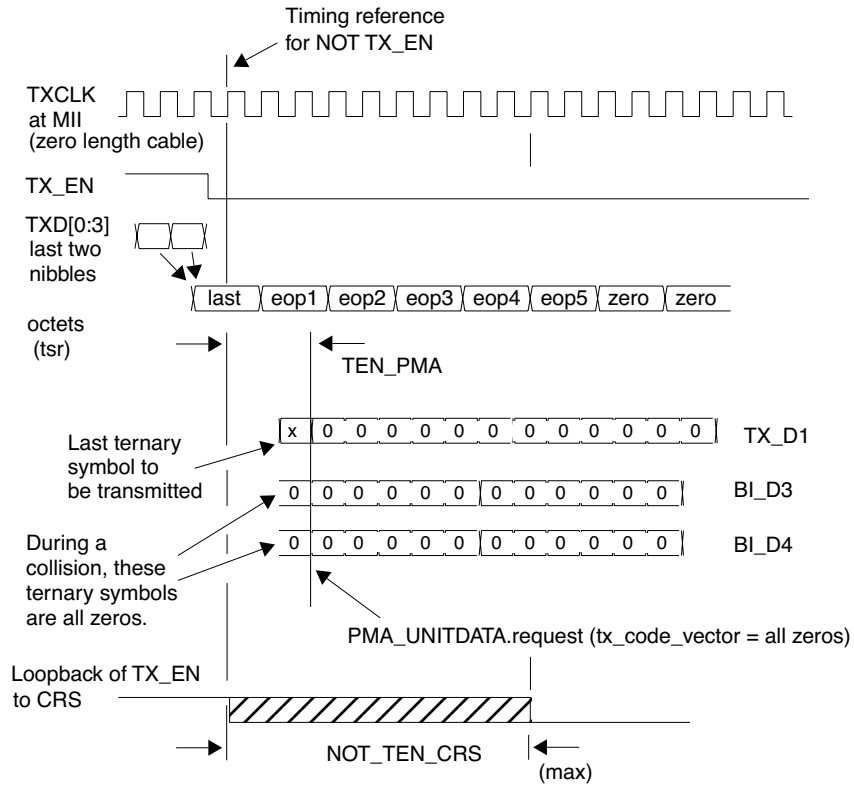


Figure 23-32—PCS TRANSMIT timing end of colliding packet

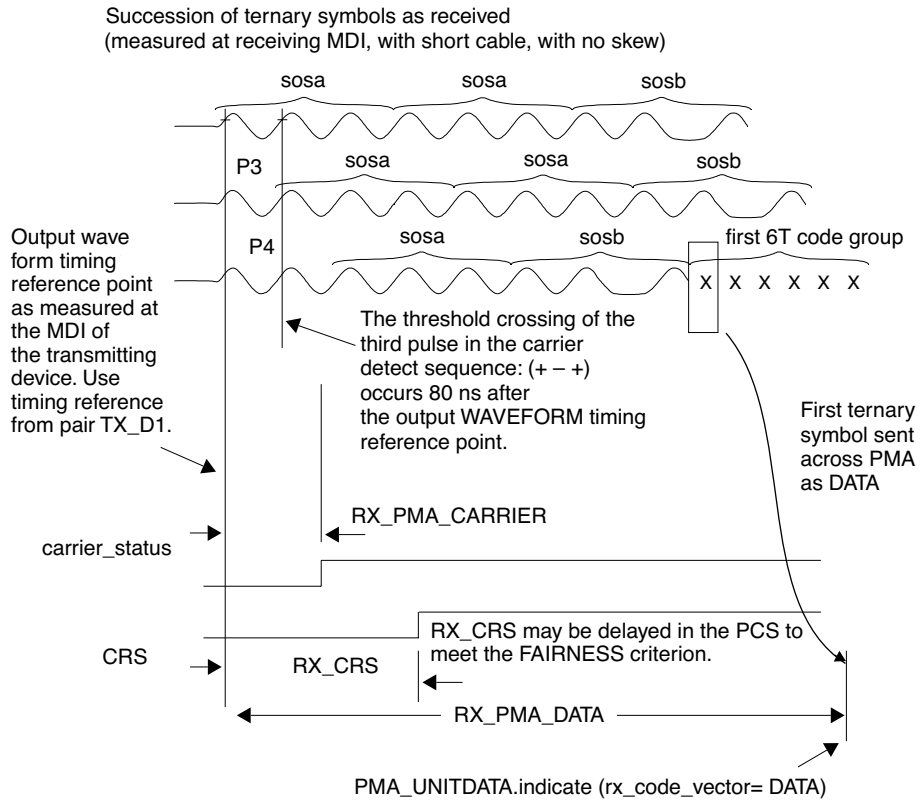
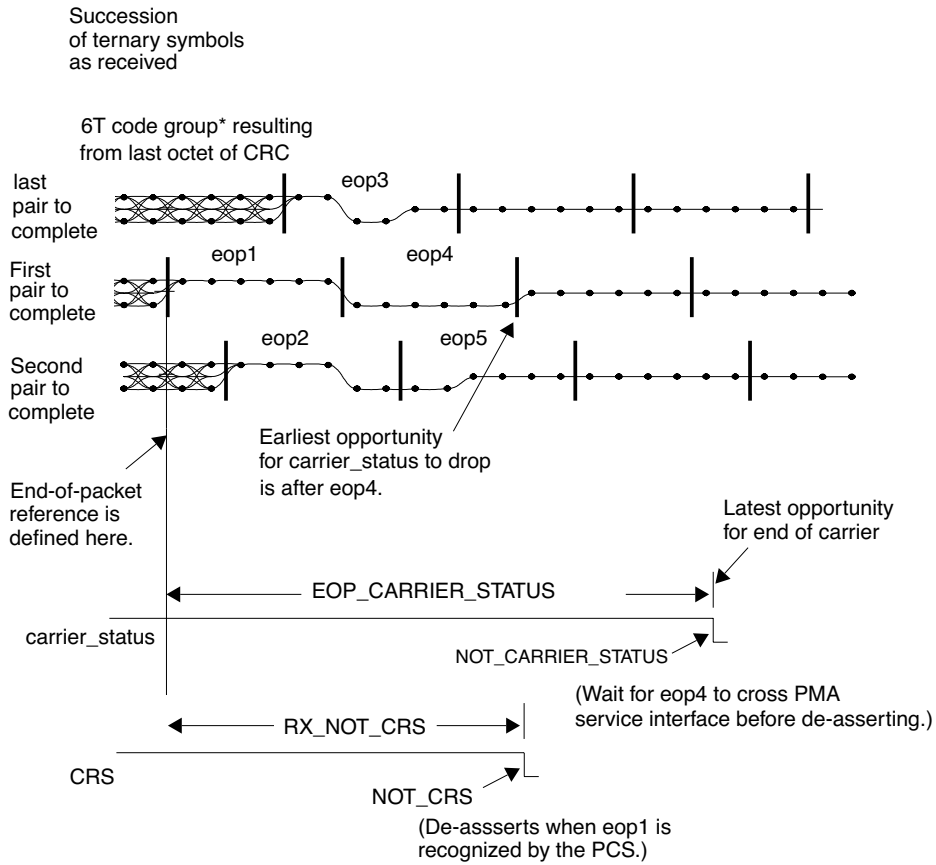


Figure 23-33—PMA RECEIVE timing start of packet



\*RX\_DV de-asserts after sending the last nibble of this decoded octet across the MII. CRS may de-assert prior to that time.

**Figure 23-34—PMA RECEIVE timing end of normal packet**

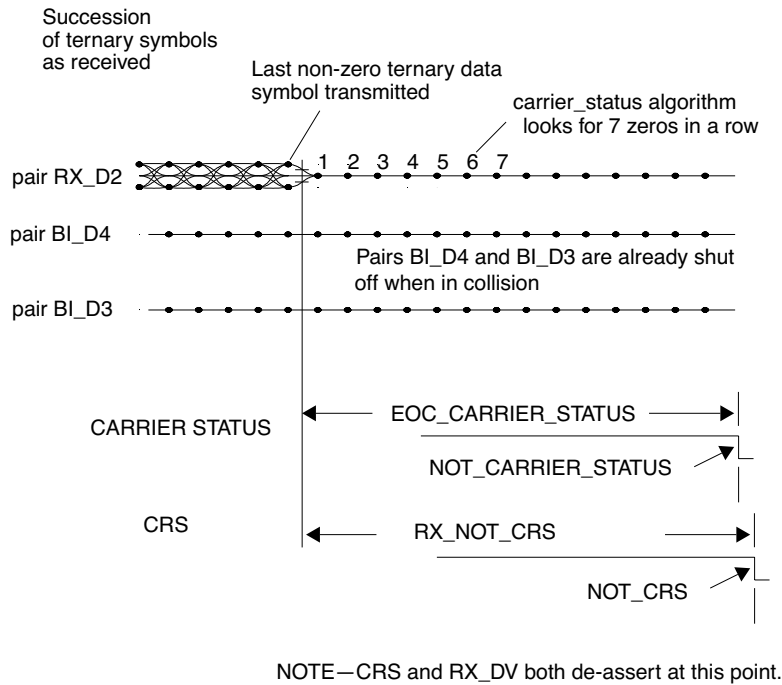


Figure 23–35—PMA RECEIVE timing end of colliding packet

### 23.12 Protocol Implementation Conformance Statement (PICS) proforma for Clause 23, Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) sublayer, and baseband medium, type 100BASE-T4<sup>4</sup>

#### 23.12.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 23, Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) sublayer, and baseband medium, type 100BASE-T4, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

<sup>4</sup>Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

**23.12.2 Identification****23.12.2.1 Implementation identification**

Supplier	
Contact point for enquiries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s)	
<p>NOTES</p> <p>1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.</p> <p>2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).</p>	

**23.12.2.2 Protocol summary**

Identification of protocol standard	IEEE Std 802.3-2002 <sup>®</sup> , Clause 23, Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) sublayer, and baseband medium, type 100BASE-T4
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
<p>Have any Exception items been required? No [ ] Yes [ ]          (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3-2002<sup>®</sup>.)</p>	

Date of Statement	
-------------------	--

### 23.12.3 Major capabilities/options

Item	Feature	Subclause	Status	Support	Value/Comment
*MII	Exposed MII interface	23.1.5.3	O		Devices supporting this option must also support the PCS option
*PCS	PCS functions	23.1.5.2	O		Required for integration with DTE or MII
*PMA	Exposed PMA service interface	23.1.5.2	O		Required for integration into symbol level repeater core
*XVR	Internal wiring crossover	23.7.2	O		Usually implemented in repeater, usually not in DTE
*NWY	Support for optional Auto-Negotiation (Clause 28)	23.1.5.4	O		Required if Auto-Negotiation is implemented
*INS	Installation / cable		O		Items marked with INS include installation practices and cable specifications not applicable to a PHY manufacturer

### 23.12.4 PICS proforma tables for the Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) sublayer and baseband medium, type 100BASE-T4

#### 23.12.4.1 Compatibility considerations

Item	Feature	Subclause	Status	Support	Value/Comment
CCO-1	Compatibility at the MDI	23.1.5.1	M		

#### 23.12.4.2 PCS Transmit functions

Item	Feature	Subclause	Status	Support	Value/Comment
PCT-1	PCS Transmit function	23.2.1.2	PCS:M		Complies with state diagram Figure 23-8
PCT-2	Data encoding	23.2.1.2	PCS:M		8B6T with DC balance encoding rules
PCT-3	Order of ternary symbol transmission	Annex 23A	PCS:M		Leftmost symbol of each 6T code group first

**23.12.4.3 PCS Receive functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PCR1	PCS Receive function	23.2.1.3	PCS:M		Complies with state diagram Figure 23–9
PCR2	Value of RXD<3:0> while RXDV is de-asserted	23.2.1.3	PCS:M		All zeroes
PCR3	Data decoding	23.2.1.3	PCS:M		8B6T with error detecting rules
PCR4	Value of dc_balance_error, eop_error and codeword_error at times other than those specified in the error detecting rules.	23.2.1.3	PCS:M		OFF
PCR5	Codeword_error indication sets RX_ER when	23.2.1.3	PCS:M		During transfer of both affected data nibbles across the MII
PCR6	Dc_balance_error sets RX_ER when	23.2.1.3	PCS:M		During transfer of both affected nibbles across the MII
PCR7	Eop_error sets RX_ER when	23.2.1.3	PCS:M		During transfer of last decoded data nibble across the MII
PCR8	Action taken if carrier_status is truncated dur to early de-assertion of carrier_status	23.2.1.3	PCS:M		Assert RX_ER, and then de-assert RX_DV

**23.12.4.4 Other PCS functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PCO1	PCS Reset function executed when	23.2.1.1	PCS:M		Power-on, or the receipt of a reset request from the management entity
PCO2	PCS Error Sense function	23.2.1.4	PCS:M		Complies with state diagram Figure 23–10
PCO3	Signaling of RX_ER to MII	23.2.1.4	PCS:M		Before last nibble of Clause 4 MAC frame has passed across MII
PCO4	Timing of rxerror_status	23.2.1.4	PCS:M		Causes RX_ER to appear on the MII no later than last nibble of first data octet
PCO5	PCS Carrier Sense function	23.2.1.5	PCS:M		Controls MII signal CRS according to rules in 23.2.1.5
PCO6	MII signal COL is asserted when	23.2.1.6	PCS:M		Upon detection of a PCS collision
PCO7	At other times COL remains	23.2.1.6	PCS:M		De-asserted
PCO8	Loopback implemented in accordance with 22.4.1.2	23.2.2.2	PCS:M		Redundantly specified in 22.2.4.1.2



**23.12.4.4 Other PCS functions (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
PCO9	No spurious signals emitted on the MDI during or after power down	23.2.2.2	M		
PCO10	PMA frame structure	23.2.3	M		Conformance to Figure 23–6
PCO11	PMA_UNITDATA messages	23.2.3	PMA:M		Must have a clock for both directions

**23.12.4.5 PCS state diagram variables**

Item	Feature	Subclause	Status	Support	Value/Comment
PCS1	Timing of eop adjusted such that the last nibble sent across the MII with RX_DV asserted is	23.2.4.2	PCS:M		Last nibble of last decoded data octet in a packet
PCS2	Transmission of octets on the three transmit pairs	23.2.4.2	PCS:M		Transmission order is: TX_D1, then BI_D3, and then BI_D4
PCS3	Value of tsr during first 16 TX_CLK cycles after TX_EN is asserted	23.2.4.2	PCS:M		sosa, sosa, sosa, sosa, sosa, sosa, sosa, sosa, sosa, sosa, sosb, sosb, sosb, sosb, sosb, sosb
PCS4	Value of tsr during first 10 TX_CLK cycles after TX_EN is de-asserted	23.2.4.2	PCS:M		eop1, eop1, eop2, eop2, eop3, eop3, eop4, eop4, eop5, eop5
PCS5	TX_ER causes transmission of	23.2.4.2	PCS:M		bad_code
PCS6	TX_ER received during the first 16 TX_CLK cycles causes	23.2.4.2	PCS:M		Transmission of <b>bad_code</b> during 17th and 18th clock cycles
PCS7	Action taken in event TX_EN falls on an odd nibble boundary	23.2.4.2	PCS:M		Extension of TX_EN by one TX_CLK cycle, and transmission of bad_code
PCS8	Transmission when TX_EN is not asserted	23.2.4.2	PCS:M		zero_code
PCS9	TX_CLK generated synchronous to	23.2.4.2	PCS:M		tw1_timer

**23.12.4.6 PMA service interface**

Item	Feature	Subclause	Status	Support	Value/Comment
PMS1	Continuous generation of PMA_TYPE	23.3.1.2	M		
PMS2	Generation of PMA_UNITDATA.indicate (DATA) messages	23.3.3.2	M		synchronous with data received at the MDI
PMS3	Generation of PMA_CARRIER.indicate message	23.3.4.2	M		ON/OFF
PMS4	Generation of PMA_LINK.indicate message	23.3.5.2	M		FAIL/READY/OK
PMS5	Link_control defaults on power-on or reset to	23.3.6.2	M		ENABLE
PMS6	Action taken in SCAN_FOR_CARRIER mode	23.3.6.4	NWY:M		Enables link integrity state diagram, but blocks passage into LINK_PASS
PMS7	Reporting of link_status while in SCAN_FOR_CARRIER mode	23.3.6.4	NWY:M		FAIL / READY
PMS8	Reporting of link_status while in DISABLE mode	23.3.6.4	NWY:M		FAIL
PMS9	Action taken in ENABLE mode	23.3.6.4	NWY:M		enables data processing functions
PMS10	Generation of PMA_RXERROR	23.3.7.2	M		ERROR / NO_ERROR

**23.12.4.7 PMA Transmit functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PMT1	Transmission while (tx_code_vector=DATA) * (pma_carrier=OFF)	23.4.1.2	M		tx_code_vector[TX_D1] tx_code_vector[BI_D3] tx_code_vector[BI_D4]
PMT2	Transmission from time (tx_code_vector=DATA) * (pma_carrier=ON), until (tx_code_vector=IDLE)	23.4.1.2	M		tx_code_vector[TX_D1] CS0 CS0
PMT3	Transmission while tx_code_vector=IDLE	23.4.1.2	M		Idle signal TP_DIL_100
PMT4	Duration of silence between link test pulses	23.4.1.2	M		1.2 ms ± 0.6 ms
PMT5	Link test pulse composed of	23.4.1.2	M		CS-1, CS1 transmitted on TX_D1

**23.12.4.7 PMA Transmit functions (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
PMT6	Following a packet, TP_IDL_100 signal starts with	23.4.1.2	M		Period of silence
PMT7	Effect of termination of TP_IDL_100	23.4.1.2	M		No delay or corruption of subsequent packet
PMT8	Zero crossing jitter of link test pulse	23.4.1.2	M		Less than 4 ns p-p
PMT9	Action taken when xmit=disable	23.4.1.2	M		Transmitter behaves as if tx_code_vector=IDLE

**23.12.4.8 PMA Receive functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PMR1	Reception and translation of data with ternary symbol error rate less than	23.4.1.3	M		One part in 10 <sup>8</sup>
PMR2	Assertion of pma_carrier=ON upon reception of test signal	23.4.1.4	M		Test signal is a succession of three data values, produced synchronously with a 25 MHz clock, both preceded and followed by 100 symbols of silence. The three values are: 467 mV, -225 mV, and then 467 mV again
PMR3	condition required to turn off pma_carrier	23.4.1.4	M		Either of a) Seven consecutive zeroes b) Reception of eop1 per 23.4.1.4
PMR4	Value of carrier_status while rcv=ENABLE	23.4.1.4	M		pma_carrier
PMR5	Value of carrier_status while rcv=DISABLE	23.4.1.4	M		OFF

**23.12.4.9 Link Integrity functions**

Item	Feature	Subclause	Status	Support	Value/Comment
LIF1	Link Integrity function complies with	23.4.1.5	M		State diagram Figure 23-12

**23.12.4.10 PMA Align functions**

Item	Feature	Subclause	Status	Support	Value/Comment
ALN1	Generation of PMA_UNITDATA.indicate (PREAMBLE) messages	23.4.1.6	M		
ALN2	Ternary symbols transferred by first PMA_UNITDATA.indicate (DATA) message	23.4.1.6	M		rx_code_vector[BI_D3]: first ternary symbol of first data code group rx_code_vector[BI_D2]: two ternary symbols prior to start of second data code group rx_code_vector[BI_D4]: four ternary symbols prior to start of third data code group
ALN3	PMA_UNITDATA.indicate (DATA) messages continue until carrier_status=OFF	23.4.1.6	M		
ALN4	While carrier_status=OFF, PMA emits message	23.4.1.6	M		PMA_UNITDATA.indicate (IDLE)
ALN5	Failure to recognize SSD generates rxerror_status=ERROR	23.4.1.6	M		
ALN6	Action taken when carrier_status=OFF	23.4.1.6	M		Clear rxerror_status
ALN7	Action taken if first packet is used for alignment	23.4.1.6	M		PMA emits PMA_UNITDATA.indicate (PREAMBLE)
ALN8	Tolerance of line skew	23.4.1.6	M		60 ns
ALN9	Detection of misplaced <b>sosb</b> 6T code group caused by 3 or fewer ternary symbols in error	23.4.1.6	M		
ALN10	Action taken if <b>rcv</b> =disable	23.4.1.6	M		PMA emits PMA_UNITDATA..indicate (IDLE)

**23.12.4.11 Other PMA functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PMO1	PMA Reset function	23.4.1.1	M		
PMO2	Suitable clock recovery	23.4.1.7	M		

**23.12.4.12 Isolation requirements**

Item	Feature	Subclause	Status	Support	Value/Comment
ISO1	Values of all components used in test circuits	23.5	M		Accurate to within $\pm 1\%$ unless required otherwise
ISO2	Electrical isolation meets	23.5.1.1	M		1500 V at 50–60 Hz for 60 s per IEC 60950: 1991 <i>or</i> 2250 Vdc for 60 s per IEC 60950: 1991 <i>or</i> Ten 2400 V pulses per IEC 60060
ISO3	Insulation breakdown during isolation test	23.5.1.1	M		None per IEC 60950: 1991
ISO4	Resistance after isolation test	23.5.1.1	M		At least 2 M $\Omega$

**23.12.4.13 PMA electrical requirements**

Item	Feature	Subclause	Status	Support	Value/Comment
PME1	Conformance to all transmitter specifications in 23.5.1.2	23.5.1.2	M		
PME2	Transmitter load unless otherwise specified	23.5.1.2	M		100 $\Omega$
PME3	Peak differential output voltage	23.5.1.2.1	M		3.15–3.85 V
PME4	Differential transmit template at MDI	23.5.1.2.2	M		Table 23–2
PME5	Differential MDI output template voltage scaling	23.5.1.2.2	M		3.15– 3.85 V
PME6	Interpolation between points on transmit template	23.5.1.2.2	M		Linear
PME7	Differential link pulse template at MDI	23.5.1.2.2	M		Table 23–2
PME8	Differential link pulse template scaling	23.5.1.2.2	M		Same value as used for differential transmit template scaling
PME9	Interpolation between point on link pulse template	23.5.1.2.2	M		Linear
PME10	State when transmitting seven or more consecutive CS0 during TP_IDL-100 signal	23.5.1.2.2	M		–50 mV to 50 mV
PME11	Limit on magnitude of harmonics measured at MDI	23.5.1.2.2	M		27 dB below fundamental
PME12	Differential output ISI	23.5.1.2.3	M		Less than 9%

**23.12.4.13 PMA electrical requirements (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
PME13	Measurement of ISI and peak-to-peak signal voltage	23.5.1.2.3	M		Halfway between nominal zero crossing of the observed eye pattern
PME14	Transfer function of 100BASE-T4 transmit test filter	23.5.1.2.3	M		Third-order Butterworth filter with -3 dB point at 25.0 MHz
PME15	Reflection loss of 100BASE-T4 transmit test filter and 100 W load across the frequency range 2–12.5 MHz	23.5.1.2.3	M		Exceeds 17 dB
PME16	Differential output impedance	23.5.1.2.4	M		Provide return loss into 100 Ω of 17 dB from 2.0 to 12.5 MHz
PME17	Maintenance of return loss	23.5.1.2.4	M		At all times PHY is fully powered
PME18	Droop as defined in Figure 23-18 during transmission of eop1 and eop4	23.5.1.2.4	M		Less than 6%
PME19	Output timing jitter	23.5.1.2.5	M		No more than 4 ns peak-to-peak
PME20	Measurement of output timing jitter	23.5.1.2.5	M		Other transmit outputs connected to 100BASE-T4 ISI test filter or 100 Ω load
PME21	Minimum transmitter impedance balance	23.5.1.2.6	M		$29 - 17 \log\left(\frac{f}{10}\right)$ dB
PME22	Transmitter common-mode rejection; effect of $E_{cm}$ as shown in Figure 23-20 upon $E_{dif}$	23.5.1.2.8	M		Less than 100 mV
PME23	Transmitter common-mode rejection; effect of $E_{cm}$ as shown in Figure 23-20 upon edge jitter	23.5.1.2.8	M		Less than 1.0 ns
PME24	$E_{cm}$ used for common-mode rejection tests	23.5.1.2.8	M		15 V peak, 10.1 MHz sine wave
PME25	Transmitter faults; response to indefinite application of short circuits	23.5.1.2.9	M		Withstand without damage and resume operation after fault is removed
PME26	Transmitter faults; response to 1000 V common-mode impulse per IEC 60060	23.5.1.2.9	M		Withstand without damage
PME27	Shape of impulse used for common-mode impulse test	23.5.1.2.9	M		0.3/50 μs as defined in IEC 60060
PME28	Ternary symbol transmission rate	23.5.1.2.10	M		25.000 MHz ± 0.01%

**23.12.4.13 PMA electrical requirements (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
PME29	Conformance to all receiver specifications in 23.5.1.3	23.5.1.3	M		
PME30	Action taken upon receipt of differential signals that were transmitted within the constraints of 23.5.1.2 and have passed through worst-case UTP model	23.5.1.3.1	M		Correctly translated into PMA_UNITDATA messages
PME31	Action taken upon receipt of link test pulse	23.5.1.3.1	M		Accept as a link test pulse
PME32	Test configuration for data reception and link test pulse tests	23.5.1.3.1	M		Using worst-case UTP model, and with a connection less than one meter in length
PME33	Bit loss	23.5.1.3.2	M		No more than that specified in 23.5.1.3.1
PME34	Reaction of pma_carrier to signal less than 325 mV peak	23.5.1.3.2	M		Must not set pma_carrier=ON
PME35	Reaction of pma_carrier to continuous sinusoid less than 1.7 MHz	23.5.1.3.2	M		Must not set pma_carrier=ON
PME36	Reaction of pma_carrier to single cycle or less	23.5.1.3.2	M		Must not set pma_carrier=ON
PME37	Reaction of pma_carrier to fast link pulse as defined in Clause 28	23.5.1.3.2	M		Must not set pma_carrier=ON
PME38	Reaction of pma_carrier to link integrity test pulse signal TP_IDL_100	23.5.1.3.2	M		Must not set pma_carrier=ON
PME39	Differential input impedance	23.5.1.3.3	M		Provide return loss into 100 $\Omega$ of 17 dB from 2.0 to 12.5 MHz
PME40	Maintenance of return loss	23.5.1.3.3	M		At all times PHY is fully powered
PME41	Droop as defined in Figure 23–18 during reception of test signal defined in Figure 23–19	23.5.1.3.3	M		Less than 6%
PME42	Receiver common-mode rejection; effect of $E_{cm}$ as shown in Figure 23–24	23.5.1.3.4	M		Receiver meets 23.5.1.3.1
PME43	$E_{cm}$ used for common-mode rejection tests	23.5.1.3.4	M		25 V peak-to-peak square wave, 500 kHz or lower in frequency, with edges no slower than 4 ns
PME44	Receiver faults; response to indefinite application of short circuits	23.5.1.3.5	M		Withstand without damage and resume operation after fault is removed

**23.12.4.13 PMA electrical requirements (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
PME45	Receiver faults; response to 1000 V common mode impulse per IEC 60060	23.5.1.3.5	M		Withstand without damage
PME46	Shape of impulse used for common mode impulse test	23.5.1.3.5	M		0.3/50 $\mu$ s as defined in IEC 60060
PME47	Receiver properly receives data have a worst-case ternary symbol range	23.5.1.3.6	M		25.00 MHz $\pm$ 0.01%
PME48	Steady-state current consumption	23.5.2	MII:M		0.75 A maximum
PME49	PHY operating voltage range	23.5.2	MII:M		Includes worst voltage available from MII
PME50	Extraneous signals induced on the MII control circuits during normal power-up and power-down	23.5.2	M		None

**23.12.4.14 Characteristics of the link segment**

Item	Feature	Subclause	Status	Support	Value/Comment
LNK1	Cable used	23.6.1	INS:M		Four pairs of balanced cabling, Category 3 or better, with a nominal characteristic impedance of 100 $\Omega$
LNK2	Source and load impedance used for cable testing (unless otherwise specified)	23.6.2	INS:M		100 $\Omega$
LNK3	Insertion loss of simplex link segment	23.6.2.1	INS:M		Less than 12 dB
LNK4	Source and load impedances used to measure cable insertion loss	23.6.2.1	INS:M		Meet 23.5.1.2.4 and 23.5.1.3.3
LNK5	Characteristic impedance over the range 2–12.5 MHz	23.6.2.2	INS:M		85–115 $\Omega$
LNK6	NEXT loss between 2 and 12.5 MHz	23.6.2.3.1	INS:M		Greater than $24.5 - 15 \log\left(\frac{f}{12.5}\right)$ dB
LNK7	MDNEXT loss between 2 and 12.5 MHz	23.6.2.3.2	INS:M		Greater than $21.4 - 15 \log\left(\frac{f}{12.5}\right)$ dB



**23.12.4.14 Characteristics of the link segment (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
LNK8	ELFEXT loss between 2 and 12.5 MHz	23.6.2.3.3	INS:M		Greater than $23.1 - 15 \log\left(\frac{f}{12.5}\right)$ dB
LNK9	MDELFEEXT loss between 2 and 12.5 MHz	23.6.2.3.4	INS:M		Greater than $20.9 - 15 \log\left(\frac{f}{12.5}\right)$ dB
LNK10	Propagation delay	23.6.2.4.1	INS:M		Less than 570 ns
LNK11	Propagation delay per meter	23.6.2.4.2	INS:M		Less than 5.7 ns/m
LNK12	Skew	23.6.2.4.3	INS:M		Less than 50 ns
LNK13	Variation in skew once installed	23.6.2.4.3	INS:M		Less than $\pm 10$ ns, within constraint of LNK8
LNK14	Noise level	23.6.3	INS:M		Such that objective error rate is met
LNK15	MDNEXT noise	23.6.3.1	INS:M		Less than 325 mVp
LNK16	MDFEEXT noise	23.6.3.2	INS:M		Less than 87 mVp
LNK17	Maximum length of Category 5, 25-pair jumper cables	23.6.3.2	INS:M		10 m

**23.12.4.15 MDI requirements**

Item	Feature	Subclause	Status	Support	Value/Comment
MDI1	MDI connector	23.7.1	M		IEC 60603-7: 1990
MDI2	Connector used on PHY	23.7.1	M		Jack (as opposed to plug)
MDI3	Crossover in every twisted-pair link	23.7.2	INS:M		
MDI4	MDI connector that implements the crossover function	23.7.2	XVR:M		Marked with "X"

**23.12.4.16 General safety and environmental requirements**

Item	Feature	Subclause	Status	Support	Value/Comment
SAF1	Conformance to safety specifications	23.9.1	M		IEC 60950: 1991
SAF2	Installation practice	23.9.2.1	INS:M		Sound practice, as defined by applicable local codes
SAF3	Any safety grounding path for an externally connected PHY shall be provided through the circuit ground of the MII connection	23.9.2.2	M		
SAF4	Care taken during installation to ensure that noninsulated network cable conductors do not make electrical contact with unintended conductors or ground	23.9.2.3	INS:M		
SAF5	Application of voltages specified in 23.9.2.4 does not result in any safety hazard	23.9.2.4	M		
SAF6	Conformance with local and national codes for the limitation of electromagnetic interference	23.9.3.1	INS:M		

**23.12.4.17 Timing requirements**

Item	Feature	Subclause	Status	Support	Value/Comment
TIM1	PMA_OUT	23.11.3	PMA:M		1 to 9.5 BT
TIM2	TEN_PMA + PMA_OUT	23.11.3	PCS:M		7 to 17.5 BT
TIM3	TEN_CRIS	23.11.3	PCS:M		0 to +4 BT
TIM4	NOT_TEN_CRIS	23.11.3	PCS:M		28 to 36 BT
TIM5	RX_PMA_CARRIER	23.11.3	PMA:M		Less than 15.5 BT
TIM6	RX_CRIS	23.11.3	PCS:M		Less than 27.5 BT
TIM7	RX_NOT_CRIS	23.11.3	PCS:M		0 to 51.5 BT
TIM8	FAIRNESS	23.11.3	PCS:M		0 to 28 BT
TIM9	RX_PMA_DATA	23.11.3	PMA:M		67 to 90.5 BT
TIM10	EOP_CARRIER_STATUS	23.11.3	M		51 to 74.5 BT
TIM11	EOC_CARRIER_STATUS	23.11.3	M		3 to 50.5 BT
TIM12	RX_RXDV	23.11.3	PCS:M		81 to 114.5 BT
TIM13	RX_PMA_ERROR	23.11.3	M		Allowed limits equal the actual RX_PMA_DATA time for the device under test plus from 0 to 20 BT
TIM14	RX_COL	23.11.3	PCS:M		Less than 27.5 BT
TIM15	RX_NOT_COL	23.11.3	PCS:M		Less than 51.5 BT
TIM16	TX_NOT_COL	23.11.3	PCS:M		Less than 36 BT
TIM17	TX_SKEW	23.11.3	M		Less than 0.5 BT
TIM18	CRS_PMA_DATA	23.11.3	PMA:M		Less than 78.5 BT
TIM19	COL_to_BI_D3/4_OFF	23.11.3	PMA:M		Less than 40 BT

## 24. Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 100BASE-X

### 24.1 Overview

#### 24.1.1 Scope

This clause specifies the Physical Coding Sublayer (PCS) and the Physical Medium Attachment (PMA) sublayer that are common to a family of 100 Mb/s Physical Layer implementations, collectively known as 100BASE-X. There are currently two embodiments within this family: 100BASE-TX and 100BASE-FX. 100BASE-TX specifies operation over two copper media: two pairs of shielded twisted-pair cable (STP) and two pairs of unshielded twisted-pair cable (Category 5 UTP).<sup>5</sup> 100BASE-FX specifies operation over two optical fibers. The term 100BASE-X is used when referring to issues common to both 100BASE-TX and 100BASE-FX.

100BASE-X leverages the Physical Layer standards of ISO/IEC 9314 and ANSI X3T12 (FDDI) through the use of their Physical Medium Dependent (PMD) sublayers, including their Medium Dependent Interfaces (MDI). For example, ANSI X3.263-1995 (TP-PMD) defines a 125 Mb/s, full duplex signaling system for twisted-pair wiring that forms the basis for 100BASE-TX as defined in Clause 25. Similarly, ISO/IEC 9314-3:1990 defines a system for transmission on optical fiber that forms the basis for 100BASE-FX as defined in Clause 26.

100BASE-X maps the interface characteristics of the FDDI PMD sublayer (including MDI) to the services expected by the CSMA/CD MAC. 100BASE-X can be extended to support any other full duplex medium requiring only that the medium be PMD compliant.

#### 24.1.2 Objectives

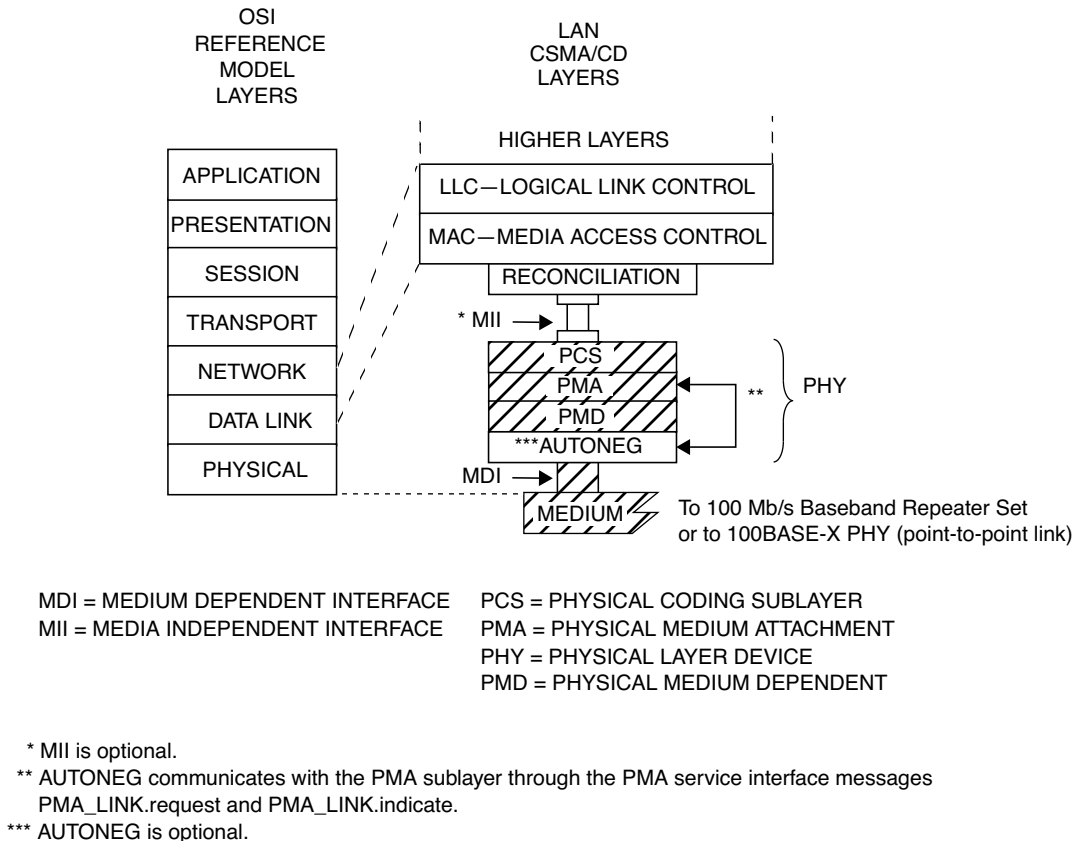
The following are the objectives of 100BASE-X:

- a) Support the CSMA/CD MAC in the half duplex and the full duplex modes of operation.
- b) Support the 100BASE-T MII, repeater, and optional Auto-Negotiation.
- c) Provide 100 Mb/s data rate at the MII.
- d) Support cable plants using Category 5 UTP, 150  $\Omega$  STP or optical fiber, compliant with ISO/IEC 11801.
- e) Allow for a nominal network extent of 200–400 m, including
  - 1) Unshielded twisted-pair links of 100 m;
  - 2) Two repeater networks of approximately 200 m span;
  - 3) One repeater network of approximately 300 m span (using fiber); and
  - 4) DTE/DTE links of approximately 400 m (half duplex mode using fiber) and 2 km (full duplex mode using multimode fiber).
- f) Preserve full duplex behavior of underlying PMD channels.

#### 24.1.3 Relationship of 100BASE-X to other standards

Figure 24–1 depicts the relationships among the 100BASE-X sublayers (shown shaded), other 100BASE-T sublayers, the CSMA/CD MAC, and the IEEE 802.2<sup>®</sup> LLC.

<sup>5</sup>ISO/IEC 11801 makes no distinction between shielded or unshielded twisted-pair cables, referring to both as balanced cables.



**Figure 24–1 — Type 100BASE-X PHY relationship to the ISO/IEC Open Systems Interconnection (OSI) reference model and the IEEE 802.3<sup>®</sup> CSMA/CD LAN model**

#### 24.1.4 Summary of 100BASE-X sublayers

The following provides an overview of the 100BASE-X sublayers that are embodied in the 100BASE-X Physical sublayer (PHY).<sup>6</sup>

##### 24.1.4.1 Physical Coding Sublayer (PCS)

The PCS interface is the Media Independent Interface (MII) that provides a uniform interface to the Reconciliation sublayer for all 100BASE-T PHY implementations (e.g., 100BASE-X and 100BASE-T4). 100BASE-X, as other 100BASE-T PHYs, is modeled as providing services to the MII. This is similar to the use of an AUI interface.

The 100BASE-X PCS realizes all services required by the MII, including:

- Encoding (decoding) of MII data nibbles to (from) five-bit code-groups (4B/5B);
- Generating Carrier Sense and Collision Detect indications;
- Serialization (deserialization) of code-groups for transmission (reception) on the underlying serial PMA, and
- Mapping of Transmit, Receive, Carrier Sense and Collision Detection between the MII and the underlying PMA.

<sup>6</sup> The 100BASE-X PHY should not be confused with the FDDI PHY, which is a sublayer functionally aligned to the 100BASE-T PCS.

### 24.1.4.2 Physical Medium Attachment (PMA) sublayer

The PMA provides a medium-independent means for the PCS and other bit-oriented clients (e.g., repeaters) to support the use of a range of physical media. The 100BASE-X PMA performs the following functions:

- a) Mapping of transmit and receive code-bits between the PMA's client and the underlying PMD;
- b) Generating a control signal indicating the availability of the PMD to a PCS or other client, also synchronizing with Auto-Negotiation when implemented;
- c) Optionally, generating indications of activity (carrier) and carrier errors from the underlying PMD;
- d) Optionally, sensing receive channel failures and transmitting the Far-End Fault Indication; and detecting the Far-End Fault Indication; and
- e) Recovery of clock from the NRZI data supplied by the PMD.

### 24.1.4.3 Physical Medium Dependent (PMD) sublayer

100BASE-X uses the FDDI signaling standards ISO/IEC 9314-3: 1990 and ANSI X3.263-1995 (TP-PMD). These signaling standards, called PMD sublayers, define 125 Mb/s, full duplex signaling systems that accommodate multi-mode optical fiber, STP and UTP wiring. 100BASE-X uses the PMDs specified in these standards with the PMD Service Interface specified in 24.4.1.

The MDI, logically subsumed within the PMD, provides the actual medium attachment, including connectors, for the various supported media.

100BASE-X does not specify the PMD and MDI other than including the appropriate standard by reference along with the minor adaptations necessary for 100BASE-X. Figure 24–2 depicts the relationship between 100BASE-X and the PMDs of ISO/IEC 9314-3: 1990 (for 100BASE-FX) and ANSI X3.263-1995 (for 100BASE-TX). The PMDs (and MDIs) for 100BASE-TX and 100BASE-FX are specified in subsequent clauses of this standard.

### 24.1.5 Inter-sublayer interfaces

There are a number of interfaces employed by 100BASE-X. Some (such as the PMA and PMD interfaces) use an abstract service model to define the operation of the interface. The PCS Interface is defined as a set of physical signals, in a medium-independent manner (MII). Figure 24–3 depicts the relationship and mapping of the services provided by all of the interfaces relevant to 100BASE-X.

It is important to note that, while this specification defines interfaces in terms of bits, nibbles, and code-groups, implementations may choose other data path widths for implementation convenience. The only exceptions are: a) the MII, which, when implemented, uses a nibble-wide data path as specified in Clause 22, and b) the MDI, which uses a serial, physical interface.

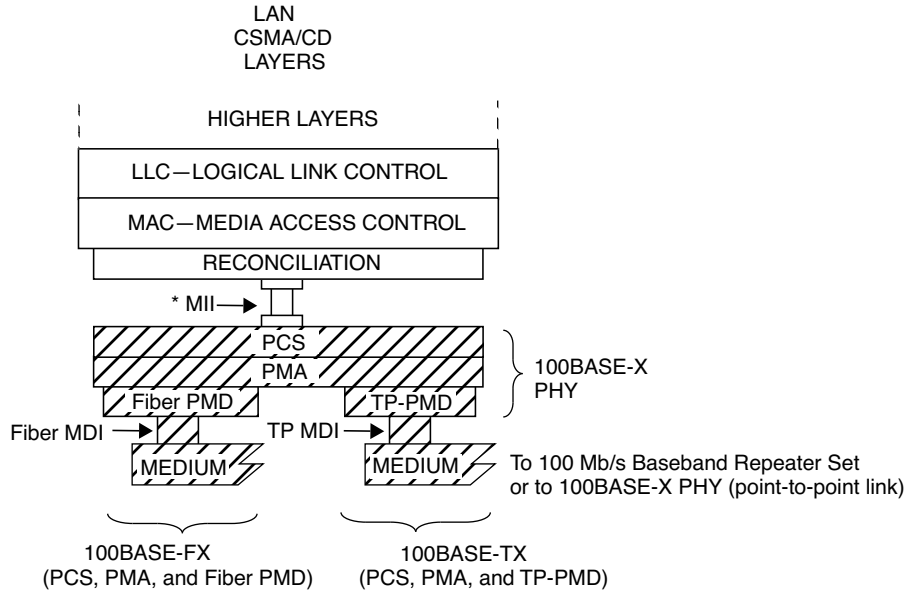
### 24.1.6 Functional block diagram

Figure 24–4 provides a functional block diagram of the 100BASE-X PHY.

### 24.1.7 State diagram conventions

The body of this standard is comprised of state diagrams, including the associated definitions of variables, constants, and functions. Should there be a discrepancy between a state diagram and descriptive text, the state diagram prevails.

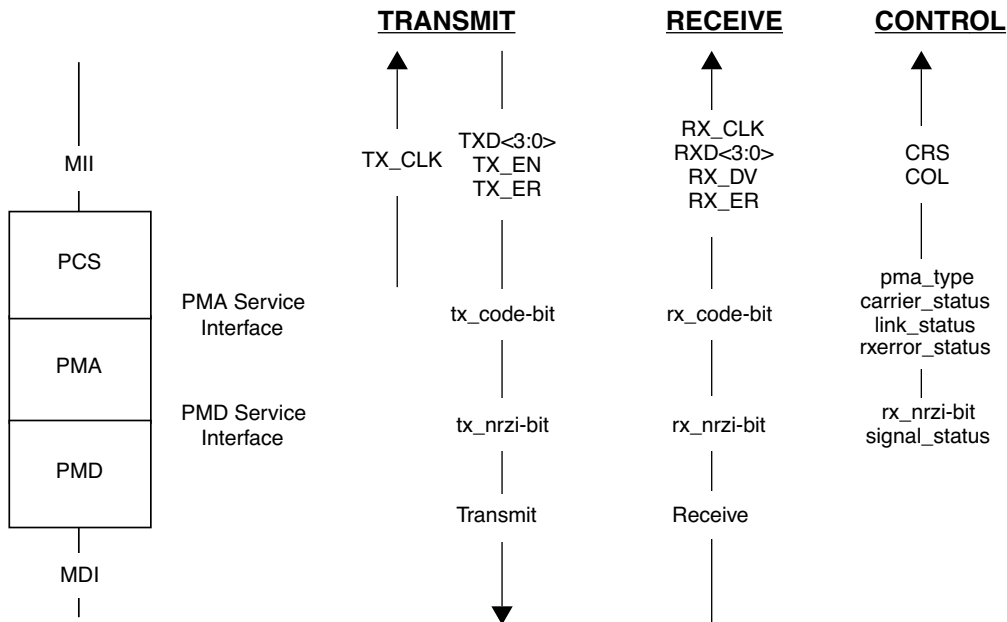
The notation used in the state diagrams follows the conventions of 21.5; state diagram timers follow the conventions of 14.2.3.2.



MDI = MEDIUM DEPENDENT INTERFACE      PMA = PHYSICAL MEDIUM ATTACHMENT  
 MII = MEDIA INDEPENDENT INTERFACE      PHY = PHYSICAL LAYER DEVICE  
 PCS = PHYSICAL CODING SUBLAYER      Fiber PMD = PHYSICAL MEDIUM DEPENDENT SUBLAYER FOR FIBER  
 TP-PMD = PHYSICAL MEDIUM DEPENDENT SUBLAYER FOR TWISTED PAIRS

NOTE—The PMD sublayers are mutually independent.  
 \* MII is optional.

**Figure 24–2—Relationship of 100BASE-X and the PMDs**



**Figure 24–3—Interface mapping**

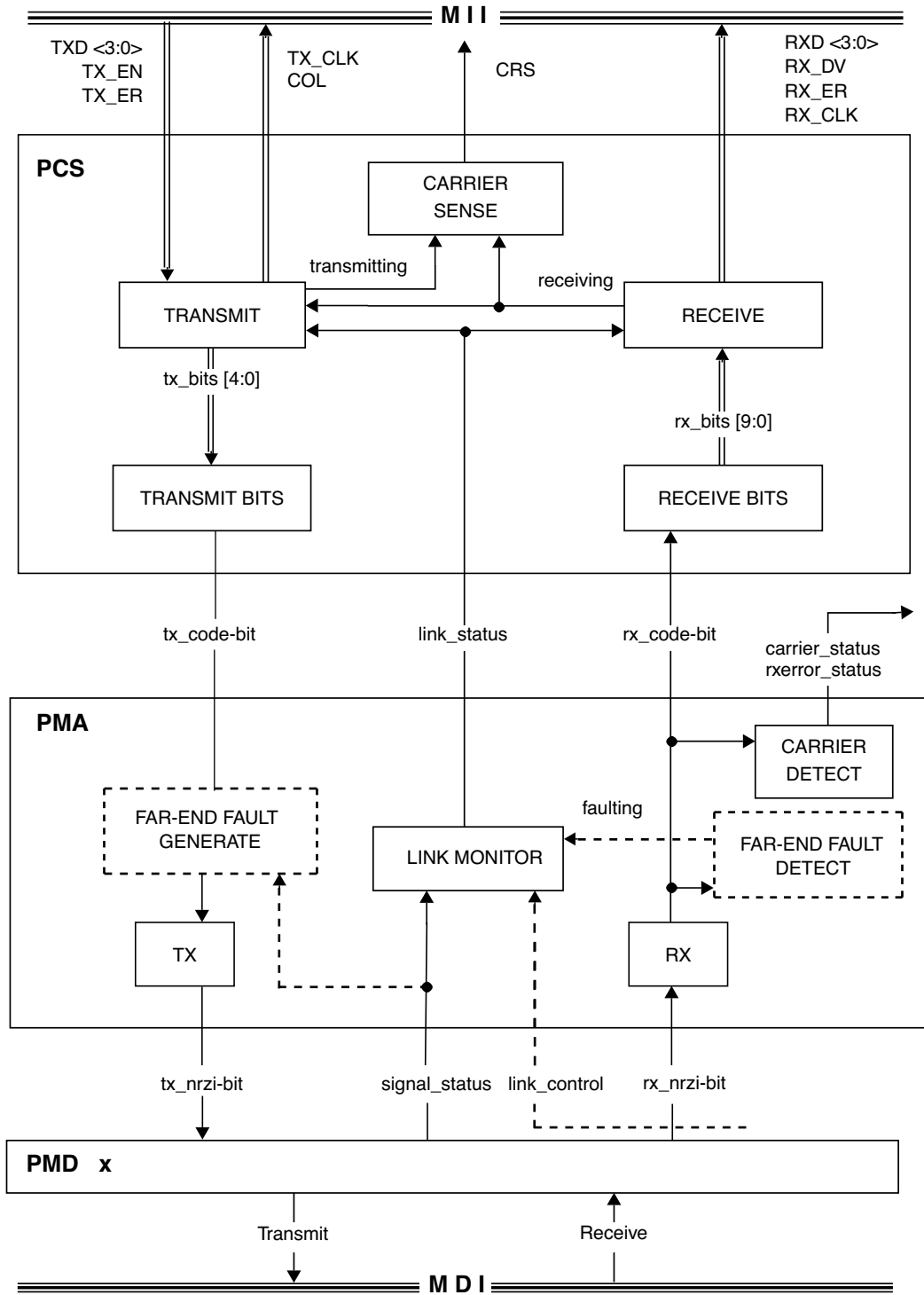


Figure 24-4—Functional block diagram



## 24.2 Physical Coding Sublayer (PCS)

### 24.2.1 Service Interface (MII)

The PCS Service Interface allows the 100BASE-X PCS to transfer information to and from the MAC (via the Reconciliation sublayer) or other PCS client, such as a repeater. The PCS Service Interface is precisely defined as the Media Independent Interface (MII) in Clause 22.

In this clause, the setting of MII variables to TRUE or FALSE is equivalent, respectively, to “asserting” or “de-asserting” them as specified in Clause 22.

### 24.2.2 Functional requirements

The PCS comprises the Transmit, Receive, and Carrier Sense functions for 100BASE-T. In addition, the collisionDetect signal required by the MAC (COL on the MII) is derived from the PMA code-bit stream. The PCS shields the Reconciliation sublayer (and MAC) from the specific nature of the underlying channel. Specifically for receiving, the 100BASE-X PCS passes to the MII a sequence of data nibbles derived from incoming code-groups, each comprised of five code-bits, received from the medium. Code-group alignment and MAC packet delimiting is performed by embedding special non-data code-groups. The MII uses a nibble-wide, synchronous data path, with packet delimiting being provided by separate TX\_EN and RX\_DV signals. The PCS provides the functions necessary to map these two views of the exchanged data. The process is reversed for transmit.

The following provides a detailed specification of the functions performed by the PCS, which comprise five parallel processes (Transmit, Transmit Bits, Receive, Receive Bits, and Carrier Sense). Figure 24–4 includes a functional block diagram of the PCS.

The Receive Bits process accepts continuous code-bits via the PMA\_UNITDATA.indicate primitive. Receive monitors these bits and generates RXD <3:0>, RX\_DV and RX\_ER on the MII, and the internal flag, receiving, used by the Carrier Sense and Transmit processes.

The Transmit process generates continuous code-groups based upon the TXD <3:0>, TX\_EN, and TX\_ER signals on the MII. These code-groups are transmitted by Transmit Bits via the PMA\_UNITDATA.request primitive. The Transmit process generates the MII signal COL based on whether a reception is occurring simultaneously with transmission. Additionally, it generates the internal flag, transmitting, for use by the Carrier Sense process.

The Carrier Sense process asserts the MII signal CRS when either transmitting or receiving is TRUE. Both the Transmit and Receive processes monitor link\_status via the PMA\_LINK.indicate primitive, to account for potential link failure conditions.

#### 24.2.2.1 Code-groups

The PCS maps four-bit nibbles from the MII into five-bit code-groups, and vice versa, using a 4B/5B block coding scheme. A code-group is a consecutive sequence of five code-bits interpreted and mapped by the PCS. Implicit in the definition of a code-group is an establishment of code-group boundaries by an alignment function within the PCS Receive process. It is important to note that, with the sole exception of the SSD, which is used to achieve alignment, code-groups are undetectable and have no meaning outside the 100BASE-X physical protocol data unit, called a “stream.”

The coding method used, derived from ISO/IEC 9314-1, provides

- a) Adequate codes (32) to provide for all Data code-groups (16) plus necessary control code-groups;

- b) Appropriate coding efficiency (4 data bits per 5 code-bits; 80%) to effect a 100 Mb/s Physical Layer interface on a 125 Mb/s physical channel as provided by FDDI PMDs; and
- c) Sufficient transition density to facilitate clock recovery (when not scrambled).

Table 24–1 specifies the interpretation assigned to each five bit code-group, including the mapping to the nibble-wide (TXD or RXD) Data signals on the MII. The 32 code-groups are divided into four categories, as shown.

For clarity in the remainder of this clause, code-group names are shown between /slashes/. Code-group sequences are shown in succession, e.g.: /1/2/....

The indicated code-group mapping is identical to ISO/IEC 9314-1: 1989, with four exceptions:

- a) The FDDI term *symbol* is avoided in order to prevent confusion with other 100BASE-T terminology. In general, the term *code-group* is used in its place.
- b) The /S/ and /Q/ code-groups are not used by 100BASE-X and are interpreted as INVALID.
- c) The /R/ code-group is used in 100BASE-X as the second code-group of the End-of-Stream delimiter rather than to indicate a Reset condition.
- d) The /H/ code-group is used to propagate receive errors rather than to indicate the Halt Line State.

#### 24.2.2.1.1 Data code-groups

A Data code-group conveys one nibble of arbitrary data between the MII and the PCS. The sequence of Data code-groups is arbitrary, where any Data code-group can be followed by any other Data code-group. Data code-groups are coded and decoded but not interpreted by the PCS. Successful decoding of Data code-groups depends on proper receipt of the Start-of-Stream delimiter sequence, as defined in Table 24–1.

#### 24.2.2.1.2 Idle code-groups

The Idle code-group (/I/) is transferred between streams. It provides a continuous fill pattern to establish and maintain clock synchronization. Idle code-groups are emitted from, and interpreted by, the PCS.

#### 24.2.2.1.3 Control code-groups

The Control code-groups are used in pairs (/J/K/, /T/R/) to delimit MAC packets. Control code-groups are emitted from, and interpreted by, the PCS.

#### 24.2.2.1.4 Start-of-Stream Delimiter (/J/K/)

A Start-of-Stream Delimiter (SSD) is used to delineate the boundary of a data transmission sequence and to authenticate carrier events. The SSD is unique in that it may be recognized independently of previously established code-group boundaries. The Receive function within the PCS uses the SSD to establish code-group boundaries. A SSD consists of the sequence /J/K/.

On transmission, the first 8 bits of the MAC preamble are replaced by the SSD, a replacement that is reversed on reception.

#### 24.2.2.1.5 End-of-Stream delimiter (/T/R/)

An End-of-Stream delimiter (ESD) terminates all normal data transmissions. Unlike the SSD, an ESD cannot be recognized independent of previously established code-group boundaries. An ESD consists of the sequence /T/R/.

Table 24-1—4B/5B code-groups

	PCS code-group [4:0] 4 3 2 1 0	Name	MII (TXD/RXD) <3:0> 3 2 1 0	Interpretation
D A T A	1 1 1 1 0	0	0 0 0 0	Data 0
	0 1 0 0 1	1	0 0 0 1	Data 1
	1 0 1 0 0	2	0 0 1 0	Data 2
	1 0 1 0 1	3	0 0 1 1	Data 3
	0 1 0 1 0	4	0 1 0 0	Data 4
	0 1 0 1 1	5	0 1 0 1	Data 5
	0 1 1 1 0	6	0 1 1 0	Data 6
	0 1 1 1 1	7	0 1 1 1	Data 7
	1 0 0 1 0	8	1 0 0 0	Data 8
	1 0 0 1 1	9	1 0 0 1	Data 9
	1 0 1 1 0	A	1 0 1 0	Data A
	1 0 1 1 1	B	1 0 1 1	Data B
	1 1 0 1 0	C	1 1 0 0	Data C
	1 1 0 1 1	D	1 1 0 1	Data D
	1 1 1 0 0	E	1 1 1 0	Data E
	1 1 1 0 1	F	1 1 1 1	Data F
	1 1 1 1 1	I	undefined	IDLE; used as inter-stream fill code
C O N T R O L	1 1 0 0 0	J	0 1 0 1	Start-of-Stream Delimiter, Part 1 of 2; always used in pairs with K
	1 0 0 0 1	K	0 1 0 1	Start-of-Stream Delimiter, Part 2 of 2; always used in pairs with J
	0 1 1 0 1	T	undefined	End-of-Stream Delimiter, Part 1 of 2; always used in pairs with R
	0 0 1 1 1	R	undefined	End-of-Stream Delimiter, Part 2 of 2; always used in pairs with T
I N V A L I D	0 0 1 0 0	H	Undefined	Transmit Error; used to force signaling errors
	0 0 0 0 0	V	Undefined	Invalid code
	0 0 0 0 1	V	Undefined	Invalid code
	0 0 0 1 0	V	Undefined	Invalid code
	0 0 0 1 1	V	Undefined	Invalid code
	0 0 1 0 1	V	Undefined	Invalid code
	0 0 1 1 0	V	Undefined	Invalid code
	0 1 0 0 0	V	Undefined	Invalid code
	0 1 1 0 0	V	Undefined	Invalid code
	1 0 0 0 0	V	Undefined	Invalid code
1 1 0 0 1	V	Undefined	Invalid code	

**24.2.2.1.6 Invalid code-groups**

The /H/ code-group indicates that the PCS’s client wishes to indicate a Transmit Error to its peer entity. The normal use of this indicator is for repeaters to propagate received errors. Transmit Error code-groups are emitted from the PCS, at the request of the PCS’s client through the use of the TX\_ER signal, as described in 24.2.4.2.

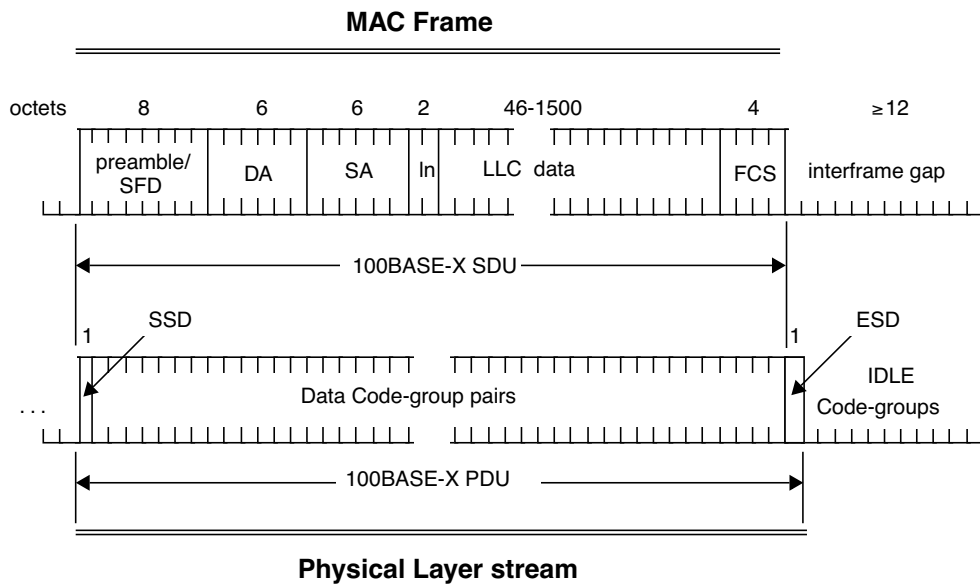
The presence of any invalid code-group on the medium, including /H/, denotes a collision artifact or an error condition. Invalid code-groups are not intentionally transmitted onto the medium by DTE's. The PCS indicates the reception of an Invalid code-group on the MII through the use of the RX\_ER signal, as described in 24.2.4.4.

**24.2.2.2 Encapsulation**

The 100BASE-X PCS accepts frames from the MAC through the Reconciliation sublayer and MII. Due to the continuously signaled nature of the underlying PMA, and the encoding performed by the PCS, the 100BASE-X PCS encapsulates the MAC frame (100BASE-X Service Data Unit, SDU) into a Physical Layer stream (100BASE-X Protocol Data Unit, PDU).

Except for the two code-group SSD, data nibbles within the SDU (including the non-SSD portions of the MAC preamble and SFD) are not interpreted by the 100BASE-X PHY. The conversion from a MAC frame to a Physical Layer stream and back to a MAC frame is transparent to the MAC.

Figure 24–5 depicts the mapping between MAC frames and Physical Layer streams.



**Figure 24–5—PCS encapsulation**

A properly formed stream can be viewed as comprising three elements:

- a) *Start-of-Stream Delimiter*. The start of a Physical Layer stream is indicated by a SSD, as defined in 24.2.2.1. The SSD replaces the first octet of the preamble from the MAC frame and vice versa.
- b) *Data Code-groups*. Between delimiters (SSD and ESD), the PCS conveys Data code-groups corresponding to the data nibbles of the MII. These Data code-groups comprise the 100BASE-X Service Data Unit (SDU). Data nibbles within the SDU (including those corresponding to the MAC preamble and SFD) are not interpreted by the 100BASE-X PCS.
- c) *End-of-Stream Delimiter*. The end of a properly formed stream is indicated by an ESD, as defined in 24.2.2.1. The ESD is transmitted by the PCS following the de-assertion of TX\_EN on the MII, which corresponds to the last data nibble composing the FCS from the MAC. It is transmitted during the period considered by the MAC to be the interframe gap (IFG). On reception, ESD is interpreted by the PCS as terminating the SDU.

Between streams, IDLE code-groups are conveyed between the PCS and PMA.

### 24.2.2.3 Data delay

The PCS maps a non-aligned code-bit data path from the PMA to an aligned, nibble-wide data path on the MII, both for Transmit and Receive. Logically, received bits must be buffered to facilitate SSD detection and alignment, coding translation, and ESD detection. These functions necessitate an internal PCS delay of at least two code-groups. In practice, alignment may necessitate even longer delays of the incoming code-bit stream.

When the MII is present as an exposed interface, the MII signals TX\_CLK and RX\_CLK, not depicted in the following state diagrams, shall be generated by the PCS in accordance with Clause 22.

### 24.2.2.4 Mapping between MII and PMA

Figure 24–6 depicts the mapping of the nibble-wide data path of the MII to the five-bit-wide code-groups (internal to the PCS) and the code-bit path of the PMA interface.

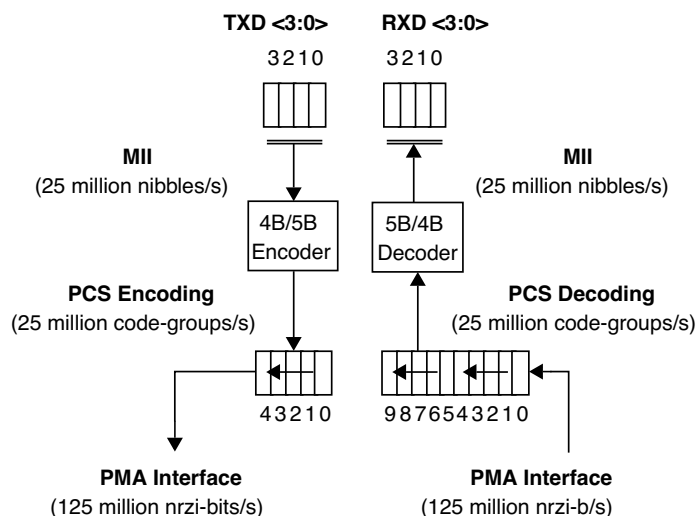


Figure 24–6—PCS reference diagram

Upon receipt of a nibble from the MII, the PCS encodes it into a five-bit code-group, according to 24.2.2.1. Code-groups are serialized into code-bits and passed to the PMA for transmission on the underlying medium, according to Figure 24–6. The first transmitted code-bit of a code-group is bit 4, and the last code-bit transmitted is bit 0. There is no numerical significance ascribed to the bits within a code-group; that is, the code-group is simply a five-bit pattern that has some predefined interpretation.

Similarly, the PCS deserializes code-bits received from the PMA, according to Figure 24–6. After alignment is achieved, based on SSD detection, the PCS converts code-groups into MII data nibbles, according to 24.2.2.1.

### 24.2.3 State variables

#### 24.2.3.1 Constants

DATA	The set of 16 code-groups corresponding to valid DATA, as specified in 24.2.2.1. (In the Receive state diagram, the set operators $\in$ and $\notin$ are used to represent set membership and non-membership, respectively.)
ESD	The code-group pair corresponding to the End-of-Stream delimiter, as specified in 24.2.2.1.
ESD1	The code-group pair corresponding to the End-of-Stream delimiter, Part 1 (/T/), as specified in 24.2.2.1.
ESD2	The code-group pair corresponding to the End-of-Stream delimiter, Part 2 (/R/), as specified in 24.2.2.1.
HALT	The Transmit Error code-group (/H/), as specified in 24.2.2.1.
IDLE	The IDLE code-group, as specified in 24.2.2.1.
IDLES	A code-group pair comprised of /II/; /I/ as specified in 24.2.2.1.
SSD	The code-group pair corresponding to the Start-of-Stream delimiter, as specified in 24.2.2.1.
SSD1	The code-group corresponding to the Start-of-Stream delimiter, Part 1 (/J/), as specified in 24.2.2.1.
SSD2	The code-group corresponding to the Start-of-Stream delimiter, Part 2 (/K/), as specified in 24.2.2.1.

#### 24.2.3.2 Variables

In the following, values for the MII parameters are definitively specified in Clause 22.

COL	The COL signal of the MII as specified in Clause 22.
CRS	The CRS signal of the MII as specified in Clause 22.

**link\_status**

The link\_status parameter as communicated by the PMA\_LINK.indicate primitive.

Values: FAIL; the receive channel is not intact  
 READY; the receive channel is intact and ready to be enabled by Auto-Negotiation  
 OK; the receive channel is intact and enabled for reception

**receiving**

A boolean set by the Receive process to indicate non-IDLE activity (after squelch). Used by the Carrier Sense process, and also interpreted by the Transmit process for indicating a collision.

Values: TRUE; unsquelched carrier being received  
 FALSE; carrier not being received

**rx\_bits [9:0]**

A vector of the 10 most recently received code-bits from the PMA as assembled by Receive Bits and processed by Receive. rx\_bits [0] is the most recently received (newest) code-bit; rx\_bits [9] is the least recently received code-bit (oldest). When alignment has been achieved, it contains the last two code-groups.

**rx\_code-bit**

The rx\_code-bit parameter as communicated by the most recent PMA\_UNITDATA.indicate primitive (that is, the value of the most recently received code-bit from the PMA).

**RX\_DV**

The RX\_DV signal of the MII as specified in Clause 22. Set by the Receive process, RX\_DV is also interpreted by the Receive Bits process as an indication that rx\_bits is code-group aligned.

**RX\_ER**

The RX\_ER signal of the MII as specified in Clause 22.

**RXD <3:0>**

The RXD <3:0> signal of the MII as specified in Clause 22.

**transmitting**

A boolean set by the Transmit Process to indicate a transmission in progress. Used by the Carrier Sense process.

Values: TRUE; the PCS's client is transmitting  
 FALSE; the PCS's client is not transmitting

**tx\_bits [4:0]**

A vector of code-bits representing a code-group prepared for transmission by the Transmit Process and transmitted to the PMA by the Transmit Bits process.

**TX\_EN**

The TX\_EN signal of the MII as specified in Clause 22.

**TX\_ER**

The TX\_ER signal of the MII as specified in Clause 22.

**TXD <3:0>**

The TXD <3:0> signal of the MII as specified in Clause 22.

**24.2.3.3 Functions****nibble DECODE (code-group)**

In Receive, this function takes as its argument a five-bit code-group and returns the corresponding MII RXD <3:0> nibble, per Table 24–1.

**code-group ENCODE (nibble)**

In the Transmit process, this function takes as its argument an MII TXD <3:0> nibble, and returns the corresponding five-bit code-group per Table 24–1.

**SHIFTLEFT (rx\_bits)**

In Receive Bits, this function shifts rx\_bits left one bit placing rx\_bits [8] in rx\_bits [9], rx\_bits [7] in rx\_bits [8] and so on until rx\_bits [1] gets rx\_bits [0].

### 24.2.3.4 Timers

#### code-bit\_timer

In the Transmit Bits process, the timer governing the output of code-bits from the PCS to the PMA and thereby to the medium with a nominal 8 ns period. This timer shall be derived from a fixed frequency oscillator with a base frequency of 125 MHz  $\pm$  0.005% and phase jitter above 20 kHz less than  $\pm$  8°.

### 24.2.3.5 Messages

#### gotCodeGroup.indicate

A signal sent to the Receive process by the Receive Bits process after alignment has been achieved signifying completion of reception of the next code-group in rx\_bits(4:0), with the preceding code-group moved to rx\_bits [9:5]. rx\_bits [9:5] may be considered as the “current” code-group.

#### PMA\_UNITDATA.indicate (rx\_code-bit)

A signal sent by the PMA signifying that the next code-bit from the medium is available in rx\_code-bit.

#### sentCodeGroup.indicate

A signal sent to the Transmit process from the Transmit Bits process signifying the completion of transmission of the code-group in tx\_bits [4:0].

## 24.2.4 State diagrams

### 24.2.4.1 Transmit Bits

Transmit Bits is responsible for taking code-groups prepared by the Transmit process and transmitting them to the PMA using PMA\_UNITDATA.request, the frequency of which determines the transmit clock. Transmit deposits these code-groups in tx\_bits with Transmit Bits signaling completion of a code-group transmission with sentCodeGroup.indicate.

The PCS shall implement the Transmit Bits process as depicted in Figure 24–7 including compliance with the associated state variables as specified in 24.2.3.

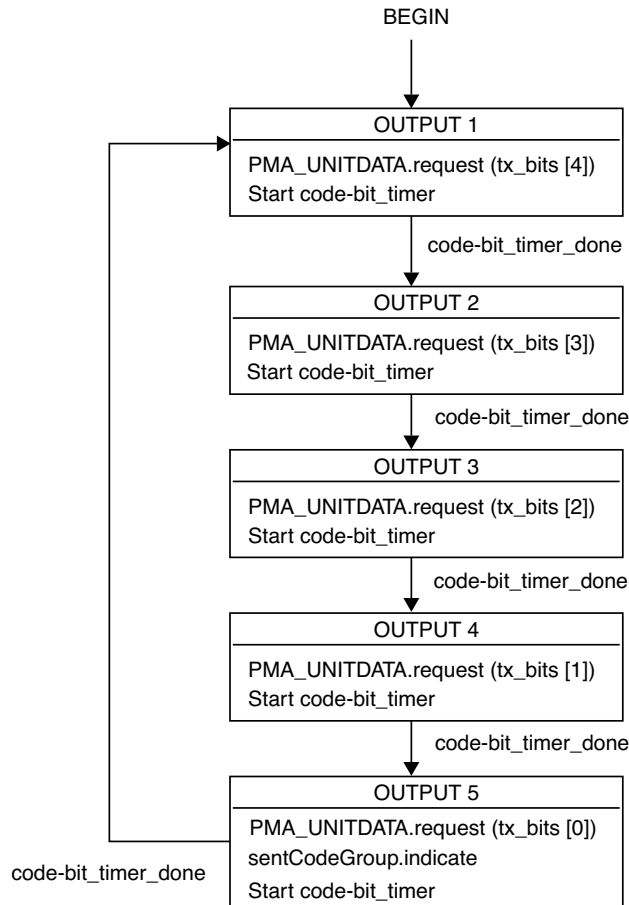
### 24.2.4.2 Transmit

The Transmit process sends code-groups to the PMA via tx\_bits and the Transmit Bits process. When initially invoked, and between streams (delimited by TX\_EN on the MII), the Transmit process sources continuous Idle code-groups (/I/) to the PMA. Upon the assertion of TX\_EN by the MII, the Transmit process passes an SSD (/J/K/) to the PMA, ignoring the TXD <3:0> nibbles during these two code-group times. Following the SSD, each TXD <3:0> nibble is encoded into a five-bit code-group until TX\_EN is deasserted. If, while TX\_EN is asserted, the TX\_ER signal is asserted, the Transmit process passes Transmit Error code-groups (/H/) to the PMA. Following the de-assertion of TX\_EN, an ESD (/T/R/) is generated, after which the transmission of Idle code-groups is resumed by the IDLE state.

Collision detection is implemented by noting the occurrence of carrier receptions during transmissions, following the model of 10BASE-T. The indication of link\_status  $\neq$  OK by the PMA at any time causes an immediate transition to the IDLE state and supersedes any other Transmit process operations.

The PCS shall implement the Transmit process as depicted in Figure 24–8 including compliance with the associated state variables as specified in 24.2.3.





**Figure 24-7—Transmit Bits state diagram**

### 24.2.4.3 Receive Bits

The Receive Bits process collects code-bits from the PMA interface passing them to the Receive process via `rx_bits`. `rx_bits [9:0]` represents a sliding, 10-bit window on the PMA code-bits, with newly received code-bits from the PMA (`rx_code-bit`) being shifted into `rx_bits [0]`. This is depicted in Figure 24-9. Bits are collected serially until Receive indicates alignment by asserting `RX_DV`, after which Receive signals Receive for every five code-bits accumulated. Serial processing resumes with the de-assertion of `RX_DV`.

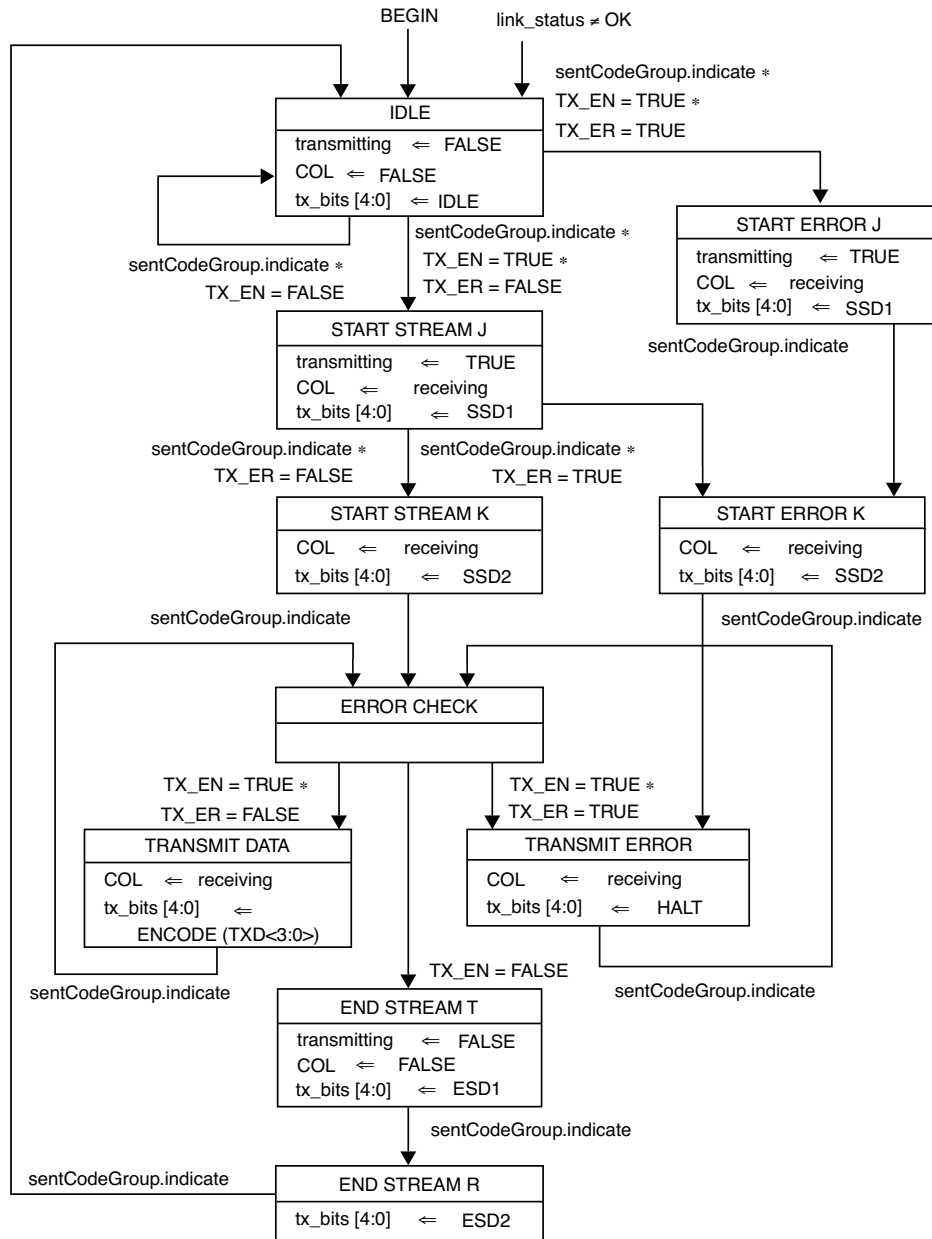
The PCS shall implement the Receive Bits process as depicted in Figure 24-10 including compliance with the associated state variables as specified in 24.2.3.

### 24.2.4.4 Receive

The Receive process state machine can be viewed as comprising two sections: prealigned and aligned. In the prealigned states, `IDLE`, `CARRIER DETECT`, and `CONFIRM K`, the Receive process is waiting for an indication of channel activity followed by a `SSD`. After successful alignment, the incoming code-groups are decoded while waiting for stream termination.

#### 24.2.4.4.1 Detecting channel activity

In a DTE operating in half duplex mode, the detection of activity on the underlying channel is used both by the MAC (via the `MII CRS` signal and the Reconciliation sublayer) for deferral purposes, and by the

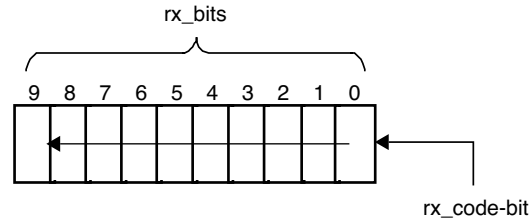


**Figure 24-8—Transmit state diagram**

Transmit process for collision detection. Activity, signaled by the assertion of receiving, is indicated by the receipt of two non-contiguous ZEROS within any 10 code-bits of the incoming code-bit stream.

#### 24.2.4.4.2 Code-group alignment

After channel activity is detected, the Receive process first aligns the incoming code-bits on code-group boundaries for subsequent data decoding. This is achieved by scanning the rx\_bits vector for a SSD (*J/K*). The MII RX\_DV signal remains deasserted during this time, which ensures that the Reconciliation sublayer will ignore any signals on RXD <3:0>. Detection of the SSD causes the Receive process to enter the START OF STREAM J state.



**Figure 24-9—Receive Bits reference diagram**

Well-formed streams contain SSD (/J/K/) in place of the first eight preamble bits. In the event that something else is sensed immediately following detection of carrier, a False Carrier Indication is signaled to the MII by asserting RX\_ER and setting RXD to 1110 while RX\_DV remains de-asserted. The associated carrier event, as terminated by 10 ONES, is otherwise ignored.

#### 24.2.4.4.3 Stream decoding

The Receive process substitutes a sequence of alternating ONE and ZERO data-bits for the SSD, which is consistent with the preamble pattern expected by the MAC.

The Receive process then performs the DECODE function on the incoming code-groups, passing decoded data to the MII, including those corresponding to the remainder of the MAC preamble and SFD. The MII signal RX\_ER is asserted upon decoding any code-group following the SSD that is neither a valid Data code-group nor a valid stream termination sequence.

#### 24.2.4.4.4 Stream termination

There are two means of effecting stream termination in the Receive process (Figure 24-11).

A normal stream termination is caused by detection of an ESD (/T/R/) in the rx\_bits vector. In order to preserve the ability of the MAC to properly delimit the FCS at the end of the frame (that is, to avoid incorrect AlignmentErrors in the MAC) the internal signal receiving (and through it, the MII CRS signal, per Clause 22) is de-asserted immediately following the last code-bit in the stream that maps to the FCS. Note that the condition link\_status  $\neq$  OK during stream reception (that is, when receiving = TRUE) causes an immediate transition to the LINK FAILED state and supersedes any other Receive process operations.

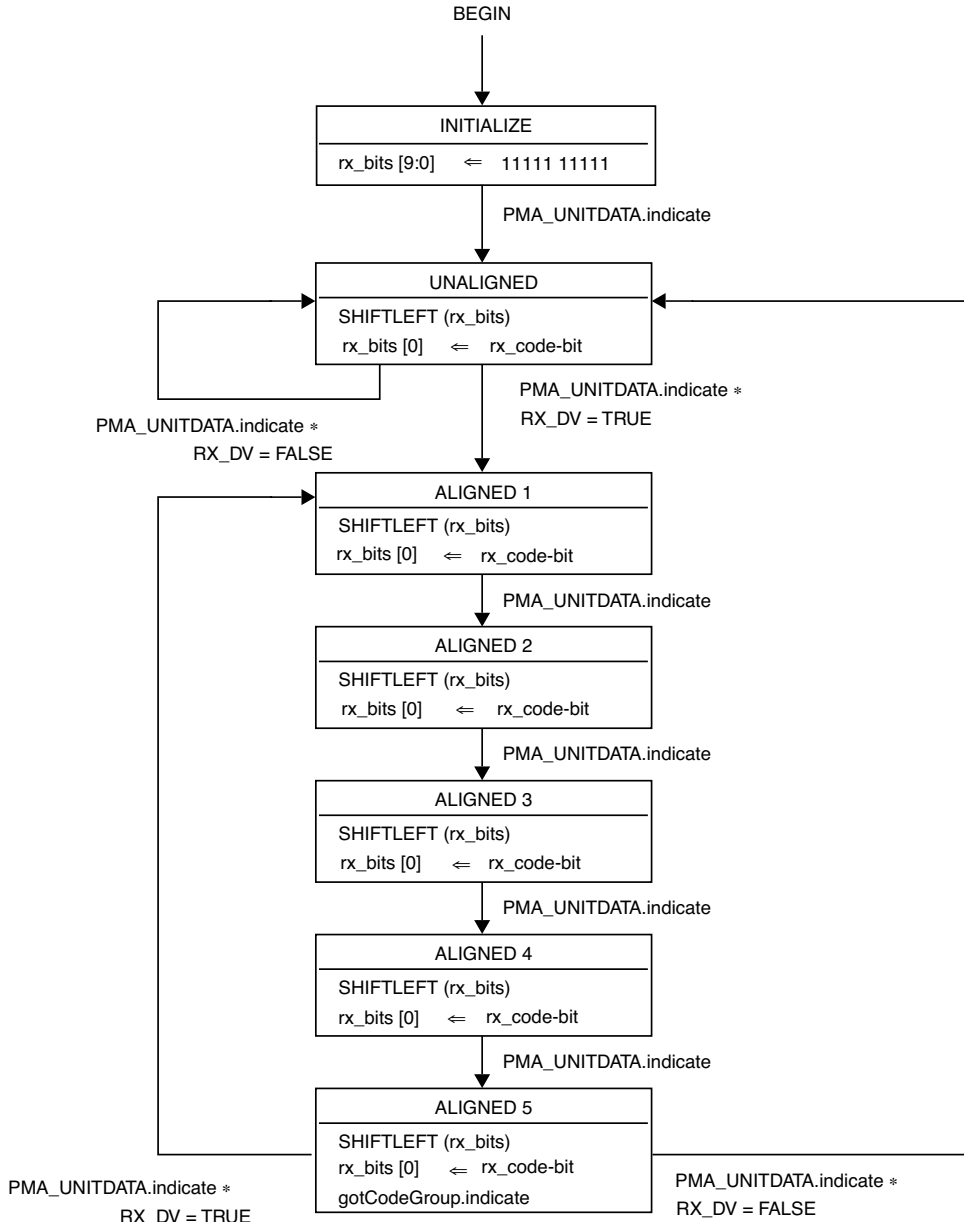
A premature stream termination is caused by the detection of two Idle code-groups (/I/I) in the rx\_bits vector prior to an ESD. Note that RX\_DV remains asserted during the nibble corresponding to the first five contiguous ONES while RX\_ER is signaled on the MII. RX\_ER is also asserted in the LINK FAILED state, which ensures that RX\_ER remains asserted for sufficient time to be detected.

Stream termination causes a transition to the IDLE state.

The PCS shall implement the Receive process as depicted in Figure 24-11 including compliance with the associated state variables as specified in 24.2.3.

#### 24.2.4.5 Carrier Sense

The Carrier Sense process generates the signal CRS on the MII, which (via the Reconciliation sublayer) a MAC operating in half duplex mode uses for deferral. The process operates by performing a logical OR operation on the internal messages receiving and transmitting, generated by the Receive and Transmit processes, respectively.



**Figure 24–10—Receive Bits state diagram**

The PCS shall implement the Carrier Sense process as depicted in Figure 24–12 including compliance with the associated state variables as specified in 24.2.3.

### 24.3 Physical Medium Attachment (PMA) sublayer

#### 24.3.1 Service interface

The following specifies the service interface provided by the PMA to the PCS or another client, such as a repeater. These services are described in an abstract manner and do not imply any particular implementation.

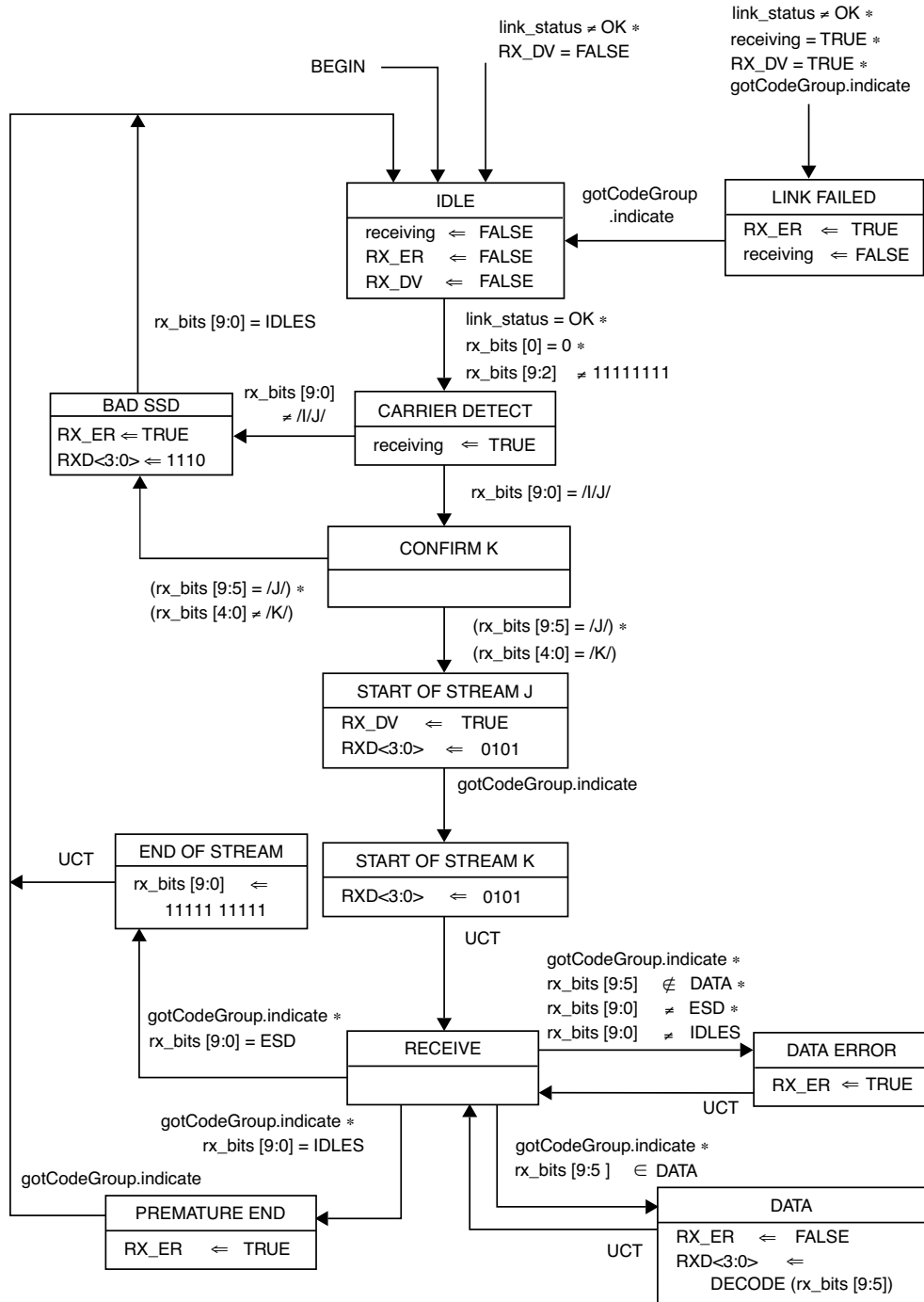
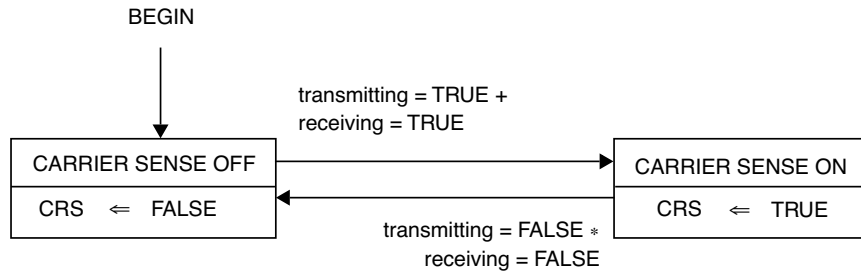


Figure 24-11 – Receive state diagram



**Figure 24–12—Carrier Sense state diagram**

The PMA Service Interface supports the exchange of code-bits between the PCS and/or Repeater entities. The PMA converts code-bits into NRZI format and passes these to the PMD, and vice versa. It also generates additional status indications for use by its client.

The following primitives are defined:

- PMA\_TYPE.indicate
- PMA\_UNITDATA.request
- PMA\_UNITDATA.indicate
- PMA\_CARRIER.indicate
- PMA\_LINK.indicate
- PMA\_LINK.request
- PMA\_RXERROR.indicate

#### 24.3.1.1 PMA\_TYPE.indicate

This primitive is generated by the PMA to indicate the nature of the PMA instantiation. The purpose of this primitive is to allow clients to support connections to the various types of 100BASE-T PMA entities in a generalized manner.

##### 24.3.1.1.1 Semantics of the service primitive

PMA\_TYPE.indicate (pma\_type)

The pma\_type parameter for use with a 100BASE-X PMA is “X”.

##### 24.3.1.1.2 When generated

The PMA continuously generates this primitive to indicate the value of pma\_type.

##### 24.3.1.1.3 Effect of receipt

The effect of receipt of this primitive by the client is unspecified by the PMA sublayer.

#### 24.3.1.2 PMA\_UNITDATA.request

This primitive defines the transfer of data (in the form of code-bits) from the PMA’s client to the PMA.

#### **24.3.1.2.1 Semantics of the service primitive**

PMA\_UNITDATA.request (tx\_code-bit)

This primitive defines the transfer of data (in the form of code-bits) from the PCS or other client to the PMA. The tx\_code-bit parameter can take one of two values: ONE or ZERO.

#### **24.3.1.2.2 When generated**

The PCS or other client continuously sends, at a nominal 125 Mb/s rate, the appropriate code-bit for transmission on the medium.

#### **24.3.1.2.3 Effect of receipt**

Upon receipt of this primitive, the PMA generates a PMD\_UNITDATA.request primitive, requesting transmission of the indicated code-bit, in NRZI format (tx\_nrzi-bit), on the MDI.

#### **24.3.1.3 PMA\_UNITDATA.indicate**

This primitive defines the transfer of data (in the form of code-bits) from the PMA to the PCS or other client.

#### **24.3.1.3.1 Semantics of the service primitive**

PMA\_UNITDATA.indicate (rx\_code-bit)

The data conveyed by PMA\_UNITDATA.indicate is a continuous code-bit sequence at a nominal 125 Mb/s rate. The rx\_code-bit parameter can take one of two values: ONE or ZERO.

#### **24.3.1.3.2 When generated**

The PMA continuously sends code-bits to the PCS or other client corresponding to the PMD\_UNITDATA.indicate primitives received from the PMD.

#### **24.3.1.3.3 Effect of receipt**

The effect of receipt of this primitive by the client is unspecified by the PMA sublayer.

#### **24.3.1.4 PMA\_CARRIER.indicate**

This primitive is generated by the PMA to indicate that a non-squelched, non-IDLE code-bit sequence is being received from the PMD. The purpose of this primitive is to give clients the earliest reliable indication of activity on the underlying continuous-signaling channel.

#### **24.3.1.4.1 Semantics of the service primitive**

PMA\_CARRIER.indicate (carrier\_status)

The carrier\_status parameter can take on one of two values, ON or OFF, indicating whether a non-squelched, non-IDLE code-bit sequence (that is, carrier) is being received (ON) or not (OFF).

#### **24.3.1.4.2 When generated**

The PMA generates this primitive to indicate a change in the value of carrier\_status.

#### **24.3.1.4.3 Effect of receipt**

The effect of receipt of this primitive by the client is unspecified by the PMA sublayer.

#### **24.3.1.5 PMA\_LINK.indicate**

This primitive is generated by the PMA to indicate the status of the underlying PMD receive link.

##### **24.3.1.5.1 Semantics of the service primitive**

PMA\_LINK.indicate (link\_status)

The link\_status parameter can take on one of three values: READY, OK, or FAIL, indicating whether the underlying receive channel is intact and ready to be enabled by Auto-Negotiation (READY), intact and enabled (OK), or not intact (FAIL). Link\_status is set to FAIL when the PMD sets signal\_status to OFF; when Auto-Negotiation (optional) sets link\_control to DISABLE; or when Far-End Fault Detect (optional) sets faulting to TRUE. When link\_status ≠ OK, then rx\_code-bit and carrier\_status are undefined.

##### **24.3.1.5.2 When generated**

The PMA generates this primitive to indicate a change in the value of link\_status.

##### **24.3.1.5.3 Effect of receipt**

The effect of receipt of this primitive by the client is unspecified by the PMA sublayer.

#### **24.3.1.6 PMA\_LINK.request**

This primitive is generated by the Auto-Negotiation algorithm, when implemented, to allow it to enable and disable operation of the PMA. See Clause 28. When Auto-Negotiation is not implemented, the primitive is never invoked and the PMA behaves as if link\_control = ENABLE.

##### **24.3.1.6.1 Semantics of the service primitive**

PMA\_LINK.request (link\_control)

The link\_control parameter takes on one of three values: SCAN\_FOR\_CARRIER, DISABLE, or ENABLE. Auto-Negotiation sets link\_control to SCAN\_FOR\_CARRIER prior to receiving any fast link pulses, permitting the PMA to sense a 100BASE-X signal. Auto-Negotiation sets link\_control to DISABLE when it senses an Auto-Negotiation partner (fast link pulses) and must temporarily disable the 100BASE-X PHY while negotiation ensues. Auto-Negotiation sets link\_control to ENABLE when full control is passed to the 100BASE-X PHY.

##### **24.3.1.6.2 When generated**

Auto-Negotiation generates this primitive to indicate a change in link\_control as described in Clause 28.

##### **24.3.1.6.3 Effect of receipt**

This primitive affects operation of the PMA Link Monitor function as described in 24.3.4.4.

#### **24.3.1.7 PMA\_RXERROR.indicate**

This primitive is generated by the PMA to indicate that an error has been detected during a carrier event.



#### 24.3.1.7.1 Semantics of the service primitive

PMA\_RXERROR.indicate (rxerror\_status)

The rxerror\_status parameter can take on one of two values: ERROR or NO\_ERROR, indicating whether the received carrier event contains a detectable error (ERROR) or not (NO\_ERROR). A carrier event is considered to be in error when it is not started by a Start-of-Stream Delimiter.

#### 24.3.1.7.2 When generated

The PMA generates this primitive whenever a new, non-squelched carrier event is not started by a Start-of-Stream Delimiter.

#### 24.3.1.7.3 Effect of receipt

The effect of receipt of this primitive by the client is unspecified by the PMA sublayer.

### 24.3.2 Functional requirements

The 100BASE-X PMA comprises the following functions:

- a) Mapping of transmit and receive code-bits between the PMA Service Interface and the PMD Service Interface;
- b) Link Monitor, which maps the PMD\_SIGNAL.indicate primitive to the PMA\_LINK.indicate primitive, indicating the availability of the underlying PMD;
- c) Carrier Detection, which generates the PMA\_CARRIER.indicate and PMA\_RXERROR.indicate primitives from inspection of received PMD signals; and
- d) Far-End Fault (optional), comprised of the Far-End Fault Generate and Far-End Fault Detect processes, which sense receive channel failures and send the Far-End Fault Indication, and sense the Far-End Fault Indication.

Figure 24–4 includes a functional block diagram of the PMA.

#### 24.3.2.1 Far-End fault

Auto-Negotiation provides a Remote Fault capability useful for detection of asymmetric link failures; i.e., channel error conditions detected by the far-end station but not the near-end station. Since Auto-Negotiation is specified only for media supporting eight-pin modular connectors, such as used by 100BASE-TX over unshielded twisted pair, Auto-Negotiation's Remote Fault capability is unavailable to other media for which it may be functionally beneficial, such as 100BASE-TX over shielded twisted pair or 100BASE-FX. A remote fault capability for 100BASE-FX is particularly useful due to this medium's applicability over longer distances (making end-station checking inconvenient) and for backbones (in which detection of link failures can trigger redundant systems).

For these reasons, 100BASE-X provides an optional Far-End Fault facility when Auto-Negotiation cannot be used. Far-End Fault shall not be implemented for media capable of supporting Auto-Negotiation.

When no signal is being received, as indicated by the PMD's signal detect function, the Far-End Fault feature permits the station to transmit a special Far-End Fault Indication to its far-end peer. The Far-End Fault Indication is sent only when a physical error condition is sensed on the receive channel. In all other situations, including reception of the Far-End Fault Indication itself, the PMA passes through tx\_code-bit. (Note that the Far-End Fault architecture is such that IDLEs are automatically transmitted when the Far-End Fault Indication is detected. This is necessary to re-establish communication when the link is repaired.)

The Far-End Fault Indication is comprised of three or more repeating cycles, each of 84 ONEs followed by a single ZERO. This signal is sent in-band and is readily detectable but is constructed so as to not satisfy the 100BASE-X carrier sense criterion. It is therefore transparent to the PMA's client and to stations not implementing Far-End Fault.

As shown in Figure 24–4, Far-End Fault is implemented through the Far-End Fault Generate, Far-End Fault Detect, and the Link Monitor processes.

The Far-End Fault Generate process, which is interposed between the incoming tx\_code-bit stream and the TX process, is responsible for sensing a receive channel failure (signal\_status=OFF) and transmitting the Far-End Fault Indication in response. The transmission of the Far-End Fault Indication may start or stop at any time depending only on signal\_status.

The Far-End Fault Detect process continuously monitors rx\_code-bits from the RX process for the Far-End Fault Indication. Detection of the Far-End Fault Indication disables the station by causing the Link Monitor process to deassert link\_status, which in turn causes the station to source IDLEs. Far-End Fault detection can also be used by management functions not specified in this clause.

#### **24.3.2.2 Comparison to previous 802.3<sup>®</sup> PMAs**

Previous 802.3<sup>®</sup> PMAs perform the additional functions of SQE Test and Jabber. Neither of these functions is implemented in the 100BASE-X PMA.

SQE Test is provided in other Physical Layers to check the integrity of the Collision Detection mechanism independently of the Transmit and Receive capabilities of the Physical Layer. Since 100BASE-X effects collision detection by sensing receptions that occur during transmissions, collision detection is dependent on the health of the receive channel. By checking the ability to properly receive signals from the PMD, the Link Monitor function therefore functionally subsumes the functions previously implemented by SQE Test.

The Jabber function prevents a DTE from causing total network failure under certain classes of faults. When using mixing media (e.g., coaxial cables or passive optical star couplers), this function must naturally be implemented in the DTE. 100BASE-X requires the use of an active repeater, with one DTE or repeater attached to each port. As an implementation optimization, the Jabber function has therefore been moved to the repeater in 100BASE-X.

### **24.3.3 State variables**

#### **24.3.3.1 Constants**

##### **FEF\_CYCLES**

The number of consecutive cycles (of FEF\_ONES ONEs and a single ZERO) necessary to indicate the Far-End Fault Indication. This value is 3.

##### **FEF\_ONES**

The number of consecutive ONEs to be transmitted for each cycle of the Far-End Fault Indication. This value is 84.

### 24.3.3.2 Variables

#### carrier\_status

The carrier\_status parameter to be communicated by the Carrier Detect process through the PMA\_CARRIER.indicate primitive. Carrier is defined as receipt of 2 noncontiguous ZEROes in 10 code-bits.

Values: ON; carrier is being received  
OFF; carrier is not being received

#### faulting

The faulting variable set by the Far-End Fault Detect process, when implemented, indicating whether or not a Far-End Fault Indication is being sensed. This variable is used by the Link Monitor process to force link\_status to FAIL. When Far-End Fault is not implemented, this variable is always FALSE.

Values: TRUE; Far-End Fault Indication is being sensed  
FALSE; Far-End Fault Indication is not being sensed

#### link\_control

The link\_control parameter as communicated by the PMA\_LINK.request primitive. When Auto-Negotiation is not implemented, the value of link\_control is always ENABLE. See Clause 28 for a complete definition.

#### link\_status

The link\_status parameter as communicated by the Link Monitor process through the PMA\_LINK.indicate primitive.

Values: FAIL; the receive channel is not intact  
READY; the receive channel is intact and ready to be enabled by Auto-Negotiation  
OK; the receive channel is intact and enabled for reception

#### r\_bits [9:0]

In Carrier Detect, a vector of the 10 most recently received code-bits from the PMD RX process. r\_bits [0] is the most recently received (newest) code-bit; r\_bits [9] is the least recently received code-bit (oldest). r\_bits is an internal variable used exclusively by the Carrier Detect process.

#### rx\_code-bit

The rx\_code-bit parameter as delivered by the RX process, which operates in synchronism with the PMD\_UNITDATA.indicate primitive. rx\_code-bit is the most recently received code-bit from the PMD after conversion from NRZI.

#### rxerror\_status

The rxerror\_status parameter to be communicated by the Carrier Detect process through the PMA\_RXERROR.indicate primitive.

Values: NO\_ERROR; no error detected in the carrier event being received  
ERROR; the carrier event being received is in error

#### signal\_status

The signal\_status parameter as communicated by the PMD\_SIGNAL.indicate primitive.

Values: ON; the quality and level of the received signal is satisfactory  
OFF; the quality and level of the received signal is not satisfactory

#### tx\_code-bit\_in

In Link Fault Generate, the tx\_code-bit parameter as conveyed to the PMA from the PMA client by the PMA\_UNITDATA.request.

#### tx\_code-bit\_out

In Link Fault Generate, the tx\_code-bit parameter to be passed to the TX process. Note that this is called tx\_code-bit by the TX process.

### 24.3.3.3 Functions

#### SHIFTLEFT (rx\_bits)

In Carrier Detect, this function shifts rx\_bits left one bit placing rx\_bits [8] in rx\_bits [9], rx\_bits [7] in rx\_bits [8] and so on until rx\_bits [1] gets rx\_bits [0].

### 24.3.3.4 Timers

#### stabilize\_timer

An implementation-dependent delay timer between 330  $\mu$ s and 1000  $\mu$ s, inclusive, to ensure that the link is stable.

### 24.3.3.5 Counters

#### num\_cycles

In Link Fault Detect, a counter containing the number of consecutive Far-End Fault cycles currently sensed. This counter gets reset on initialization or when the bit stream fails to qualify as a potential Far-End Fault Indication. It never exceeds FEF\_CYCLES.

#### num\_ones

This represents two separate and independent counters: In Link Fault Generate, a counter containing the number of consecutive ONEs already sent during this cycle of the Far-End Fault Indication. In Link Fault Detect, a counter containing the number of consecutive ONEs currently sensed; it gets reset whenever a ZERO is detected or when the bit stream fails to qualify as a potential Far-End Fault Indication. These counters never exceed FEF\_ONES.

### 24.3.3.6 Messages

#### PMD\_UNITDATA.indicate (rx\_nrzi-bit)

A signal sent by the PMD signifying that the next nrzi-bit is available from the medium. nrzi-bit is converted (instantaneously) to code-bit by the RX process and used by the Carrier Detect process.

#### 5xPMD\_UNITDATA.indicates

In Carrier Detect, this shorthand notation represents repetition of the preceding state five times synchronized with five successive PMD\_UNITDATA.indicates.

#### PMA\_UNITDATA.request (tx\_code-bit)

A signal sent by the PMA's client signifying that the next nrzi-bit is available for transmission. For this process, the tx\_code-bit parameter is interpreted as tx\_code-bit\_in.

## 24.3.4 Process specifications and state diagrams

### 24.3.4.1 TX

The TX process passes data from the PMA's client directly to the PMD. The PMA shall implement the TX process as follows: Upon receipt of a PMA\_UNITDATA.request (tx\_code-bit), the PMA performs a conversion to NRZI format and generates a PMD\_UNITDATA.request (tx\_nrzi-bit) primitive with the same logical value for the tx\_nrzi-bit parameter. Note that tx\_code-bit is equivalent to tx\_code-bit\_out of the Link Fault Generate process when implemented.

### 24.3.4.2 RX

The RX process passes data from the PMD directly to the PMA's client and to the Carrier Detect process. The PMA shall implement the RX process as follows: Upon receipt of a PMD\_UNITDATA.indicate (rx\_nrzi-bit),

the PMA performs a conversion from NRZI format and generates a PMA\_UNITDATA.indicate (rx\_code-bit) primitive with the same logical value for the rx\_code-bit parameter.

#### 24.3.4.3 Carrier detect

The PMA Carrier Detect process provides repeater clients an indication that a carrier event has been sensed and an indication if it is deemed in error. A carrier event is defined as receipt of two non-contiguous ZEROS within any 10 rx\_code-bits. A carrier event is in error if it does not start with an SSD. The Carrier Detect process performs this function by continuously monitoring the code-bits being delivered by the RX process, and checks for specific patterns which indicate non-IDLE activity and SSD bit patterns.

The Carrier Detect process collects code-bits from the PMD RX process. r\_bits [9:0] represents a sliding, 10-bit window on the code-bit sequence, with newly received code-bits from the RX process being shifted into r\_bits [0]. The process shifts the r\_bits vector to the left, inserts the newly received code-bit into position 0, and waits for the next PMD\_UNITDATA.indicate before repeating the operation. This is depicted in Figure 24–13. The Carrier Detect process monitors the r\_bits vector until it detects two noncontiguous ZEROS in the incoming code-bit sequence. This signals a transition of carrier\_status from OFF to ON. Each new carrier is further examined for a leading SSD (1100010001) with rxerror\_status set to ERROR if it is not confirmed. A pattern of 10 contiguous ONES in the stream indicates a return to carrier\_status = OFF. Code-bit patterns of contiguous ONES correspond to IDLE code-groups in the PCS, per the encoding specified in 24.2.2.1.

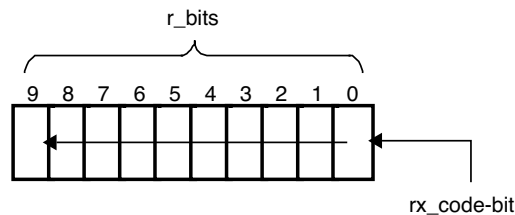


Figure 24–13—Carrier Detect reference diagram

The PMA shall, if it is supporting a repeater, implement the Carrier Detect process as depicted in Figure 24–14 including compliance with the associated state variables as specified in 24.3.3.

#### 24.3.4.4 Link Monitor

The Link Monitor process is responsible for determining whether the underlying receive channel is providing reliable data. Failure of the underlying channel typically causes the PMA's client to suspend normal actions. The Link Monitor process takes advantage of the PMD sublayer's continuously signaled transmission scheme, which provides the PMA with a continuous indication of signal detection on the channel through signal\_status as communicated by the PMD\_SIGNAL.indicate primitive. It responds to control by Auto-Negotiation, when implemented, which is effected through the link\_control parameter of PMA\_SIGNAL.request.

The Link Monitor process monitors signal\_status, setting link\_status to FAIL whenever signal\_status is OFF or when Auto-Negotiation sets link\_control to DISABLE. The link is deemed to be reliably operating when signal\_status has been continuously ON for a period of time. This period is implementation dependent but not less than 330  $\mu$ s or greater than 1000  $\mu$ s. If so qualified, Link Monitor sets link\_status to READY in order to synchronize with Auto-Negotiation, when implemented. Auto-Negotiation permits full operation by setting link\_control to ENABLE. When Auto-Negotiation is not implemented, Link Monitor operates with link\_control always set to ENABLE.

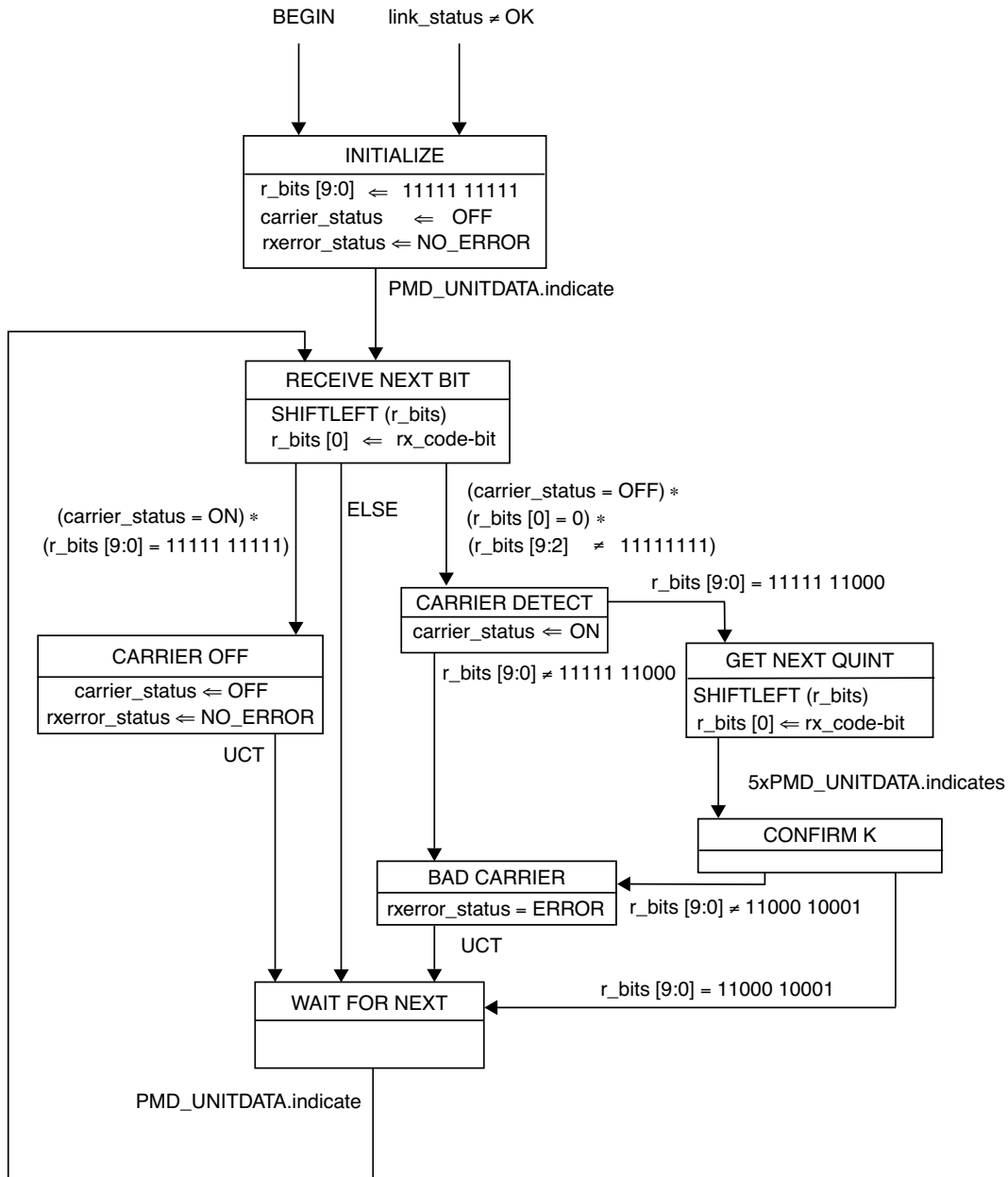


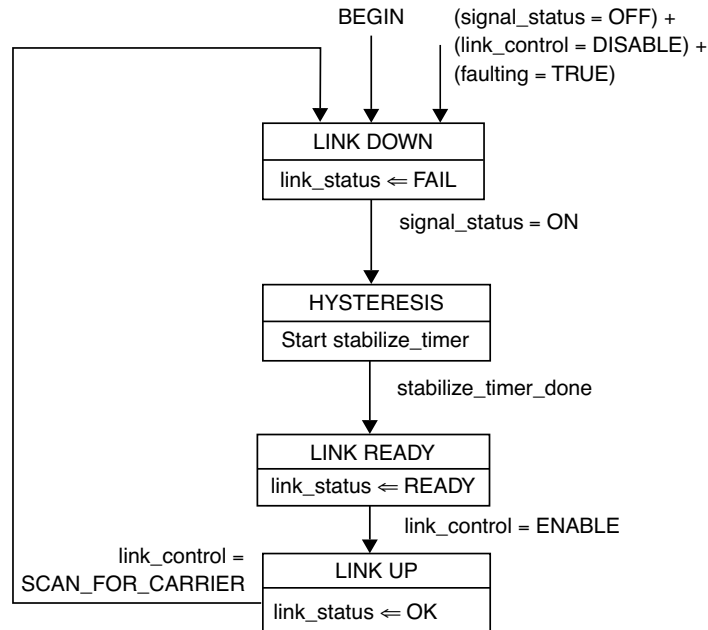
Figure 24–14—Carrier Detect state diagram

The PMA shall implement the Link Monitor process as depicted in Figure 24–15 including compliance with the associated state variables as specified in 24.3.3.

**24.3.4.5 Far-End Fault Generate**

Far-End Fault Generate simply passes tx\_code-bits to the TX process when signal\_status=ON. When signal\_status=OFF, it repetitively generates each cycle of the Far-End Fault Indication until signal\_status is reasserted.

If Far-End Fault is implemented, the PMA shall implement the Far-End Fault Generate process as depicted in Figure 24–16 including compliance with the associated state variables as specified in 24.3.3.



NOTE—The variables `link_control` and `link_status` are designated as `link_control_[TX]` and `link_status_[TX]`, respectively, by the Auto-Negotiation Arbitration state diagram (Figure 28–16).

**Figure 24–15—Link Monitor state diagram**

#### 24.3.4.6 Far-End Fault Detect

Far-End Fault Detect passively monitors the `rx_code-bit` stream from the RX process for the Far-End Fault Indication. It does so by maintaining counters for the number of consecutive ONEs seen since the last ZERO (`num_ones`) and the number of cycles of 84 ONEs and a single ZERO (`num_cycles`). The Far-End Fault Indication is denoted by three or more cycles, each of 84 ONEs and a single ZERO. Note that the number of consecutive ONEs may exceed 84 on the first cycle.

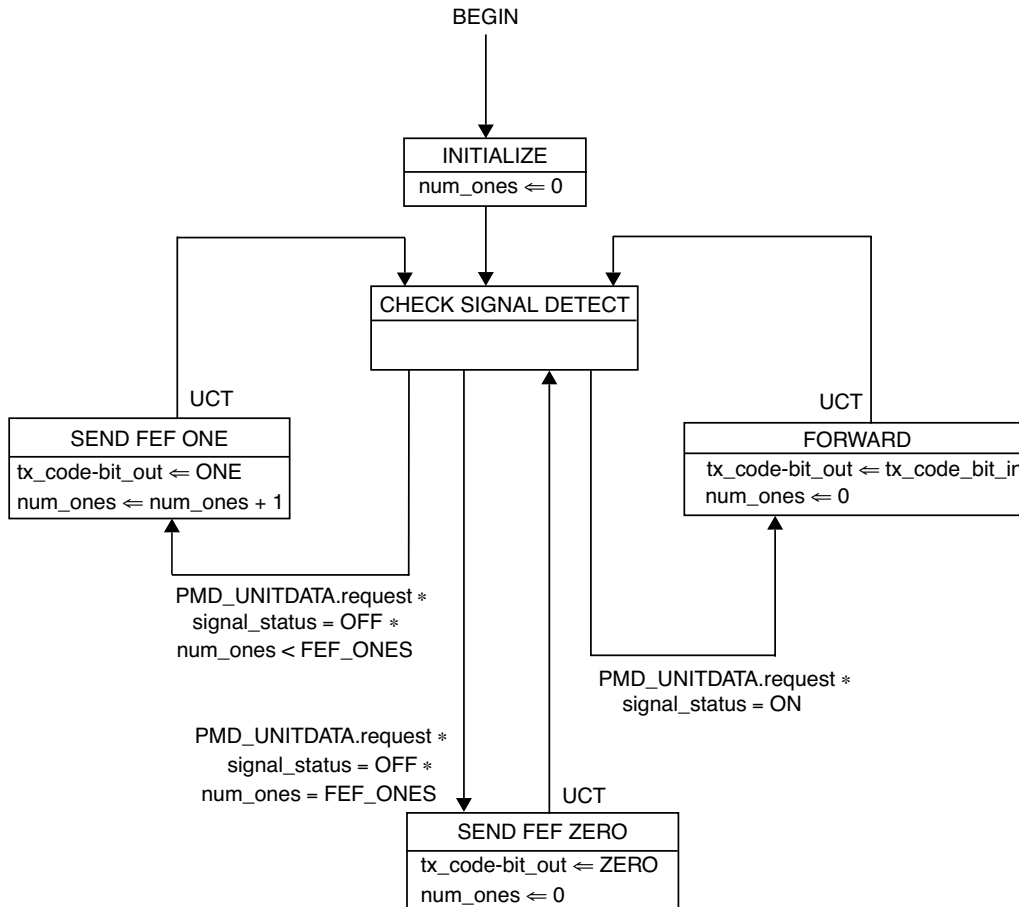
If Far-End Fault is implemented, the PMA shall implement the Far-End Fault Detect process as depicted in Figure 24–17 including compliance with the associated state variables as specified in 24.3.3.

### 24.4 Physical Medium Dependent (PMD) sublayer service interface

#### 24.4.1 PMD service interface

The following specifies the services provided by the PMD. The PMD is a sublayer within 100BASE-X and may not be present in other 100BASE-T PHY specifications. PMD services are described in an abstract manner and do not imply any particular implementation. It should be noted that these services are functionally identical to those defined in the FDDI standards, such as ISO/IEC 9314-3: 1990 and ANSI X3.263-1995, with two exceptions:

- 100BASE-X does not include a Station Management (SMT) function; therefore the PMD-to-SMT interface defined in ISO/IEC 9314-3: 1990 and ANSI X3.263-1995.
- 100BASE-X does not support multiple instances of a PMD in service to a single PMA; therefore, no qualifiers are needed to identify the unique PMD being referenced.



**Figure 24–16—Far-End Fault Generate state diagram**

There are also *editorial* differences between the interfaces specified here and in the referenced standards, as required by the context of 100BASE-X.

The PMD Service Interface supports the exchange of nrzi-bits between PMA entities. The PMD translates the nrzi-bits to and from signals suitable for the specified medium.

The following primitives are defined:

PMD\_UNITDATA.request  
 PMD\_UNITDATA.indicate  
 PMD\_SIGNAL.indicate

#### 24.4.1.1 PMD\_UNITDATA.request

This primitive defines the transfer of data (in the form of nrzi-bits) from the PMA to the PMD.

##### 24.4.1.1.1 Semantics of the service primitive

PMD\_UNITDATA.request (tx\_nrzi-bit)



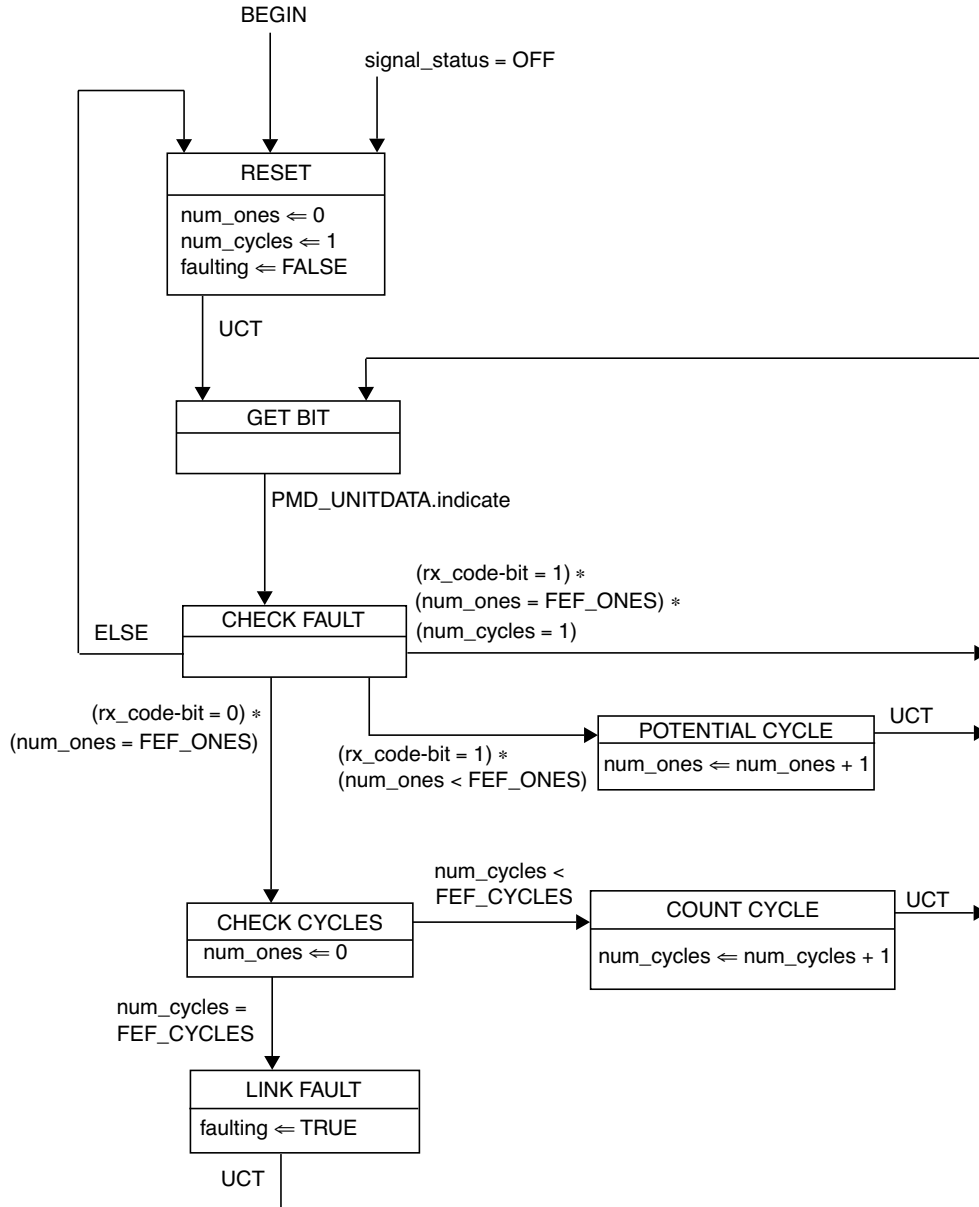


Figure 24–17 – Far-End Fault Detect state diagram

The data conveyed by PMD\_UNITDATA.request is a continuous sequence of nrzi-bits. The tx\_nrzi-bit parameter can take one of two values: ONE or ZERO.

**24.4.1.1.2 When generated**

The PMA continuously sends, at a nominal 125 Mb/s rate, the PMD the appropriate nrzi-bits for transmission on the medium.

**24.4.1.1.3 Effect of receipt**

Upon receipt of this primitive, the PMD converts the specified nrzi-bit into the appropriate signals on the MDI.

#### **24.4.1.2 PMD\_UNITDATA.indicate**

This primitive defines the transfer of data (in the form of nrzi-bits) from the PMD to the PMA.

##### **24.4.1.2.1 Semantics of the service primitive**

PMD\_UNITDATA.indicate (rx\_nrzi-bit)

The data conveyed by PMD\_UNITDATA.indicate is a continuous nrzi-bit sequence. The rx\_nrzi-bit parameter can take one of two values: ONE or ZERO.

##### **24.4.1.2.2 When generated**

The PMD continuously sends nrzi-bits to the PMA corresponding to the signals received from the MDI.

##### **24.4.1.2.3 Effect of receipt**

The effect of receipt of this primitive by the client is unspecified by the PMD sublayer.

#### **24.4.1.3 PMD\_SIGNAL.indicate**

This primitive is generated by the PMD to indicate the status of the signal being received from the MDI.

##### **24.4.1.3.1 Semantics of the service primitive**

PMD\_SIGNAL.indicate (signal\_status)

The signal\_status parameter can take on one of two values: ON or OFF, indicating whether the quality and level of the received signal is satisfactory (ON) or unsatisfactory (OFF). When signal\_status = OFF, then rx\_nrzi-bit is undefined, but consequent actions based on PMD\_SIGNAL.indicate, where necessary, interpret rx\_nrzi-bit as logic ZERO.

##### **24.4.1.3.2 When generated**

The PMD generates this primitive to indicate a change in the value of signal\_status.

##### **24.4.1.3.3 Effect of receipt**

The effect of receipt of this primitive by the client is unspecified by the PMD sublayer.

#### **24.4.2 Medium Dependent Interface (MDI)**

The MDI, a physical interface associated with a PMD, is comprised of an electrical or optical medium connector. The 100BASE-X MDIs, defined in subsequent clauses, are specified by reference to the appropriate FDDI PMD, such as in ISO/IEC 9314-3: 1990 and ANSI X3.263-1995, together with minor modifications (such as connectors and pin-outs) necessary for 100BASE-X.

#### **24.5 Compatibility considerations**

There is no requirement for a compliant device to implement or expose any of the interfaces specified for the PCS, PMA, or PMD. However, if an exposed interface is provided to the PCS, it shall comply with the requirements for the MII, as specified in Clause 22.

## 24.6 Delay constraints

In half duplex mode, proper operation of a CSMA/CD LAN demands that there be an upper bound on the propagation delays through the network. This implies that MAC, PHY, and repeater implementors must conform to certain delay minima and maxima, and that network planners and administrators conform to constraints regarding the cable topology and concatenation of devices.

In full duplex mode, predictable operation of the (optional) MAC Control PAUSE operation (Clause 31, Annex 31B) also demands that there be an upper bound on the propagation delays through the network. This implies that MAC, MAC Control sublayer, and PHY implementors must conform to certain delay maxima, and that network planners and administrators conform to constraints regarding the cable topology and concatenation of devices.

MAC constraints are contained in Clause 21. Topological constraints are contained in Clause 29. MAC Control sublayer constraints are contained in Clause 31.

The reference point for all MDI measurements is the 50% point of the mid-cell transition corresponding to the reference code-bit, as measured at the MDI. Although 100BASE-TX output is scrambled, it is assumed that these measurements are made via apparatuses that appropriately account for this.

### 24.6.1 PHY delay constraints (exposed MII)

Every 100BASE-X PHY with an exposed MII shall comply with the bit delay constraints specified in Table 24–2. These figures apply for all 100BASE-X PMDs.

**Table 24–2—Bit delay constraints**

**a) MDI to MII delay constraints (exposed MII, half duplex mode)**

Sublayer measurement points	Event	Min (bits)	Max (bits)	Input timing reference	Output timing reference
MII ↔ MDI	TX_EN sampled to MDI output	6	14	TX_CLK rising	1st bit of /J/
	MDI input to CRS assert		20	1st bit of /J/	
	MDI input to CRS de-assert (aligned)	13	24	1st bit of /T/	
	MDI input to CRS de-assert (unaligned)	13	24	1st ONE	
	MDI input to COL assert		20	1st bit of /J/	
	MDI input to COL de-assert (aligned)	13	24	1st bit of /T/	
	MDI input to COL de-assert (unaligned)	13	24	1st ONE	
	TX_EN sampled to CRS assert	0	4	TX_CLK rising	
	TX_EN sampled to CRS de-assert	0	16	TX_CLK rising	

**Table 24–3—Bit delay constraints (Continued)****b) PHY delay constraints (exposed MII, full duplex mode)**

Sublayer measurement points	Event	Min (bits)	Max (bits)	Input timing reference	Output timing reference
MII ↔ MDI	TX_EN sampled to MDI output		14	TX_CLK rising	1st bit of /J/
	MDI Input to RX_DV de-assert		32	first bit of /T/	RX_CLK rising

**24.6.2 DTE delay constraints (unexposed MII)**

Every 100BASE-X DTE with no exposed MII shall comply with the bit delay constraints specified in Table 24–3. These figures apply for all 100BASE-X PMDs.

**Table 24–4—DTE delay constraints (unexposed MII, half duplex mode)**

Sublayer measurement points	Event	Min (bits)	Max (bits)	Input timing reference	Output timing reference
MAC ↔ MDI	MAC transmit start to MDI output		18		1st bit of /J/
	MDI input to MDI output (worst-case nondeferred transmit)		54	1st bit of /J/	1st bit of /J/
	MDI input to collision detect		28	1st bit of /J/	
	MDI input to MDI output = Jam (worst case collision response)		54	1st bit of /J/	1st bit of jam

**24.6.3 Carrier de-assertion/assertion constraint (half duplex mode only)**

To ensure fair access to the network, each DTE shall, additionally, satisfy the following:

$$(\text{MAX MDI to MAC Carrier De-assert Detect}) - (\text{MIN MDI to MAC Carrier Assert Detect}) < 13$$

**24.7 Environmental specifications**

All equipment subject to this clause shall conform to the requirements of 14.7 and applicable sections of ISO/IEC 11801.

## 24.8 Protocol Implementation Conformance Statement (PICS) proforma for Clause 24, Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 100BASE-X<sup>7</sup>

### 24.8.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 24, Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 100BASE-X, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

### 24.8.2 Identification

#### 24.8.2.1 Implementation identification

Supplier	
Contact point for enquiries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s)	
NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.	
NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

#### 24.8.2.2 Protocol summary

Identification of protocol standard	IEEE Std 802.3-2002 <sup>®</sup> , Clause 24, Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 100BASE-X
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required? (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3-2002 <sup>®</sup> .)	No [ ]      Yes [ ]

Date of Statement	
-------------------	--

<sup>7</sup>Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

**24.8.2.3 Major capabilities/options**

Item	Feature	Subclause	Status	Support	Value/Comment
*DTE	Supports DTE without MII	24.4	O/I		
*REP	Supports Repeater without MII	24.4	O/I		
*MII	Supports exposed MII interface	24.4	O/I		
*PCS	Implements PCS functions	24.2	REP: O DTE: M MII: M		
PMA	Implements PMA RX, TX and Link Monitor functions	24.3	M		
*NWC	Medium capable of supporting Auto-Negotiation		O		See Clause 28
*FEF	Implements Far-End Fault	24.3.2.1	NWC: X		
NWY	Supports Auto-Negotiation (Clause 28)		NWC: O		See Clause 28

**24.8.3 PICS proforma tables for the Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 100BASE-X****24.8.3.1 General compatibility considerations**

Item	Feature	Subclause	Status	Support	Value/Comment
GN1	Compliance with MII requirements	24.4	MII:M		See Clause 22
GN2	Environmental specifications	24.7	M		

**24.8.3.2 PCS functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PS1	Transmit Bits process	24.2.3	PCS:M		
PS2	Transmit process	24.2.4.2	PCS:M		
PS3	Receive Bits process	24.2.4.3	PCS:M		
PS4	Receive process	24.2.4.4	PCS:M		
PS5	Carrier Sense process	24.2.4.5	PCS:M		

**24.8.3.3 PMA functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PA1	TX process	24.3.4.1	M		
PA2	RX process	24.3.4.2	M		
PA3	Carrier Detect process	24.3.2.1	REP: M		
PA4	Link Monitor process	24.3.4.4	M		
PA5	Far-End Fault Generate process	24.3.4.5	FEF: M		
PA6	Far-End Fault Detect process	24.3.4.6	FEF: M		

**24.8.3.4 Timing**

Item	Feature	Subclause	Status	Support	Value/Comment
TM1	Support for MII signals TX_CLK and RX_CLK	24.2.2.3	MII:M		See Clause 22
TM2	Accuracy of code-bit_timer	24.2.3	M		
TM3	Compliance with PHY bit delay constraints	24.6.1	MII:M REP: O		
TM4	Compliance with DTE bit delay constraints	24.6.2	DTE:M		
TM5	Compliance with Carrier De-assert/Assert Constraint	24.6.3	DTE:M		

## 25. Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-TX

### 25.1 Overview

This clause specifies the 100BASE-X PMD (including MDI) and baseband medium for twisted-pair wiring, 100BASE-TX. In order to form a complete 100BASE-TX Physical Layer, the 100BASE-X PMD (including MDI) shall be integrated with the 100BASE-X PCS and PMA of Clause 24, which are assumed incorporated by reference. As such, the 100BASE-TX PMD shall comply with the PMD service interface specified in 24.4.1.

### 25.2 Functional specifications

The 100BASE-TX PMD (and MDI) is specified by incorporating the FDDI TP-PMD standard, ANSI X3.263: 1995 (TP-PMD), by reference, with the modifications noted below. This standard provides support for Category 5 unshielded twisted pair (UTP) and shielded twisted pair (STP). For improved legibility in this clause, ANSI X3.263: 1995 (TP-PMD), will henceforth be referred to as TP-PMD.

### 25.3 General exceptions

The 100BASE-TX PMD is precisely the PMD specified as TP-PMD, with the following general modifications:

- a) The Scope and General description discussed in TP-PMD 1 and 5 relate to the use of those standards with an FDDI PHY, ISO/IEC 9314-1: 1989, and MAC, ISO/IEC 9314-2: 1989. These sections are not relevant to the use of the PMD with 100BASE-X.
- b) The Normative references, Definitions and Conventions of TP-PMD 2, 3, and 4 are used only as necessary to interpret the applicable sections of TP-PMD referenced in this clause.
- c) The PMD Service Specifications of TP-PMD 6 are replaced by those specified in 24.4.1. The 100BASE-TX PMD Service specification is a proper subset of the PMD Service Specification in TP-PMD.
- d) The cable plant specifications for untwisted shielded pair (UTP) of TP-PMD 11.1 are replaced by those specified in 25.4.6.
- e) There are minor terminology differences between this standard and TP-PMD that do not cause ambiguity. The terminology used in 100BASE-X was chosen to be consistent with other IEEE 802<sup>®</sup> standards, rather than with FDDI. Terminology is both defined and consistent within each standard. Special note should be made of the interpretations shown in Table 25-1.

**Table 25-1 – Interpretation of general FDDI terms and concepts**

FDDI term or concept	Interpretation for 100BASE-TX
bypass	<unused>
Connection Management (CMT)	<no comparable entity>
frame	stream
Halt Line State (HLS)	<unused>
hybrid mode	<no comparable entity>
MAC (or MAC-2)	MAC



**Table 25–1 – Interpretation of general FDDI terms and concepts (continued)**

FDDI term or concept	Interpretation for 100BASE-TX
Master Line State (MLS)	<unused>
maximum frame size = 9000 symbols	maximum stream size = 3062 code-groups
PHY (or PHY-2)	PMA; i.e., PMD client
PHY Service Data Unit (SDU)	stream
PM_SIGNAL.indication (Signal_Detect)	PMD_SIGNAL.indicate (signal_status)
PM_UNITDATA.indication (PM_Indication)	PMD_UNITDATA.indicate (nrzi-bit)
PM_UNITDATA.request (PM_Request)	PMD_UNITDATA.request (nrzi-bit)
preamble	inter-packet IDLEs
Quiet Line State (QLS)	<unused>
SM_PM_BYPASS.request (Control_Action)	Assume: SM_PM_BYPASS.request(Control_Action = Insert)
SM_PM_CONTROL.request (Control_Action)	Assume: SM_PM_CONTROL.request (Control_Action = Transmit_Enable)
SM_PM_SIGNAL.indication (Signal_Detect)	<unused>
Station Management (SMT)	<no comparable entity>
symbol	code-group

## 25.4 Specific requirements and exceptions

The 100BASE-TX PMD (including MDI) shall comply to the requirements of TP-PMD, 7, 8, 9, 10, and 11, and normative Annex A with the exceptions listed below. In TP-PMD, informative annexes B, C, E, G, I, and J, with exceptions listed below, provide additional information useful to PMD sublayer implementors. Where there is conflict between specifications in TP-PMD and those in this standard, those of this standard shall prevail.

### 25.4.1 Change to 7.2.3.1.1, “Line state patterns”

Descrambler synchronization on the Quiet Line State (QLS), Halt Line State (HLS), and Master Line State (MLS) Line State Patterns cited in TP-PMD 7.2.3.1.1 is optional.

### 25.4.2 Change to 7.2.3.3, “Loss of synchronization”

The synchronization error triggered by PH\_Invalid as defined in TP-PMD 7.2.3.3a is not applicable.

### 25.4.3 Change to Table 8-1, “Contact assignments for unshielded twisted pair”

100BASE-TX for unshielded twisted pair adopts the contact assignments of 10BASE-T. Therefore, the contact assignments shown in TP-PMD Table 8-1 shall instead be as depicted in Table 25–2.

**Table 25–2—UTP MDI contact assignments**

Contact	PHY without internal crossover MDI SIGNAL	PHY with internal crossover MDI SIGNAL
1	Transmit +	Receive +
2	Transmit –	Receive –
3	Receive +	Transmit +
4		
5		
6	Receive –	Transmit –
7		
8		

**25.4.4 Deletion of 8.3, “Station labelling”**

Clause 8.3 of TP-PMD shall not be applied to 100BASE-TX.

**25.4.5 Change to 9.1.9, “Jitter”**

The jitter measurement specified in 9.1.9 of TP-PMD may be performed using scrambled IDLEs.

**25.4.6 UTP cable plant**

The cable plant specification for unshielded twisted pair (UTP) of TP-PMD 11.1 is replaced by that specified in this subclause. The term “link segment” used in this subclause refers to a duplex channel of two pairs. Specifications for a link segment apply equally to each of the two pairs of a duplex channel. All implementations of the balanced cabling link shall be compatible at the MDI.

**25.4.6.1 Cabling system characteristics**

The cabling system used to support a 100BASE-TX duplex channel requires two pairs of Category 5 balanced cabling with a nominal impedance of 100 Ω. The cabling system components (cables, cords, and connectors) used to provide the link segment shall consist of Category 5 components as specified in ANSI/TIA/EIA-568-A:1995 and ISO/IEC 11801:1995 (Class D).

**25.4.6.2 Link transmission parameters**

The transmission parameters contained in this subclause are specified to ensure that a Category 5 link segment of up to 100 m, will provide a reliable medium. The transmission parameters of the link segment include insertion loss, characteristics impedance, return loss, NEXT loss, and external coupled noise.

**25.4.6.2.1 Insertion loss**

The insertion loss of the link segment shall be less than,

$$\text{Insertion\_Loss}(f) < 2.1f^{0.529} + 0.4/f \quad (\text{dB})$$

at all frequencies from 1 MHz to 100 MHz. This includes the attenuation of the balanced cabling pairs, including work area and equipment cables plus connector losses within the link segment.

The insertion loss specification shall be met when the link segment is terminated in 100 Ω.

NOTE—The above equation approximates the insertion loss specification at 20°C for discrete frequencies of Category 5 100-meter links specified in ANSI/TIA/EIA-568-A Annex E and in TIA/EIA TSB-67.

#### 25.4.6.2.2 Differential characteristic impedance

The nominal differential characteristic impedance of each link segment, which includes cable cords and connecting hardware, is 100 Ω for all frequencies between 1 MHz and 100 MHz.

#### 25.4.6.2.3 Return loss

Each link segment shall meet or exceed the return loss specified in the following equation at all frequencies from 1 MHz to 100 MHz.

$$\text{Return\_Loss}(f) \begin{cases} 15 & (1 - 20 \text{ MHz}) \\ 15 - 10\log_{10}(f/20) & (20 - 100 \text{ MHz}) \end{cases} \text{ (dB)}$$

where  $f$  is the frequency in MHz. The reference impedance shall be 100 Ω.

#### 25.4.6.2.4 Differential Near-End Crosstalk (NEXT)

In order to limit the crosstalk at the near end of a duplex channel, the differential pair-to-pair Near-End Crosstalk (NEXT) loss between the two pairs of a duplex channel shall be at least,

$$27.1 - 16.8\log_{10}(f/100) \text{ (dB)}$$

where  $f$  is the frequency over the range of 1 MHz to 100 MHz.

NOTE—The above equation approximates the NEXT loss specification at discrete frequencies for Category 5 100-meter links specified in ANSI/TIA/EIA-568-A Annex E and in TIA/EIA TSB-67.

#### 25.4.6.3 Noise environment

The 100BASE-TX noise environment consists of noise from external sources and could impact the objective BER. This noise may consist of sources outside the cabling that couple into the link segment via electric and magnetic fields. In addition noise from adjacent cables, referred to as alien crosstalk, may couple into the link segment. This alien crosstalk is generally present when cables are bound tightly together. To ensure robust operation a 100BASE-TX PHY should operate in the presence of an external noise as specified in 25.4.6.3.1.

##### 25.4.6.3.1 External coupled noise

The differential noise coupled from external sources that is measured at the output of a filter connected to the output of the near end of a disturbed link segment should not exceed 40 mV peak-to-peak. The filter for this measurement is a fifth order Butterworth filter with a 3 dB cutoff at 100 MHz.

**25.4.7 Replacement of 11.2, “Crossover function”**

Clause 11.2 of TP-PMD is replaced with the following:

A crossover function compliant with 14.5.2 shall be implemented except that a) the signal names are those used in TP-PMD, and b) the contact assignments for STP are those shown in Table 8-2 of TP-PMD. Note that compliance with 14.5.2 implies a recommendation that crossover (for both UTP and STP) be performed within repeater PHYs.

**25.4.8 Change to A.2, “DDJ test pattern for baseline wander measurements”**

The length of the test pattern specified in TP-PMD annex A.2 may be shortened to accommodate feasible 100BASE-X measurements, but shall not be shorter than 3000 code-groups.

NOTE—This pattern is to be applied to the MII. (When applied to the MAC, the nibbles within each byte are to be swapped. E.g., as delivered to the MAC, the test pattern would start, "60 c9 16 ...".)

**25.4.9 Change to annex G, “Stream cipher scrambling function”**

An example of a stream cipher scrambling implementation is shown in TP-PMD annex G. This may be modified to allow synchronization solely on the IDLE sequences between packets.

**25.4.10 Change to annex I, “Common mode cable termination”**

The contact assignments shown in TP-PMD figures I-1 and I-2 shall instead comply with those specified in Table 25-2.

## 25.5 Protocol Implementation Conformance Statement (PICS) proforma for Clause 25, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-TX<sup>8</sup>

### 25.5.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 25, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-TX, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

### 25.5.2 Identification

#### 25.5.2.1 Implementation identification

Supplier	
Contact point for enquiries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s)	
NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.	
NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

#### 25.5.2.2 Protocol summary

Identification of protocol standard	IEEE Std 802.3-2002 <sup>®</sup> , Clause 25, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-TX
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required? No [ ] Yes [ ] (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3-2002 <sup>®</sup> .)	

Date of Statement	
-------------------	--

<sup>8</sup>Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

**25.5.3 Major capabilities/options**

Item	Feature	Subclause	Status	Support	Value/Comment
*TXU	Supports unshielded twisted pair	25.2	O/1		
TXS	Supports shielded twisted pair	25.2	O/1		

**25.5.4 PICS proforma tables for the Physical Medium Dependent (PMD) sublayer and base-band medium, type 100BASE-TX****25.5.4.1 General compatibility considerations**

Item	Feature	Subclause	Status	Support	Value/Comment
GN1	Integrates 100BASE-X PMA and PCS	25.1	M		See Clause 24

**25.5.4.2 PMD compliance**

Item	Feature	Subclause	Status	Support	Value/Comment
PD1	Compliance with 100BASE-X PMD Service Interface	25.1	M		See 24.2.3
PD2	Compliance with ANSI X3.263-1995, 7, 8 (excluding 8.3), 9, 10, 11 and normative annex A, with listed exceptions	25.4 25.4.5	M		
PD3	Precedence over ANSI X3.263-1995	25.4	M		
PD4	MDI contact assignments for unshielded twisted pair	25.4.4 25.4.3	TXU: M		
PD5	Compliance with crossover function of 14.5.2 with listed adaptations	25.4.8	M		
PD6	Minimum jitter test pattern length	25.4.9	M		3000 code-groups

**25.5.4.3 Characteristics of link segment**

Item	Feature	Subclause	Status	Support	Value/Comment
LKS1	Implementations of balanced cabling link	25.4.6	M		Compatible at the MDI.
LKS2	100BASE-T links	25.4.6.1	M		Category 5 components as specified in ANSI/TIA/EIA-568-A:1995 and ISO/IEC 11801:1995 (Class D).
LKS3	Insertion loss	25.4.6.2.1	M		As specified in 25.4.6.2.1 at all frequencies from 1 MHz to 100 MHz when link segment is terminated in 100 $\Omega$ .
LKS4	Return loss	25.4.6.2.3	M		As specified in 25.4.6.2.3 at all frequencies from 1 MHz to 100 MHz when link segment is terminated in 100 $\Omega$ .
LKS5	Differential Near-End Crosstalk (NEXT)	25.4.6.2.4	M		As specified in 25.4.6.2.4 at all frequencies from 1 MHz to 100 MHz.

## 26. Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-FX

### 26.1 Overview

This clause specifies the 100BASE-X PMD (including MDI) and fiber optic medium for multi-mode fiber, 100BASE-FX. In order to form a complete 100BASE-FX Physical Layer it shall be integrated with the 100BASE-X PCS and PMA of Clause 24, which are assumed incorporated by reference. As such, the 100BASE-FX PMD shall comply with the PMD service interface specified in 24.4.1.

### 26.2 Functional specifications

The 100BASE-FX PMD (and MDI) is specified by incorporating the FDDI PMD standard, ISO/IEC 9314-3: 1990, by reference, with the modifications noted below. This standard provides support for two optical fibers. For improved legibility in this clause, ISO/IEC 9314-3: 1990 will henceforth be referred to as fiber-PMD.

### 26.3 General exceptions

The 100BASE-FX PMD is precisely the PMD specified as fiber-PMD, with the following general modifications:

- a) The Scope and General description discussed in fiber-PMD 1 and 5 relate to the use of those standards with an FDDI PHY, ISO/IEC 9314-1: 1989, and MAC, ISO/IEC 9314-2: 1989. These clauses are not relevant to the use of the PMD with 100BASE-X.
- b) The Normative references, Definitions and Conventions of fiber-PMD 2, 3, and 4 are used only as necessary to interpret the applicable sections of fiber-PMD referenced in this clause.
- c) The PMD Service Specifications of fiber-PMD 6 are replaced by those specified in 24.4.1. The 100BASE-FX PMD Service specification is a proper subset of the PMD service specification in fiber-PMD.
- d) There are minor terminology differences between this standard and fiber-PMD that do not cause ambiguity. The terminology used in 100BASE-X was chosen to be consistent with other IEEE 802® standards, rather than with FDDI. Terminology is both defined and consistent within each standard. Special note should be made of the interpretations shown in Table 26-1.

**Table 26-1 – Interpretation of general FDDI terms and concepts**

FDDI term or concept	Interpretation for 100BASE-X
bypass	<unused>
Connection Management (CMT)	<no comparable entity>
frame	stream
Halt Line State (HLS)	<unused>
hybrid mode	<no comparable entity>
MAC (or MAC-2)	MAC
Master Line State (MLS)	<unused>
maximum frame size = 9000 symbols	maximum stream size = 3062 code-groups



**Table 26–1—Interpretation of general FDDI terms and concepts (continued)**

FDDI term or concept	Interpretation for 100BASE-X
PHY (or PHY-2)	PMA; i.e., PMD client
PHY Service Data Unit (SDU)	stream
PM_SIGNAL.indication (Signal_Detect)	PMD_SIGNAL.indicate (signal_status)
PM_UNITDATA.indication (PM_Indication)	PMD_UNITDATA.indicate (nrzi-bit)
PM_UNITDATA.request (PM_Request)	PMD_UNITDATA.request (nrzi-bit)
preamble	inter-packet IDLEs
Quiet Line State (QLS)	<unused>
SM_PM_BYPASS.request (Control_Action)	Assume: SM_PM_BYPASS.request (Control_Action = Insert)
SM_PM_CONTROL.request (Control_Action)	Assume: SM_PM_CONTROL.request (Control_Action = Transmit_Enable)
SM_PM_SIGNAL.indication (Signal_Detect)	<unused>
Station Management (SMT)	<no comparable entity>
symbol	code-group

## 26.4 Specific requirements and exceptions

The 100BASE-FX PMD (including MDI) and baseband medium shall conform to the requirements of fiber-PMD Clauses 8, 9, and 10. In the referenced standard, fiber-PMD, informative Annexes A through G provide additional information useful to PMD sublayer implementors. Where there is conflict between specifications in fiber-PMD and those in this standard, those of this standard shall prevail.

### 26.4.1 Medium Dependent Interface (MDI)

The 100BASE-FX medium dependent interface (MDI) shall conform to one of the following connectors. The recommended alternative is the Low Cost Fibre Optical Interface Connector.

- a) Low Cost Fibre Optical Interface Connector (commonly called the duplex SC connector) as specified in ANSI X3.237-1995, 7.1.1 through 7.3.1, inclusive.
- b) Media Interface Connector (MIC) as specified in fiber-PMD 7 and Annex F. When the MIC is used, the receptacle shall be keyed as “M”.
- c) Optical Medium Connector Plug and Socket (commonly called ST connector) as specified in 15.3.2.

### 26.4.2 Crossover function

A crossover function shall be implemented in every cable-pair link. The crossover function connects the transmitter of one PHY to the receiver of the PHY at the other end of the cable-pair link. For 100BASE-FX, the crossover function is realized in the cable plant.

**26.5 Protocol Implementation Conformance Statement (PICS) proforma for Clause 26, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-FX<sup>9</sup>**

**26.5.1 Introduction**

The supplier of a protocol implementation that is claimed to conform to Clause 26, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-FX, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

**26.5.2 Identification**

**26.5.2.1 Implementation identification**

Supplier	
Contact point for enquiries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s)	
NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.	
NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier’s terminology (e.g., Type, Series, Model).	

**26.5.3 Protocol summary**

Identification of protocol standard	IEEE Std 802.3-2002 <sup>®</sup> , Clause 26, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-FX
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required? (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3-2002 <sup>®</sup> .)	No [ ]      Yes [ ]

Date of Statement	
-------------------	--

<sup>9</sup>Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

### 26.5.4 Major capabilities/options

Item	Feature	Subclause	Status	Support	Value/Comment
FSC	Supports Low Cost Fibre Optical Interface Connector (duplex SC)	26.4.1	O/1		Recommended. See ANSI X3.237-1995, 7.1.1 through 7.3.1
*FMC	Supports Media Interface Connector (MIC)	26.4.1	O/1		See ISO/IEC 9314-3: 1990, 7 and Annex F
FST	Supports Optical Medium Connector Plug and Socket (ST)	26.4.1	O/1		See 15.3.2

### 26.5.5 PICS proforma tables for Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-FX

#### 26.5.5.1 General compatibility considerations

Item	Feature	Subclause	Status	Support	Value/Comment
GN1	Integrates 100BASE-X PMA and PCS	26.1	M		See Clause 24

#### 26.5.5.2 PMD compliance

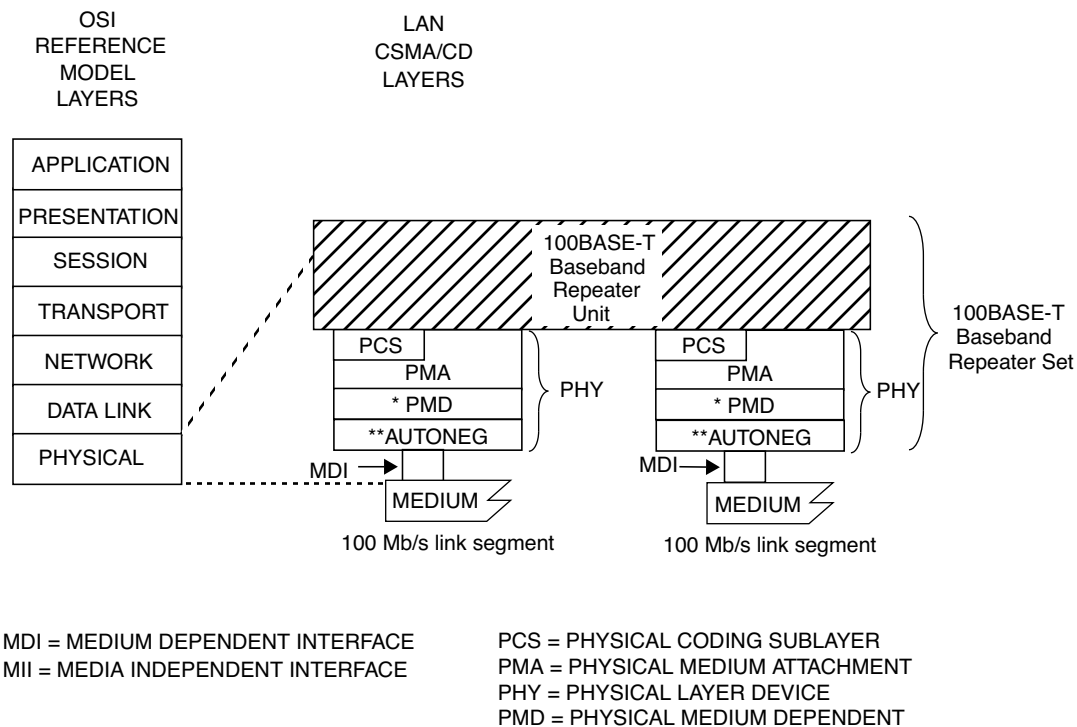
Item	Feature	Subclause	Status	Support	Value/Comment
PD1	Compliance with 100BASE-X PMD Service Interface	26.1	M		See 24.4.1
PD2	Compliance with ISO/IEC 9314-3: 1990 8, 9, and 10	26.4	M		
PD3	Precedence over ISO/IEC 9314-3: 1990	26.4	M		
PD4	MIC receptacle keying	26.4.1	FMC: M		“M”
PD5	Crossover function in cable	26.4.2	M		

## 27. Repeater for 100 Mb/s baseband networks

### 27.1 Overview

#### 27.1.1 Scope

Clause 27 defines the functional and electrical characteristics of a repeater for use with 100BASE-T 100 Mb/s baseband networks. A repeater for any other IEEE 802.3<sup>®</sup> network type is beyond the scope of this clause. The relationship of this standard to the entire ISO/IEC 8802-3 CSMA/CD LAN standard is shown in Figure 27–1. The purpose of the repeater is to provide a simple, inexpensive, and flexible means of coupling two or more segments.



\* PMD is specified for 100BASE-TX and -FX only; 100BASE-T4 does not use this layer.  
Use of MII between PCS and baseband repeater unit is optional.  
\*\* AUTONEG is optional.

**Figure 27–1 — 100BASE-T repeater set relationship to the ISO/IEC OSI reference model**

#### 27.1.1.1 Repeater set

Repeater sets are an integral part of all 100 Mb/s baseband networks with more than two DTEs and are used to extend the physical system topology by providing a means of coupling two or more segments. Multiple repeater sets are permitted within a single collision domain to provide the maximum connection path length. Segments may be connected directly by a repeater or a pair of repeaters that are, in turn, connected by an inter-repeater link (IRL). Allowable topologies shall contain only one operative signal path between any two

points on the network. A repeater set is not a station and does not count toward the overall limit of 1024 stations on a network.

A repeater set can receive, and if necessary decode, data from any segment under worst-case noise, timing, and signal amplitude conditions. It retransmits the data to all other segments attached to it with timing, amplitude, and, if necessary, coding restored. The retransmission of data occurs simultaneously with reception. If a collision occurs, the repeater set propagates the collision event throughout the network by transmitting a Jam signal. A repeater set also provides a degree of protection to a network by isolating a faulty segment's carrier activity from propagating through the network.

### **27.1.1.2 Repeater unit**

A repeater unit is a subset of a repeater set containing all the repeater-specific components and functions, exclusive of PHY components and functions. A repeater unit connects to the PMA and, if necessary, the PCS sublayers of its PHYs.

### **27.1.1.3 Repeater classes**

Two classes of repeater sets are defined—Class I and Class II.

Class I:

A type of repeater set specified such that in a maximum length segment topology, only one such repeater set may exist between any two DTEs within a single collision domain.

Class II:

A type of repeater set specified such that in a maximum length segment topology, only two such repeater sets may exist between any two DTEs within a single collision domain.

More complex topologies are possible in systems that do not use worst-case cable. See Clause 29 for requirements.

## **27.1.2 Application perspective**

This subclause states the broad objectives and assumptions underlying the specification defined through Clause 27.

### **27.1.2.1 Objectives**

- a) Provide physical means for coupling two or more LAN segments at the Physical Layer.
- b) Support interoperability of independently developed physical, electrical, and optical interfaces.
- c) Provide a communication channel with a mean bit error rate, at the physical service interface equivalent to that for the attached PHY.
- d) Provide for ease of installation and service.
- e) Ensure that fairness of DTE access is not compromised.
- f) Provide for low-cost networks, as related to both equipment and cabling.
- g) Make use of building wiring appropriate for the supported PHYs and telephony wiring practices.

### **27.1.2.2 Compatibility considerations**

All implementations of the repeater set shall be compatible at the MDI. The repeater set is defined to provide compatibility among devices designed by different manufacturers. Designers are free to implement circuitry within the repeater set in an application-dependent manner provided the appropriate PHY specifications are met.

### 27.1.2.2.1 Internal segment compatibility

Implementations of the repeater set that contain a MAC layer for network management or other purposes, irrespective of whether they are connected through an exposed repeater port or are internally ported, shall conform to the requirements of Clause 30 on that port if repeater management is implemented.

### 27.1.3 Relationship to PHY

A close relationship exists between Clause 27 and the PHY clauses, Clause 23 for the 100BASE-T4 PHY and Clauses 24 to 26 for the 100BASE-X PHYs, and Clause 32 for the 100BASE-T2 PHY. The PHY's PMA, PCS, and MDI specifications provide the actual medium attachment, including drivers, receivers, and Medium Interface Connectors for the various supported media. The repeater clause does not define a new PHY; it utilizes the existing PHYs complete and without modification.

## 27.2 PMA interface messages

The messages between the repeater unit and the PMA in the PHY utilizes the PMA service interface defined in 23.3, 24.3, and 32.4.2. The PMA service interface primitives are summarized below:

```
PMA_TYPE.indicate
PMA_UNITDATA.request
PMA_UNITDATA.indicate
PMA_CARRIER.indicate
PMA_LINK.indicate
PMA_RXERROR.indicate
```

## 27.3 Repeater functional specifications

A repeater set provides the means whereby data from any segment can be received under worst case noise, timing, and amplitude conditions and then retransmitted with timing and amplitude restored to all other attached segments. Retransmission of data occurs simultaneously with reception. If a collision occurs, the repeater set propagates the collision event throughout the network by transmitting a Jam signal. If an error is received by the repeater set, no attempt is made to correct it and it is propagated throughout the network by transmitting an invalid signal.

The repeater set provides the following functional capability to handle data flow between ports:

- a) *Signal restoration.* Provides the ability to restore the timing and amplitude of the received signal prior to retransmission.
- b) *Transmit function.* Provides the ability to output signals on the appropriate port and encoded appropriately for that port. Details of signal processing are described in the specifications for the PHYs.
- c) *Receive function.* Provides the ability to receive input signals presented to the ports. Details of signal processing are described in the specifications for the PHYs.
- d) *Data Handling function.* Provides the ability to transfer code-elements between ports in the absence of a collision.
- e) *Received Event Handling requirement.* Provides the ability to derive a carrier signal from the input signals presented to the ports.
- f) *Collision Handling function.* Provides the ability to detect the simultaneous reception of frames at two or more ports and then to propagate a Jam message to all connected ports.
- g) *Error Handling function.* Provides the ability to prevent substandard links from generating streams of false carrier and interfering with other links.
- h) *Partition function.* Provides the ability to prevent a malfunctioning port from generating an excessive number of consecutive collisions and indefinitely disrupting data transmission on the network.

- i) *Receive Jabber function.* Provides the ability to interrupt the reception of abnormally long streams of input data.

### **27.3.1 Repeater functions**

The repeater set shall provide the Signal Restoration, Transmit, Receive, Data Handling, Received Event Handling, Collision Handling, Error Handling, Partition, and Receive Jabber functions. The repeater is transparent to all network acquisition activity and to all DTEs. The repeater will not alter the basic fairness criterion for all DTEs to access the network or weigh it toward any DTE or group of DTEs regardless of network location.

The Transmit and Receive functional requirements are specified by the PHY clauses, Clause 23 for 100BASE-T4, Clauses 24 to 26 for 100BASE-X, and Clause 32 for 100BASE-T2.

#### **27.3.1.1 Signal restoration functional requirements**

##### **27.3.1.1.1 Signal amplification**

The repeater set (including its integral PHYs) shall ensure that the amplitude characteristics of the signals at the MDI outputs of the repeater set are within the tolerances of the specification for the appropriate PHY type. Therefore, any loss of signal-to-noise ratio due to cable loss and noise pickup is regained at the output of the repeater set as long as the incoming data is within system specification.

##### **27.3.1.1.2 Signal wave-shape restoration**

The repeater set (including its integral PHYs) shall ensure that the wave-shape characteristics of the signals at the MDI outputs of a repeater set are within the specified tolerance for the appropriate PHY type. Therefore, any loss of wave-shape due to PHYs and media distortion is restored at the output of the repeater set.

##### **27.3.1.1.3 Signal retiming**

The repeater set (including its integral PHYs) shall ensure that the timing of the encoded data output at the MDI outputs of a repeater set are within the specified tolerance for the appropriate PHY type. Therefore, any receive jitter from the media is removed at the output of the repeater set.

#### **27.3.1.2 Data handling functional requirements**

##### **27.3.1.2.1 Data frame forwarding**

The repeater set shall ensure that the data frame received on a single input port is distributed to all other output ports in a manner appropriate for the PHY type of that port. The data frame is that portion of the packet after the SFD and before the end-of-frame delimiter. The only exceptions to this rule are when contention exists among any of the ports, when the receive port is partitioned as defined in 27.3.1.6, when the receive port is in the Jabber state as defined in 27.3.1.7, or when the receive port is in the Link Unstable state as defined in 27.3.1.5.1. Between unpartitioned ports, the rules for collision handling (see 27.3.1.4) take precedence.

##### **27.3.1.2.2 Received code violations**

The repeater set shall ensure that any code violations received while forwarding a packet are propagated to all outgoing segments. These code violations shall be forwarded as received or replaced by bad\_code (see 23.2.1.2), /H/ (see 24.2.2.1), or (ESC/0, 0/ESC) (see Figure 32–9) code-groups, as appropriate for the outgoing PHY type. Once a received code violation has been replaced by the bad\_code, /H/, or (ESC/0, 0/ESC) code-groups, this substitution shall continue for the remainder of the packet regardless of its content. The only exception to this rule is when contention exists among any of the ports, where the rules for collision handling (see 27.3.1.4) then take precedence.

### 27.3.1.3 Received event handling functional requirements

#### 27.3.1.3.1 Received event handling

For all its ports, the repeater set shall implement a function (*scarrier\_present*) that represents a received event. Received events include both the Data Frame and any encapsulation of the Data Frame such as Preamble, SFD, and the code-groups /H/, /J/, /K/, *bad\_code*, *eop*, /T/, /R/, SSD, ESD, etc. A received event is exclusive of the IDLE pattern. Upon detection of *scarrier\_present* from one port, the repeater set repeats all received signals in the Data Frame from that port to the other port (or ports) as described in Figure 27–2.

#### 27.3.1.3.2 Preamble regeneration

The repeater set shall output preamble as appropriate for the outgoing PHY type followed by the SFD.

#### 27.3.1.3.3 Start-of-packet propagation delay

The start-of-packet propagation delay for a repeater set is the time delay between the start of the packet (see 24.6, 23.11.3, and 32.12.1) on its repeated-from (input) port to the start of the packet on its repeated-to (output) port (or ports). This parameter is referred to as the SOP delay. The maximum value of this delay is constrained by Table 27–2.

#### 27.3.1.3.4 Start-of-packet variability

The start-of-packet variability for a repeater set is defined as the total worst-case difference between start-of-packet propagation delays for successive packets separated by 104 bit times (BT) or less at the same input port. The variability shall be less than or equal to those specified in Table 27–1.

**Table 27–1 — Start-of-packet variability**

Input port type	Variability (BT)
100BASE-FX	7.0
100BASE-TX	7.0
100BASE-T4	8.0
100BASE-T2	8.0

### 27.3.1.4 Collision handling functional requirements

#### 27.3.1.4.1 Collision detection

The repeater performs collision detection by monitoring all its enabled input ports for received events. When the repeater detects received events on more than one input port, it shall enter a collision state and transmit the Jam message to all of its output ports.

#### 27.3.1.4.2 Jam generation

While a collision is occurring between any of its ports, the repeater unit shall transmit the Jam message to all of the PMAs to which it is connected. The Jam message shall be transmitted in accordance with the repeater state diagram in Figure 27–4 and Figure 27–5.



#### 27.3.1.4.3 Collision-jam propagation delay

The start-of-collision Jam propagation delay for a repeater set is the time delay between the start of the second packet input signals to arrive at its port and the start of Jam (see 24.6, 23.11, and 32.12.1) out on all ports. This parameter is referred to as the SOJ delay. The delay shall be constrained by Table 27–2. Note that a device defined by two columns in Table 27–2 must meet the performance requirements specified in both columns.

**Table 27–2—Start-of-packet propagation and start-of-collision jam propagation delays**

Class I repeater	Class II repeater with all ports TX/FX	Class II repeater with any port T4	Class II repeater with any port T2
SOP + SOJ ≤ 140 BT	SOP ≤ 46 BT, SOJ ≤ 46 BT	SOP+SOJ ≤ 67 BT	SOP+SOJ ≤ 90 BT

#### 27.3.1.4.4 Cessation-of-collision Jam propagation delay

The cessation-of-collision Jam propagation delay for a repeater set is the time delay between the end of the packet (see 24.6 and 23.11.3) that creates a state such that Jam should end at a port and the end of Jam (see 24.6 and 23.11.3) at that port. The states of the input signals that should cause Jam to end are covered in detail in the repeater state diagrams. This parameter is referred to as the EOJ delay. The delay shall be constrained by Table 27–3.

**Table 27–3—Cessation-of-collision Jam propagation delay**

Class I repeater	Class II repeater
EOJ ≤ SOP	EOJ ≤ SOP

#### 27.3.1.5 Error handling functional requirements

##### 27.3.1.5.1 100BASE-X and 100BASE-T2 carrier integrity functional requirements

In 100BASE-TX, 100BASE-FX, and 100BASE-T2 systems, it is desirable that the repeater set protect the network from some transient fault conditions that would disrupt network communications. Potential likely causes of such conditions are DTE and repeater power-up and power-down transients, cabling disconnects, and faulty wiring.

Each 100BASE-TX, 100BASE-FX, and 100BASE-T2 repeater PMA interface shall contain a self-interrupt capability, as described in Figure 27–9, to prevent a segment's spurious carrier activity from reaching the repeater unit and hence propagating through the network.

The repeater PMA interface shall count consecutive false carrier events. A false carrier event is defined as a carrier event that does not begin with a valid start of stream delimiter (see 24.2.2.1.4 and 32.3.1.2.3). The count shall be incremented on each false carrier event and shall be reset on reception of a valid carrier event. In addition, each PMA interface shall contain a false carrier timer, which is enabled at the beginning of a false carrier event and reset at the conclusion of such an event. A repeater unit shall transmit the JAM message to all of the PMAs to which it is connected for the duration of the false carrier event or until the duration of the event exceeds the time specified by the false\_carrier\_timer (see 27.3.2.1.4), whichever is shorter. The JAM message shall be transmitted in accordance with the Repeater state diagram in Figure 27–4 and Figure 27–5. The LINK UNSTABLE condition shall be detected when the False Carrier Count exceeds

the value FCCLimit (see 27.3.2.1.1) or the duration of a false carrier event exceeds the time specified by the false\_carrier\_timer. In addition, the LINK UNSTABLE condition shall be detected upon power-up reset.

Upon detection of LINK UNSTABLE, the port shall perform the following:

- a) Inhibit sending further messages to the repeater unit.
- b) Inhibit sending further output messages from the repeater unit.
- c) Continue to monitor activity on that PMA interface.

The repeater shall exit the LINK UNSTABLE condition when one of the following is met:

- a) The repeater has detected no activity (Idle) for more than the time specified by ipg\_timer plus idle\_timer (see 27.3.2.1.4) on Port X.
- b) A valid carrier event with a duration greater than the time specified by valid\_carrier\_timer (see 27.3.2.1.4) has been received, preceded by no activity (Idle) for more than the time specified by ipg\_timer (see 27.3.2.1.4) on Port X.

### 27.3.1.5.2 Speed handling

If the PHY has the capability of detecting speeds other than 100 Mb/s, then the repeater set shall have the capability of blocking the flow of non-100 Mb/s signals. The incorporation of 100 Mb/s and 10 Mb/s repeater functionality within a single repeater set is beyond the scope of this standard.

### 27.3.1.6 Partition functional requirements

In large multisegment networks it may be desirable that the repeater set protect the network from some fault conditions that would disrupt network communications. A potentially likely cause of this condition could be due to a cable fault.

Each repeater PMA interface shall contain a self-interrupt capability, as described in Figure 27–8, to prevent a faulty segment's carrier activity from reaching the repeater unit and hence propagating through the network. The repeater PMA interface shall count collisions. The count shall be incremented on each transmission that suffers a collision. The count shall be reset on a carrier event of duration in excess of no\_collision\_timer (see 27.3.2.1.4) without incurring a collision. If this count reaches the value CCLimit (see 27.3.2.1.1), the Partition condition shall be detected.

Upon detection of Partition, the port shall perform the following:

- a) Inhibit sending further input messages to the repeater unit.
- b) Continue to output messages from the repeater unit.
- c) Continue to monitor activity on that PMA interface.

The repeater shall reset the Partition function when one of the following conditions is met:

- a) On power-up reset.
- b) The repeater has transmitted on the port for a duration in excess of no\_collision\_timer (see 27.3.2.1.4) without incurring a collision.

NOTE—It is possible that under some network conditions the partition state machine will partition a port due to normal network collisions rather than a fault condition. It is also possible that some double fault conditions will remain undetected. To reduce the likelihood of these events occurring, the following optional measures, as described in Figure 27–8, are recommended:

- a) The collision count is additionally reset when the repeater has transmitted on the port for a duration in excess of no\_collision\_timer without detecting a collision.

- b) The Partition function is additionally reset when the repeater has received activity on the port for a duration in excess of `no_collision_timer` without detecting a collision.
- c) The Partition condition is additionally detected due to a carrier event of duration in excess of `jabber_timer` (see 27.3.1.7) in which a collision has occurred.

### 27.3.1.7 Receive jabber functional requirements

Each repeater PMA interface shall contain a self-interrupt capability, as described in Figure 27–7, to prevent an illegally long reception of data from reaching the repeater unit. The repeater PMA interface shall provide a window of duration `jabber_timer` bit times (see 27.3.2.1.4) during which the input messages may be passed on to other repeater unit functions. If a reception exceeds this duration, the jabber condition shall be detected.

Upon detection of jabber, the port shall perform the following:

- a) Inhibit sending further input messages to the repeater unit.
- b) Inhibit sending further output messages from the repeater unit.

The repeater PMA interface shall reset the Jabber function and re-enable data transmission and reception when either one of the following conditions is met:

- a) On power-up reset.
- b) When carrier is no longer detected.

## 27.3.2 Detailed repeater functions and state diagrams

A precise algorithmic definition is given in this subclause, providing a complete procedural model for the operation of a repeater, in the form of state diagrams. Note that whenever there is any apparent ambiguity concerning the definition of repeater operation, the state diagrams should be consulted for the definitive statement.

The model presented in this subclause is intended as a primary specification of the functions to be provided by any repeater unit. It is important to distinguish, however, between the model and a real implementation. The model is optimized for simplicity and clarity of presentation, while any realistic implementation should place heavier emphasis on such constraints as efficiency and suitability to a particular implementation technology.

It is the functional behavior of any repeater unit implementation that shall match the standard, not the internal structure. The internal details of the procedural model are useful only to the extent that they help specify the external behavior clearly and precisely. For example, the model uses a separate Receive Port Jabber state diagram for each port. However, in actual implementation, the hardware may be shared.

The notation used in the state diagram follows the conventions of 1.2.1. Note that transitions shown without source states are evaluated at the completion of every state and take precedence over other transition conditions.

### 27.3.2.1 State diagram variables

#### 27.3.2.1.1 Constants

##### CCLimit

The number of consecutive collisions that must occur before a segment is partitioned.

Values: Positive integer greater than 60.

##### FCCLimit

The number of consecutive False Carrier events that must occur before a segment is isolated.

Value: 2.

**27.3.2.1.2 Variables****activity**(Port designation)

Indicates port activity status. The repeater core effects a summation of this variable received from all its attached ports and responds accordingly.

Values: 0; no frame or packet activity at any port.  
 1; exactly 1 port of the repeater set has frame or packet activity input.  
 >1; more than 1 port of the repeater set has frame or packet activity input. Alternately, one or more ports has detected a carrier that is not valid.

**all\_data\_sent**

Indicates if all received data frame bits or code-groups from the current frame have been sent. During or after collision the all\_data\_sent variable follows the inverse of the carrier of port N.

Values: true; all received data frame bits or code-groups have been sent.  
 false; all received data frame bits or code-groups have not been sent.

**begin**

The Interprocess flag controlling state diagram initialization values.

Values: true  
 false

**carrier\_status**(X)

Signal received from PMA; indicates the status of sourced Carrier input at Port X.

Values: ON; the carrier\_status parameter of the PMA\_CARRIER.indicate primitive for Port X is ON.  
 OFF; the carrier\_status parameter of the PMA\_CARRIER.indicate primitive for Port X is OFF.

**data\_ready**

Indicates if the repeater has detected and/or decoded the MAC SFD and is ready to send the received data.

Values: true; the MAC SFD has been detected and/or decoded.  
 false; the MAC SFD has not been detected nor decoded.

**force\_jam**(X)

Flag from Carrier Integrity state diagram for Port X, which determines whether all ports should transmit Jam.

Values: true; the Carrier Integrity Monitor has determined that it requires all ports be forced to transmit Jam.  
 false; the Carrier Integrity Monitor has determined that it does not require all ports be forced to transmit Jam.

Default: for T4 ports: false

**isolate**(X)

Flag from Carrier Integrity state diagram for Port X, which determines whether a port should be enabled or disabled.

Values: true; the Carrier Integrity Monitor has determined the port should be disabled.  
 false; the Carrier Integrity Monitor has determined the port should be enabled.

**jabber**(X)

Flag from Receive Timer state diagram for Port X, which indicates that the port has received excessive length activity.

Values: true; port has exceeded the continuous activity limit.

false; port has not exceeded the continuous activity limit.

#### link\_status(X)

Signal received from PMA; indicates link status for Port X (see 23.3.5, 24.3.1.5, and 32.4.1.3.1).

Values: OK; the link\_status parameter of the PMA\_LINK.indicate primitive for Port X is OK.  
 READY; the link\_status parameter of the PMA\_LINK.indicate primitive for Port X is READY (for 100BASE-TX and 100BASE-T4).  
 FAIL; the link\_status parameter of the PMA\_LINK.indicate primitive for Port X is FAIL.

#### opt(X)

Implementation option. Either value may be chosen for repeater implementation.

Values: true; port will emit the JamT4 pattern in response to collision conditions.  
 false; port will append Jam pattern after preamble and SFD in response to collision conditions.

#### OUT(X)

Type of output repeater is sourcing at Port X.

Values: Idle; repeater is transmitting an IDLE pattern as described by 23.4.1.2, 24.2.2.1.2, or 32.3.1.2.3.  
 In(N); repeater is transmitting rx\_code\_bit(s) as received from Port (N) except /J/K/ (see 24.3.4.2), or recoded rx\_symbol\_vector as received from Port (N) except SSD (see 32.3.5.2).  
 Pream; repeater is sourcing preamble pattern as defined by the PMA or PCS of the port type (see 23.2.1.2, 24.2.2.2, 32.3.1.2, Figure 23–6 and Figure 24–5.)  
 Data; repeater is transmitting data frame on Port X. This data represents the original MAC source data field, properly encoded for the PHY type (see 23.2.1.2, 24.2.2.2, and 32.3.1.2.3).  
 Jam; repeater is sourcing well formed arbitrary data encodings, excluding SFD, to the port PMA.  
 JamX; repeater is sourcing a pattern representing 010101... repetitively on Port X.  
 JamT4; repeater is sourcing the pattern +-+... repetitively on Port X.  
 JamT2; repeater is sending a pattern representing 010101... repetitively on Port X.  
 SFD; repeater is sourcing the Start Frame Delimiter on Port X encoded as defined by the appropriate PHY (see 23.2.3, Figure 24–5 and 32.3.1.2).  
 /J/K/; repeater is sourcing the code-groups /J/K/ as defined by the PMA on Port X (see 24.2.2.1.4).  
 /T/R/; repeater is sourcing the code-groups /T/R/ as defined by the PMA on Port X (see 24.2.2.1.5).  
 SSD; repeater is sourcing the code-groups SSD as defined by the PMA on Port X (see 32.4.2.5).  
 ESD; repeater is sourcing the code-groups ESD as defined by the PMA on Port X (see 32.3.1.2.3).  
 DF; repeater is sourcing the Data Frame of the packet on Port X. These are code elements originating on Port N exclusive of EOP1-5, SOSA and SOSB (see 23.2.3 and 23.2.4).  
 EOP; repeater is sourcing end of packet delimiter (EOP1-5) as defined by the appropriate PMA on Port X (see 23.2.1.2 and 23.2.4.1).  
 bad\_code; repeater is sourcing bad\_code as defined by the PMA of the transmit port (see 23.2.4.1).  
 tx\_err; repeater is sourcing a transmit error code element, either bad\_code (see 23.2.4.1) or the code-group /H/ (see 24.2.2.1), or (ESC/0, 0/ESC) (see Figure 32–12) as appropriate to the outgoing PHY type.

**partition(X)**

Flag from Partition state diagram for Port X, which determines whether a port receive path should be enabled or disabled.

Values: true; port has exceeded the consecutive collision limit.  
false; port has not exceeded the consecutive collision limit.

**part\_opt(X)**

Implementation option. Either value may be chosen for repeater implementation (see 27.3.1.6).

Values: true; port supports the recommended optional measures in the partition state machine.  
false; port does not support the recommended optional measures in the partition state machine.

**rxerror\_status(X)**

Signal received from PMA; indicates if Port X has detected an error condition from the PMA (see 23.3.7.1, Figure 24–14, and Figure 32–14). The repeater need not propagate this error condition during collision events.

Values: ERROR; the rxerror\_status parameter of the PMA\_RXERROR.indicate primitive for Port X is ERROR.  
NO\_ERROR; the rxerror\_status parameter of the PMA\_RXERROR.indicate primitive for Port X is NO\_ERROR.

**RX\_ER(X)**

Signal received from PCS; indicates if Port X has detected an error condition from the PCS (see 23.2.1.4, 24.2.3.2, 32.3.4.1, Figure 23–10, Figure 24–11, and Figure 32–13). The repeater need not propagate this error condition during collision events.

Values: true; the PCS RX\_ER signal for Port X is asserted.  
false; the PCS RX\_ER signal for Port X is negated.

**scarrier\_present(X)**

Signal received from PMA; indicates the status of sourced Carrier input at Port X.

Values: true; the carrier\_status parameter of the PMA\_CARRIER.indicate primitive for Port X is ON.  
false; the carrier\_status parameter of the PMA\_CARRIER.indicate primitive for Port X is OFF.

**source\_type(X)**

Signal received from PMA; indicates PMA type for Port X. The first port to assert activity maintains the source type status for all transmitting port(s) until activity is deasserted. Repeaters may optionally force nonequality on comparisons using this variable. It must then follow the behavior of the state diagrams accordingly and meet all the delay parameters as applicable for the real implemented port type(s).

Values: FCTX; the pma\_type parameter of the PMA\_TYPE.indicate primitive for Port X is X.  
T4; the pma\_type parameter of the PMA\_TYPE.indicate primitive for Port X is T4.  
T2; the pma\_type parameter of the PMA\_TYPE.indicate primitive for Port X is T2.

**27.3.2.1.3 Functions****command(X)**

A function that passes an inter-process flag to all ports specified by X.

Values: copy; indicates that the repeater core has summed the activity levels of its active ports and is in the ACTIVE state.  
collision; indicates that the repeater core has summed the activity levels of its active ports and is in the JAM state.

quiet; indicates that the repeater core has summed the activity levels of its active ports and is in the IDLE state.

port(Test)

A function that returns the designation of a port passing the test condition. For example, port(activity = scarrier\_present) returns the designation: X for a port for which scarrier\_present = true. If multiple ports meet the test condition, the Port function will be assigned one and only one of the acceptable values.

#### 27.3.2.1.4 Timers

All timers operate in the same fashion. A timer is reset and starts timing upon entering a state where “start x\_timer” is asserted. At time “x” after the timer has been started, “x\_timer\_done” is asserted and remains asserted until the timer is reset. At all other times, “x\_timer\_not\_done” is asserted.

When entering a state where “start x\_timer” is asserted, the timer is reset and restarted even if the entered state is the same as the exited state.

The timers used in the repeater state diagrams are defined as follows:

false\_carrier\_timer

Timer for length of false carrier (27.3.1.5.1) that must be present before the ISOLATION state is entered. The timer is done when it reaches 450 – 500 BT.

idle\_timer

Timer for length of time without carrier activity that must be present before the ISOLATION state is exited (27.3.1.5.1). The timer is done when it reaches 33 000 ± 25% BT.

ipg\_timer

Timer for length of time without carrier activity that must be present before carrier integrity tests (27.3.1.5.1) are re-enabled. The timer is done when it reaches 64 – 86 BT.

jabber\_timer

Timer for length of carrier that must be present before the Jabber state (27.3.1.7), and optionally during a collision the Partition state (27.3.1.6), is entered. The timer is done when it reaches 40 000 – 75 000 BT.

no\_collision\_timer

Timer for length of packet without collision before the Partition state is exited (27.3.1.6). The timer is done when it reaches 450 – 560 BT.

valid\_carrier\_timer

Timer for length of valid carrier that must be present before the Isolation state is exited (27.3.1.5.1). The timer is done when it reaches 450 – 500 BT.

#### 27.3.2.1.5 Counters

CC(X)

Consecutive port collision count for Port X. Partitioning occurs on a terminal count of CCLimit being reached.

Values: Non-negative integers up to a terminal count of CCLimit.

FCC(X)

False Carrier Counter for Port X. Isolation occurs on a terminal count of FCCLimit being reached.

Values: Non-negative integers up to a terminal count of FCCLimit.

### 27.3.2.1.6 Port designation

Ports are referred to by number. Port information is obtained by replacing the X in the desired function with the number of the port of interest. Ports are referred to in general as follows:

X

Generic port designator. When X is used in a state diagram, its value is local to that diagram and not global to the set of state diagrams.

N

Is defined by the Port function on exiting the IDLE or JAM states of Figure 27–2. It indicates a port that caused the exit from these states.

ALL

Indicates all repeater ports are to be considered. All ports shall meet test conditions in order for the test to pass.

ALLXN

Indicates all ports except N should be considered. All ports considered shall meet the test conditions in order for the test to pass.

ANY

Indicates all ports are to be considered. One or more ports shall meet the test conditions in order for the test to pass.

ANYXN

Indicates any port other than N meeting the test conditions shall cause the test to pass.



27.3.2.2 State diagrams

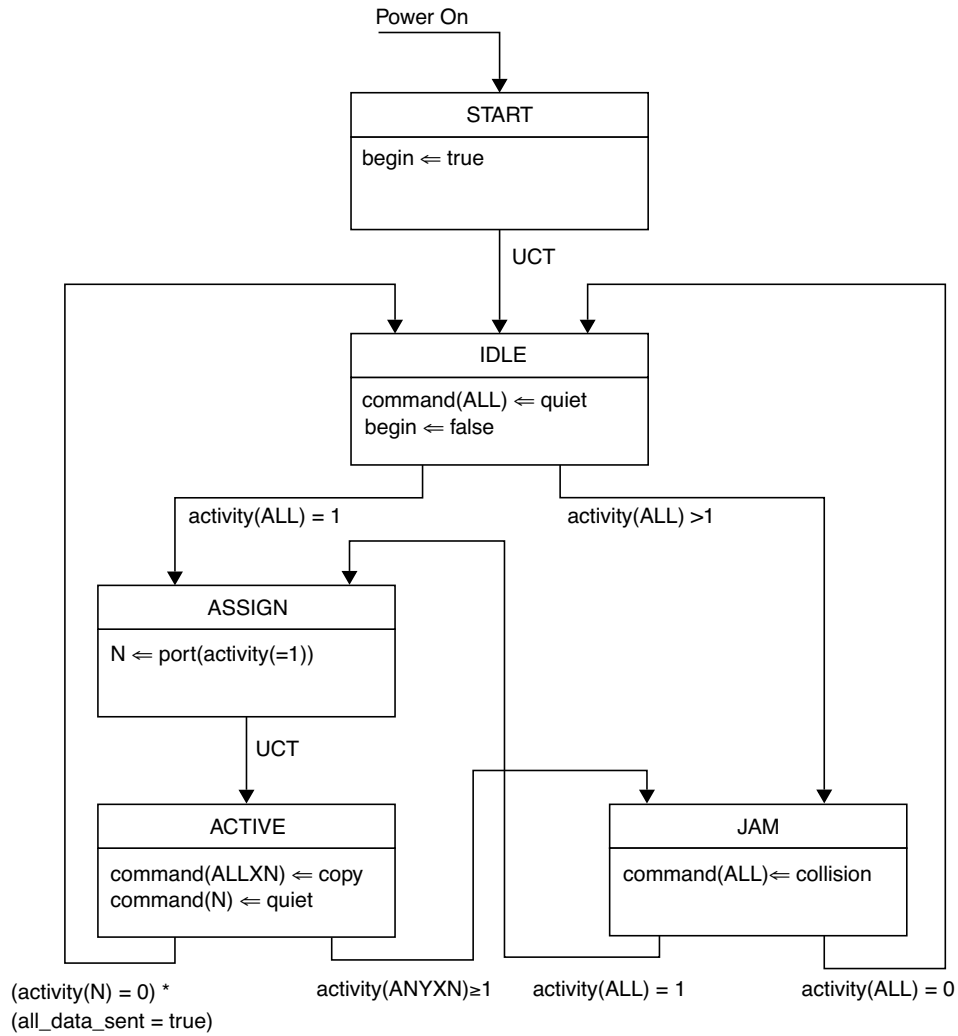


Figure 27-2—Repeater Core state diagram

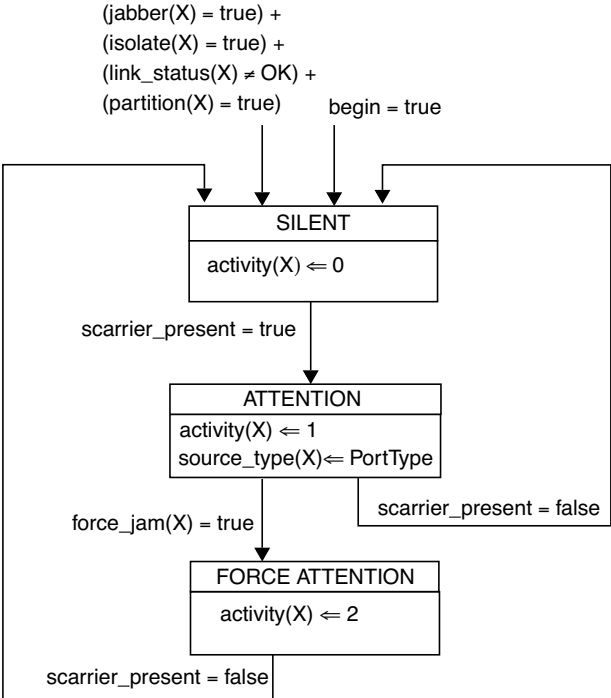


Figure 27-3—Receive state diagram for Port X

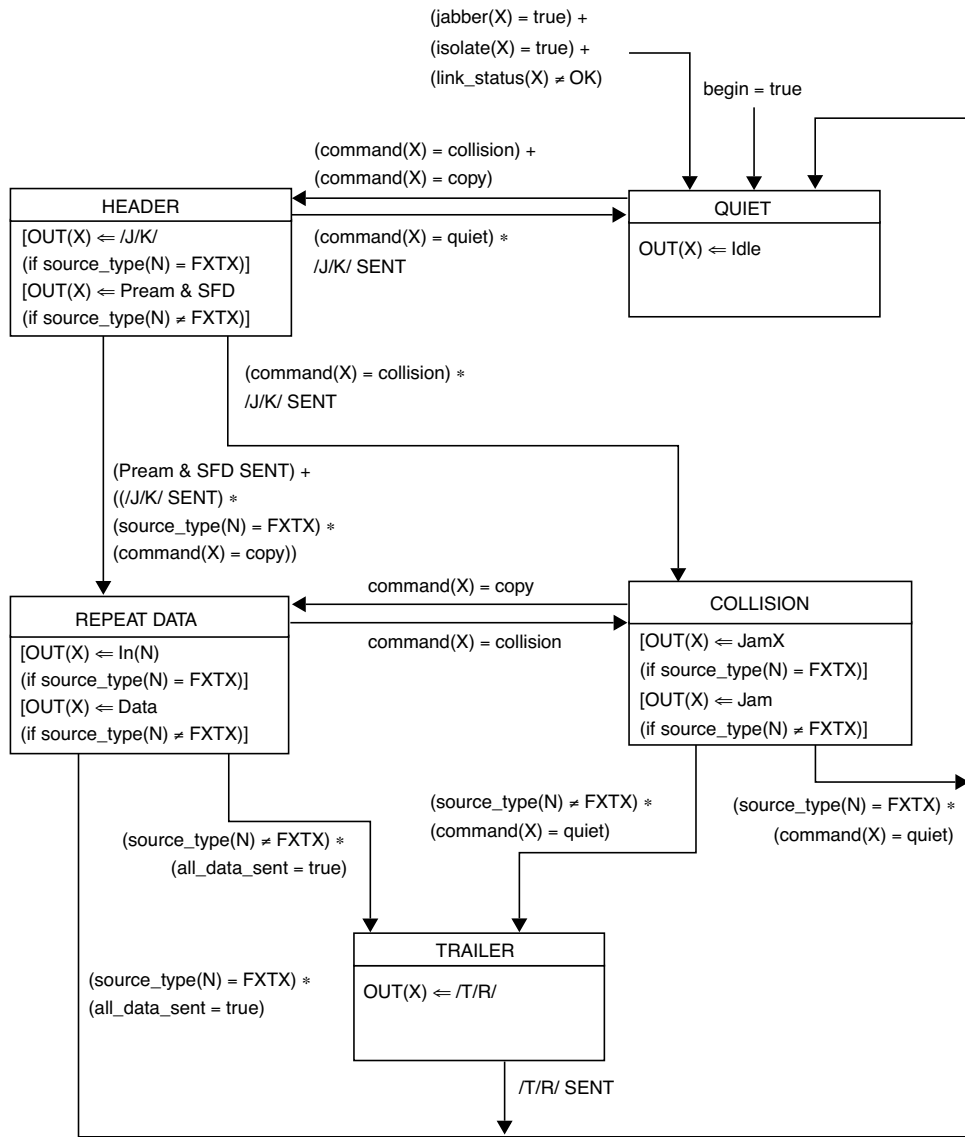


Figure 27-4—100BASE-TX and 100BASE-FX Transmit state diagram for Port X

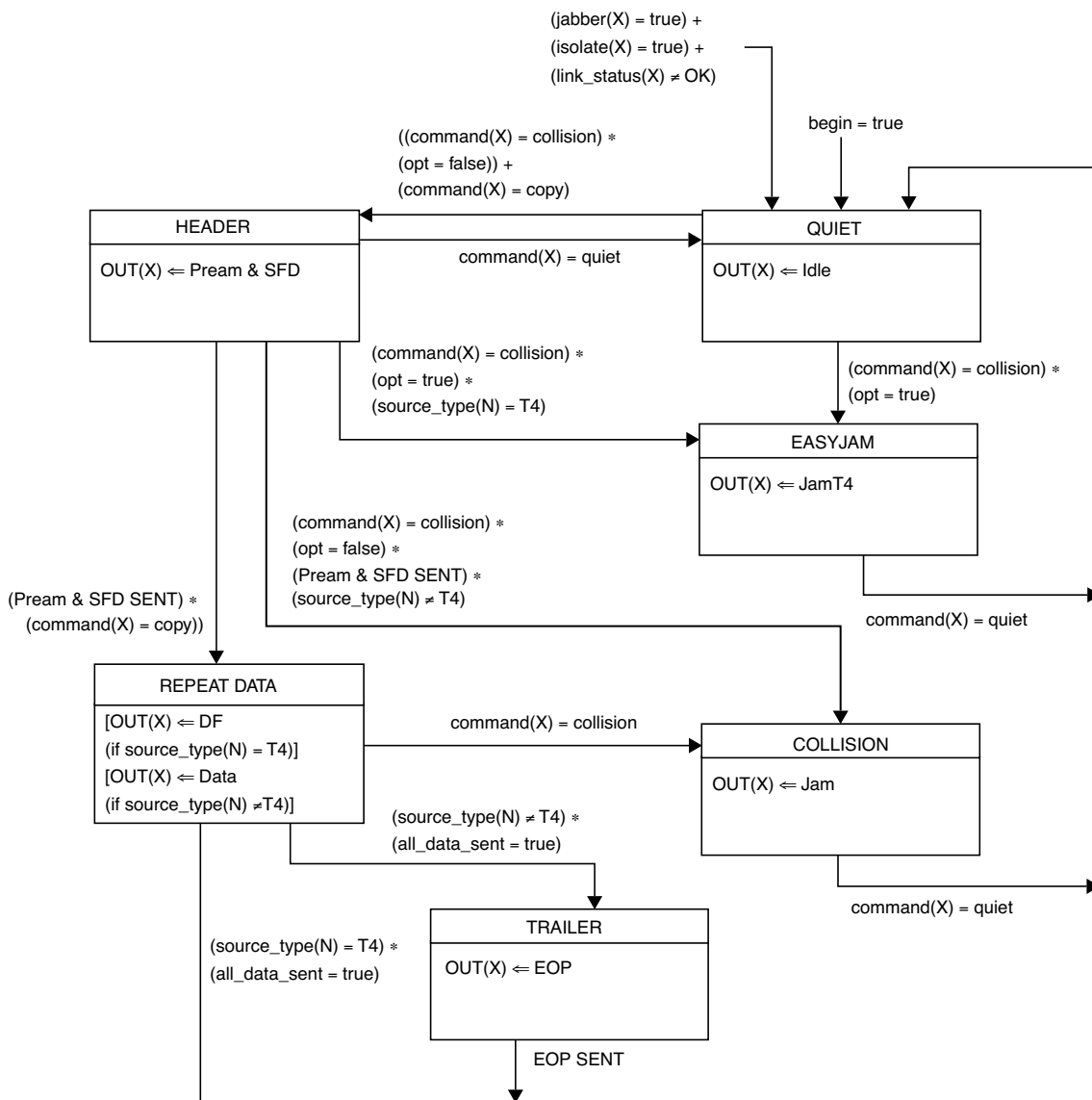


Figure 27–5—100BASE-T4 Transmit state diagram for Port X

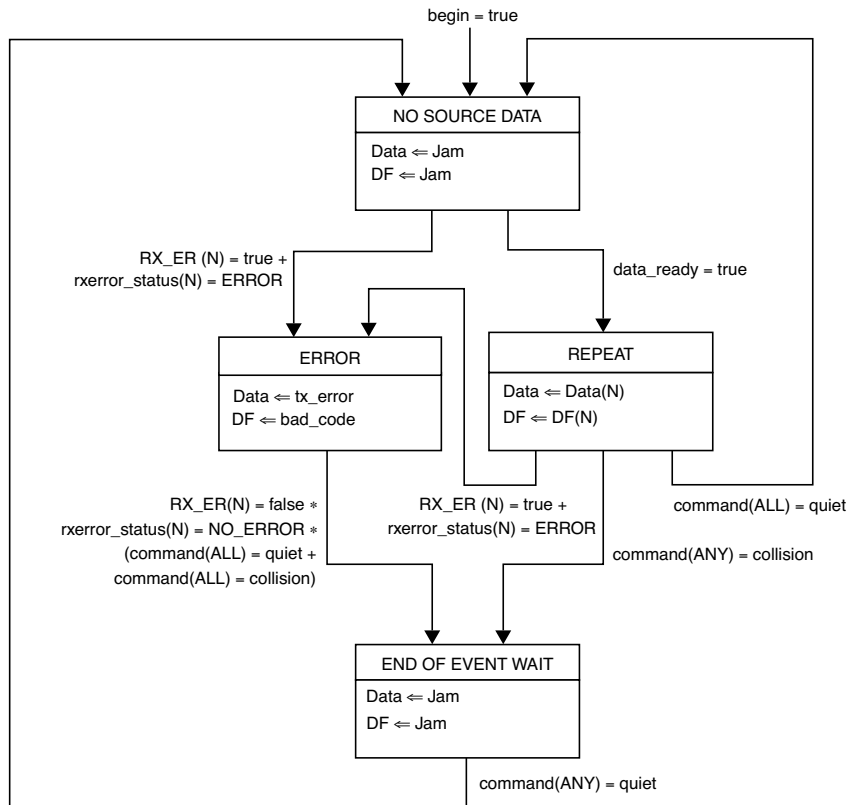


Figure 27-6—Repeater Data Handler state diagram

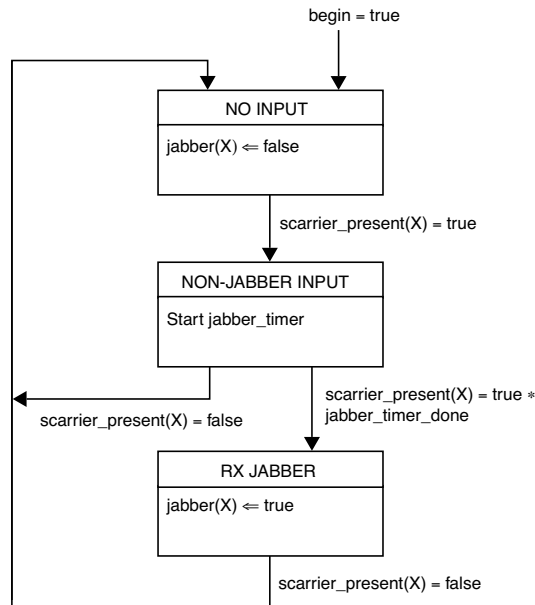


Figure 27-7—Receive Timer state diagram for Port X

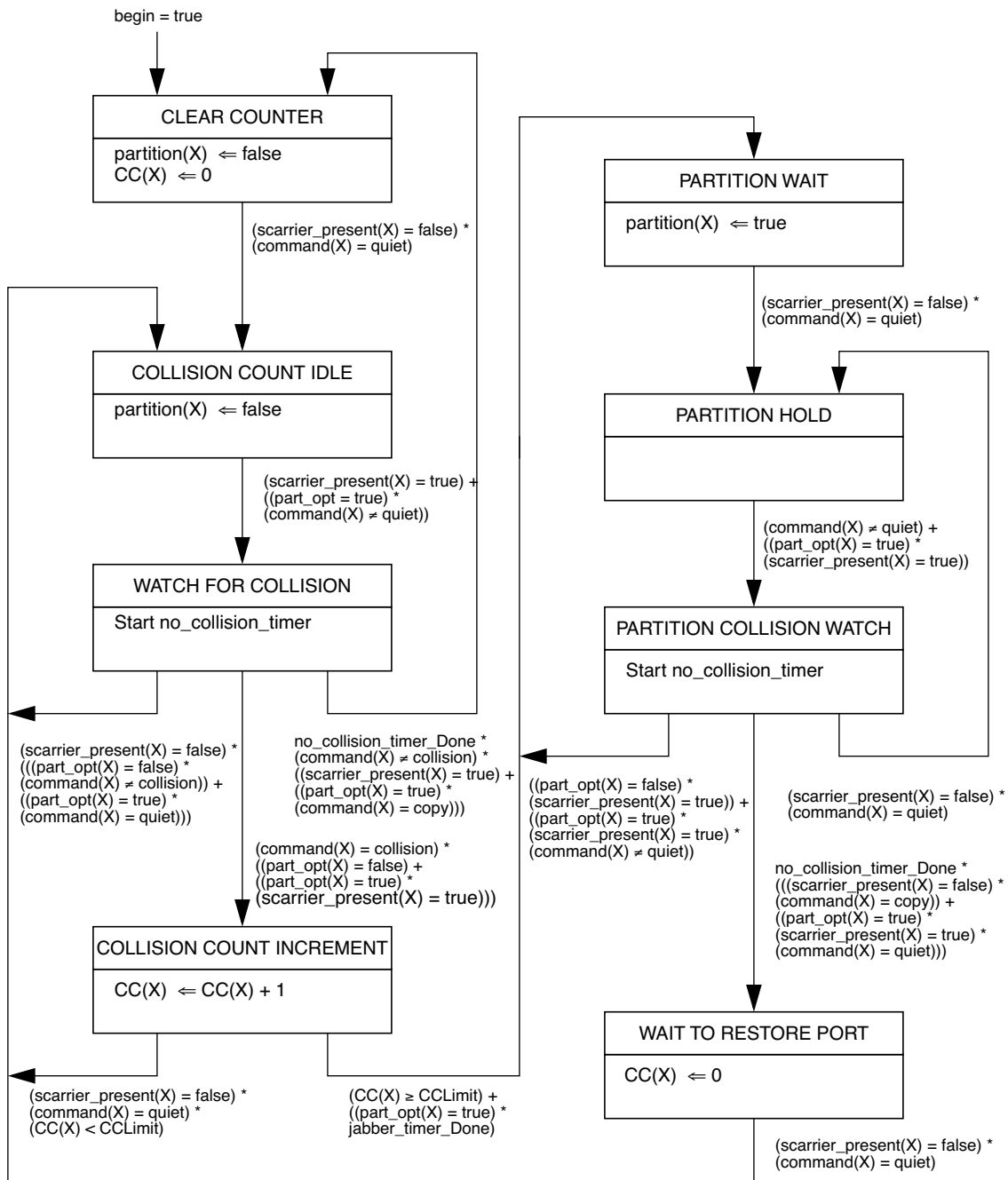


Figure 27–8— Partition state diagram for Port X

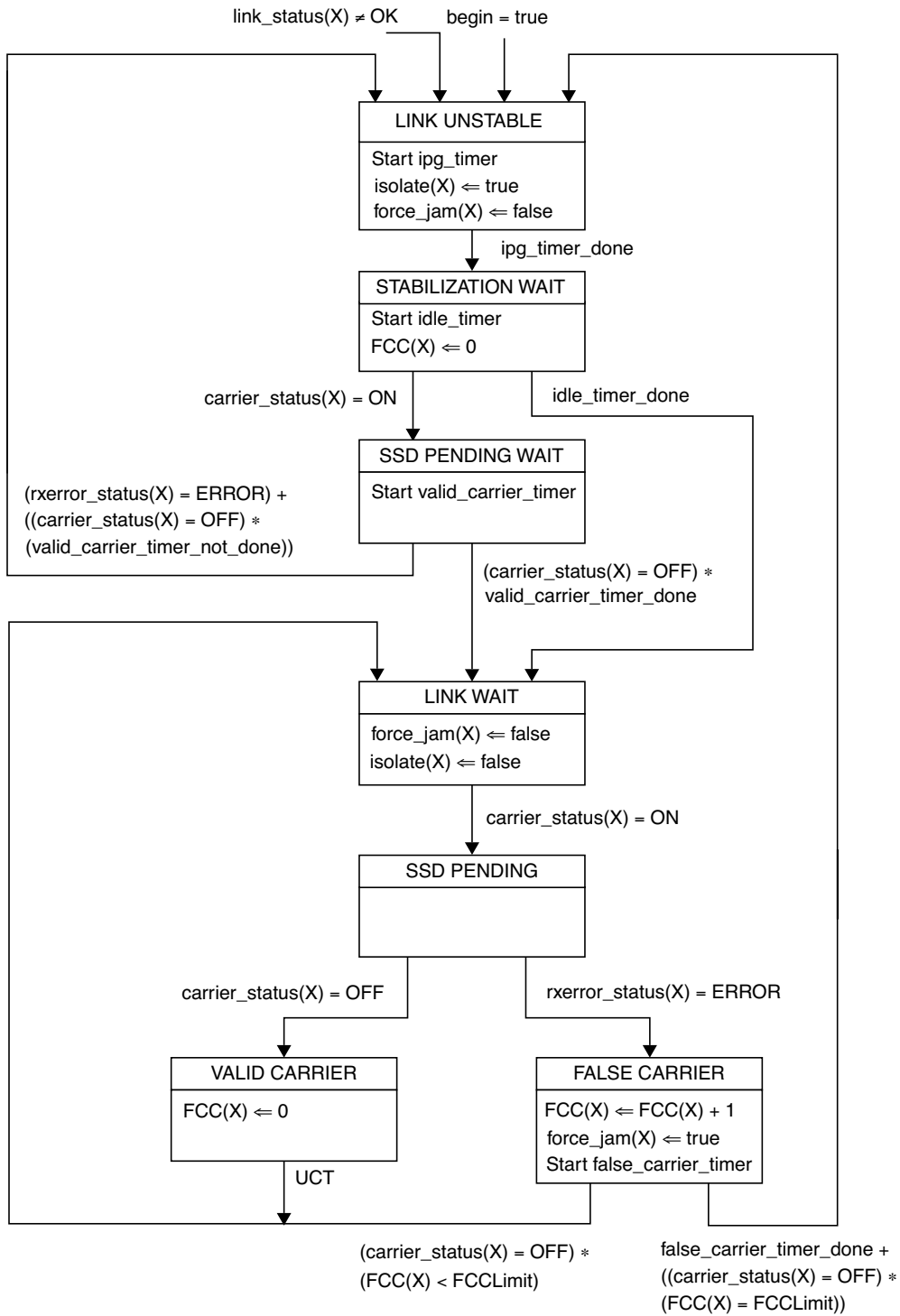


Figure 27-9—100BASE-X/T2 Carrier Integrity Monitor state diagram for Port X

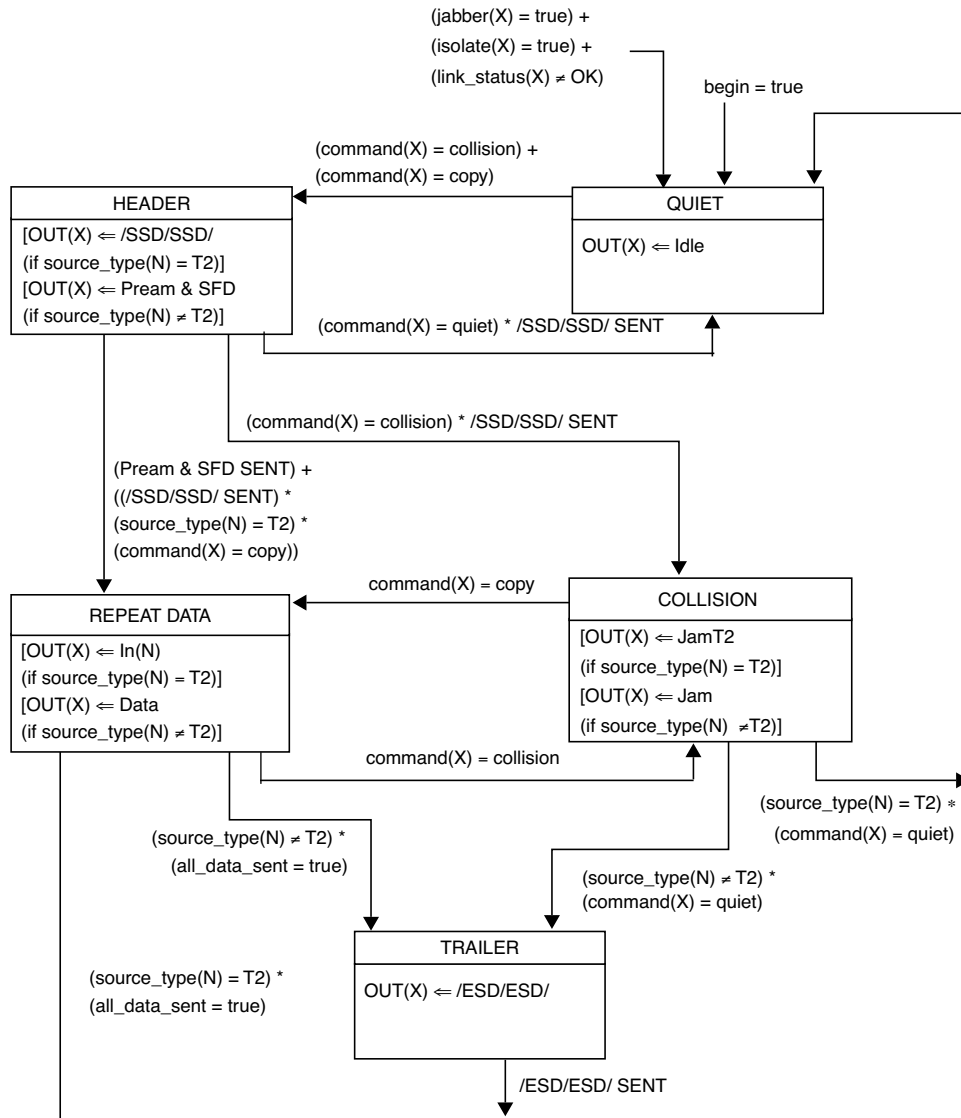


Figure 27-10—100BASE-T2 Transmit state diagram for Port X



## 27.4 Repeater electrical specifications

### 27.4.1 Electrical isolation

Network segments that have different isolation and grounding requirements shall have those requirements provided by the port-to-port isolation of the repeater set.

## 27.5 Environmental specifications

### 27.5.1 General safety

All equipment meeting this standard shall conform to IEC 60950: 1991.

### 27.5.2 Network safety

This subclause sets forth a number of recommendations and guidelines related to safety concerns; the list is neither complete nor does it address all possible safety issues. The designer is urged to consult the relevant local, national, and international safety regulations to ensure compliance with the appropriate requirements.

LAN cable systems described in this subclause are subject to at least four direct electrical safety hazards during their installation and use. These hazards are as follows:

- a) Direct contact between LAN components and power, lighting, or communications circuits.
- b) Static charge buildup on LAN cables and components.
- c) High-energy transients coupled onto the LAN cable system.
- d) Voltage potential differences between safety grounds to which the various LAN components are connected.

Such electrical safety hazards must be avoided or appropriately protected against for proper network installation and performance. In addition to provisions for proper handling of these conditions in an operational system, special measures must be taken to ensure that the intended safety features are not negated during installation of a new network or during modification or maintenance of an existing network. Isolation requirements are defined in 27.5.3.

#### 27.5.2.1 Installation

Sound installation practice, as defined by applicable local codes and regulations, shall be followed in every instance in which such practice is applicable.

#### 27.5.2.2 Grounding

The safety ground, or chassis ground for the repeater set, shall be provided through the main ac power cord via the third wire ground as defined by applicable local codes and regulations. It is recommended that an external PHY to the repeater should also be mechanically grounded to the repeater unit through the power and ground signals in the MII connection and via the metal shell and shield of the MII connector if available.

If the MDI connector should provide a shield connection, the shield may be connected to the repeater safety ground. A network segment connected to the repeater set through the MDI may use a shield. If both ends of the network segment have a shielded MDI connector available, then the shield may be grounded at both ends according to local regulations and ISO/IEC 11801: 1995, and as long as the ground potential difference between both ends of the network segment is less than 1 V rms. The same rules apply towards an inter-repeater link between two repeaters. Multiple repeaters should reside on the same power main; if not, then it is highly recommended that the repeaters be connected via fiber.

**WARNING**

It is assumed that the equipment to which the repeater is attached is properly grounded and not left floating nor serviced by a “doubly insulated ac power distribution system.” The use of floating or insulated equipment, and the consequent implications for safety, are beyond the scope of this standard.

**27.5.2.3 Installation and maintenance guidelines**

During installation and maintenance of the cable plant, care should be taken to ensure that uninsulated network cable connectors do not make electrical contact with unintended conductors or ground.

**27.5.3 Electrical isolation**

There are two electrical power distribution environments to be considered that require different electrical isolation properties:

- a) *Environment A.* When a LAN or LAN segment, with all its associated interconnected equipment, is entirely contained within a single low-voltage power distribution system and within a single building.
- b) *Environment B.* When a LAN crosses the boundary between separate power distribution systems or the boundary of a single building.

**27.5.3.1 Environment A requirements**

Attachment of network segments via repeater sets requires electrical isolation of 500 V rms, one-minute withstand, between the segment and the protective ground of the repeater unit.

**27.5.3.2 Environment B requirements**

The attachment of network segments that cross Environment B boundaries requires electrical isolation of 1500 V rms, one-minute withstand, between each segment and all other attached segments and also the protective ground of the repeater unit.

The requirements for interconnected electrically conducting LAN segments that are partially or fully external to a single building environment may require additional protection against lightning strike hazards. Such requirements are beyond the scope of this standard. It is recommended that the above situation be handled by the use of nonelectrically conducting segments (e.g., fiber optic).

It is assumed that any nonelectrically conducting segments will provide sufficient isolation within that media to satisfy the isolation requirements of Environment B.

**27.5.4 Reliability**

A two-port repeater set shall be designed to provide a mean time between failure (MTBF) of at least 50 000 hours of continuous operation without causing a communications failure among stations attached to the network medium. Repeater sets with more than two ports shall add no more than  $3.46 \times 10^{-6}$  failures per hour for each additional port.

The repeater set electronics should be designed to minimize the probability of component failures within the repeater electronics that prevent communications among other PHYs on the individual segments. Connectors and other passive components comprising the means of connecting the repeater to the cable should be designed to minimize the probability of total network failure.

## 27.5.5 Environment

### 27.5.5.1 Electromagnetic emission

The repeater shall comply with applicable local and national codes for the limitation of electromagnetic interference.

### 27.5.5.2 Temperature and humidity

The repeater is expected to operate over a reasonable range of environmental conditions related to temperature, humidity, and physical handling (such as shock and vibration). Specific requirements and values for these parameters are considered to be beyond the scope of this standard.

It is recommended that manufacturers indicate in the literature associated with the repeater the operating environmental conditions to facilitate selection, installation, and maintenance.

## 27.6 Repeater labeling

It is required that each repeater (and supporting documentation) shall be labeled in a manner visible to the user with these parameters:

- a) Crossover ports appropriate to the respective PHY should be marked with an X.
- b) The repeater set class type should be labeled in the following manner:
  - 1) Class I: a Roman numeral "I" centered within a circle.
  - 2) Class II: a Roman numeral "II" centered within a circle.

Additionally, it is recommended that each repeater (and supporting documentation) also be labeled in a manner visible to the user with at least these parameters:

- a) Data rate capability in Mb/s
- b) Any applicable safety warnings
- c) Port type, i.e., 100BASE-TX, 100BASE-T4, or 100BASE-T2
- d) Worst-case bit time delays between any two ports appropriate for
  - 1) Start-of-packet propagation delay
  - 2) Start-of-collision Jam propagation delay
  - 3) Cessation-of-collision Jam propagation delay

## 27.7 Protocol Implementation Conformance Statement (PICS) proforma for Clause 27, Repeater for 100 Mb/s baseband networks<sup>10</sup>

### 27.7.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 27, Repeater for 100 Mb/s baseband networks, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

### 27.7.2 Identification

#### 27.7.2.1 Implementation identification

Supplier	
Contact point for enquiries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s)	
NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.	
NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

#### 27.7.2.2 Protocol summary

Identification of protocol standard	IEEE Std 802.3-2002 <sup>®</sup> , Clause 27, Repeater for 100 Mb/s baseband networks
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required? No [ ] Yes [ ] (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3-2002 <sup>®</sup> .)	

Date of Statement	
-------------------	--

<sup>10</sup>Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

### 27.7.3 Major capabilities/options

Item	Feature	Subclause	Status	Support	Value/Comment
*FXP	Repeater supports 100BASE-FX connections	27.1.2.2	O		
*TXP	Repeater supports 100BASE-TX connections	27.1.2.2	O		
*T4P	Repeater supports 100BASE-T4 connections	27.1.2.2	O		
*T2P	Repeater supports 100BASE-T2 connections	27.1.2.2	O		
*CLI	Repeater meets Class I delays	27.1.1.3	O		
*CLII	Repeater meets Class II delays	27.1.1.3	O		
*PHYS	PHYs capable of detecting non 100BASE-T signals	27.3.1.5.2	O		
*OPF	Partition function supports the recommended optional measures as described	27.3.1.6	O		

In addition, the following predicate name is defined for use when different implementations from the set above have common parameters:

\*XP:FXP or TXP

### 27.7.4 PICS proforma tables for the repeater for 100 Mb/s baseband networks

#### 27.7.4.1 Compatibility considerations

Item	Feature	Subclause	Status	Support	Value/Comment
CC1	100BASE-FX port compatible at the MDI	27.1.2.2	FXP:M		
CC2	100BASE-TX port compatible at the MDI	27.1.2.2	TXP:M		
CC3	100BASE-T4 port compatible at the MDI	27.1.2.2	T4P:M		
CC4	Internal segment compatibility	27.1.2.2.1	M		Internal port meets Clause 30 when repeater management implemented
CC5	100BASE-T2 port compatible at the MDI	27.1.2.2	T2P:M		

**27.7.4.2 Repeater functions**

Item	Feature	Subclause	Status	Support	Value/Comment
RF1	Signal Restoration	27.3.1	M		
RF2	Data Handling	27.3.1	M		
RF3	Received Event Handling	27.3.1	M		
RF4	Collision Handling	27.3.1	M		
RF5	Error Handling	27.3.1	M		
RF6	Partition	27.3.1	M		
RF7	Received Jabber	27.3.1	M		

**27.7.4.3 Signal Restoration function**

Item	Feature	Subclause	Status	Support	Value/Comment
SR1	Output amplitude as required by 100BASE-FX	27.3.1.1.1	FXP:M		
SR2	Output amplitude as required by 100BASE-TX	27.3.1.1.1	TXP:M		
SR3	Output amplitude as required by 100BASE-T4	27.3.1.1.1	T4P:M		
SR4	Output signal wave-shape as required by 100BASE-FX	27.3.1.1.2	FXP:M		
SR5	Output signal wave-shape as required by 100BASE-TX	27.3.1.1.2	TXP:M		
SR6	Output signal wave-shape as required by 100BASE-T4	27.3.1.1.2	T4P:M		
SR7	Output data timing as required by 100BASE-FX	27.3.1.1.3	FXP:M		
SR8	Output data timing as required by 100BASE-TX	27.3.1.1.3	TXP:M		
SR9	Output data timing as required by 100BASE-T4	27.3.1.1.3	T4P:M		
SR10	Output amplitude as required by 100BASE-T2	27.3.1.1.1	T2P:M		
SR11	Output signal wave-shape as required by 100BASE-T2	27.3.1.1.2	T2P:M		
SR12	Output data timing as required by 100BASE-T2	27.3.1.1.3	T2P:M		

**27.7.4.4 Data Handling function**

Item	Feature	Subclause	Status	Support	Value/Comment
DH1	Data frames forwarded to all ports except receiving port	27.3.1.2.1	M		
DH2	Data frames transmitted as appropriate for 100BASE-FX	27.3.1.2.1	FXP:M		
DH3	Data frames transmitted as appropriate for 100BASE-TX	27.3.1.2.1	TXP:M		
DH4	Data frames transmitted as appropriate for 100BASE-T4	27.3.1.2.1	T4P:M		
DH5	Code Violations forwarded to all transmitting ports	27.3.1.2.2	M		
DH6	Code Violations forwarded as received	27.3.1.2.2	O.1		
DH7	Received Code Violation forwarded as /H/ or as received	27.3.1.2.2	XP:O.1		
DH8	Received Code Violation forwarded as bad_code or as received	27.3.1.2.2	T4P:O.1		
DH9	Code element substitution for remainder of packet after received Code Violation	27.3.1.2.2	M		
DH10	Data frames transmitted as appropriate for 100BASE-T2	27.3.1.2.1	T2P:M		

**27.7.4.5 Receive Event Handling function**

Item	Feature	Subclause	Status	Support	Value/Comment
RE1	scarrier_present detect implemented	27.3.1.3.1	M		
RE2	Repeat all received signals	27.3.1.3.1	M		
RE3	Preamble encoded as required by 100BASE-FX	27.3.1.3.2	FXP:M		
RE4	Preamble encoded as required by 100BASE-TX	27.3.1.3.2	TXP:M		
RE5	Preamble encoded as required by 100BASE-T4	27.3.1.3.2	T4P:M		
RE6	Start-of-packet propagation delay, Class I repeater	27.3.1.3.3	CLI:M		
RE7	Start-of-packet propagation delay, Class II repeater	27.3.1.3.3	CLII:M		
RE8	Start-of-packet variability for 100BASE-FX input port	27.3.1.3.4	FXP:M		7.0 BT

**27.7.4.5 Receive Event Handling function (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
RE8	Start-of-packet variability for 100BASE-TX input port	27.3.1.3.4	TXP:M		7.0 BT
RE9	Start-of-packet variability for 100BASE-T4 input port	27.3.1.3.4	T4P:M		8.0 BT
RE10	Preamble encoded as required by 100BASE-T2	27.3.1.3.2	T2P:M		
RE11	Start of packet variability for 100BASE-T2 input port	27.3.1.3.4	T2P:M		8.0 BT

**27.7.4.6 Collision Handling function**

Item	Feature	Subclause	Status	Support	Value/Comment
CO1	Collision Detection	27.3.1.4.1	M		Receive event on more than one port
CO2	Jam Generation	27.3.1.4.2	M		Transmit Jam message while collision is detected
CO3	Collision-jam propagation delay, Class I repeater	27.3.1.4.3	CLI:M		$SOP + SOJ \leq 140$ BT
CO4	Collision-jam propagation delay, Class II repeater with any port T4	27.3.1.4.3	CLII:M		$SOP + SOJ \leq 67$ BT
CO5	Collision-jam propagation delay, Class II repeater, all TX/FX ports	27.3.1.4.3	CLII:M		$SOP \leq 46, SOJ \leq 46$ BT
CO6	Cessation of collision propagation delay, Class I repeater	27.3.1.4.4	CLI:M		$EOJ \leq SOP$
CO7	Cessation of collision propagation delay, Class II repeater	27.3.1.4.4	CLII:M		$EOJ \leq SOP$
CO8	Collision-jam propagation delay, Class II repeater with any port T2	27.3.1.4.3	CLII * T2P:M		$SOP + SOJ \leq 90$ BT



**27.7.4.7 Error Handling function**

Item	Feature	Subclause	Status	Support	Value/Comment
EH1	Carrier Integrity function implementation	27.3.1.5.1	XP:M		Self-interrupt of data reception
EH2	False carrier count for Link Unstable detection	27.3.1.5.1	XP:M		False carrier count in excess of FCCLimit
EH3	False carrier count reset	27.3.1.5.1	XP:M		Count reset on valid carrier
EH4	False carrier timer for Link Unstable detection	27.3.1.5.1	XP:M		False carrier of length in excess of false_carrier_timer
EH5	Jam message duration	27.3.1.5.1	XP:M		Equals duration of false carrier event, but not greater than duration of false_carrier_timer
EH6	Link Unstable detection	27.3.1.5.1	XP:M		False Carrier count exceed FCCLimit or False carrier exceeds the false_carrier_timer or power-up reset
EH7	Messages sent to repeater unit in Link Unstable state	27.3.1.5.1	XP:M		Inhibited sending messages to repeater unit
EH8	Messages sent from repeater unit in Link Unstable state	27.3.1.5.1	XP:M		Inhibited sending output messages
EH9	Monitoring activity on PMA interface in Link Unstable state	27.3.1.5.1	XP:M		Continue monitoring activity at PMA interface
EH10	Reset of Link Unstable state	27.3.1.5.1	XP:M		No activity for more than ipg_timer plus idle_timer or Valid carrier event of duration greater than valid_carrier_timer preceded by Idle of duration greater than ipg_timer
EH11	Block flow of non-100 Mb/s signals	27.3.1.5.2	PHYS:M		

**27.7.4.8 Partition functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PA1	Partition function implementation	27.3.1.6	M		Self-interrupt of data reception
PA2	Collision count for entry into partition state	27.3.1.6	M		Collision count greater than or equal to CCLimit
PA3	Collision counter incrementing	27.3.1.6	M		Count incremented on a collision
PA4	Collision counter reset	27.3.1.6	M		Count reset on receive activity in excess of no_collision_timer without collision

**27.7.4.8 Partition functions (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
PA5			OPF:M		Count reset on transmission in excess of no_collision_timer without collision
PA6	Messages sent to repeater unit in Partition state	27.3.1.6	M		Inhibited sending messages to repeater unit
PA7	Messages sent from repeater unit in Partition state	27.3.1.6	M		Continue sending output messages
PA8	Monitoring activity on PMA interface in Partition state	27.3.1.6	M		Continue monitoring activity at PMA interface
PA9	Reset of Partition state	27.3.1.6	M		Power-up reset or transmission in excess of no_collision_timer without collision
PA10			OPF:M		Receive activity in excess of no_collision_timer without collision
PA11	Excessive carrier entry into Partition state	27.3.1.6	OPF:M		Carrier duration in excess of jabber_timer in which a collision occurs

**27.7.4.9 Receive Jabber function**

Item	Feature	Subclause	Status	Support	Value/Comment
RJ1	Receive Jabber function implementation	27.3.1.7	M		Self-interrupt of data reception
RJ2	Excessive receive duration timer for Receive Jabber detection	27.3.1.7	M		Reception duration in excess of jabber_timer
RJ3	Messages sent to repeater unit in Receive Jabber state	27.3.1.7	M		Inhibit sending input messages to repeater unit
RJ4	Messages sent from repeater unit in Receive Jabber state	27.3.1.7	M		Inhibit sending output messages
RJ5	Reset of Receive Jabber state	27.3.1.7	M		Power-up reset or Carrier no longer detected

**27.7.4.10 Repeater state diagrams**

Item	Feature	Subclause	Status	Support	Value/Comment
SD1	Repeater Core state diagram	27.3.2.2	M		Meets the requirements of Figure 27-2
SD2	Receive state diagram for Port X	27.3.2.2	M		Meets the requirements of Figure 27-3
SD3	100BASE-TX and 100BASE-FX Transmit state diagram for Port X	27.3.2.2	XP:M		Meets the requirements of Figure 27-4
SD4	100BASE-T4 Transmit state diagram for Port X	27.3.2.2	T4P:M		Meets the requirements of Figure 27-5
SD5	Repeater Data Handler state diagram	27.3.2.2	M		Meets the requirements of Figure 27-6
SD6	Receive Timer for Port X state diagram	27.3.2.2	M		Meets the requirements of Figure 27-7
SD7	Repeater Partition state diagram for Port X	27.3.2.2	M		Meets the requirements of Figure 27-8
SD8	Carrier Integrity Monitor for Port X state diagram	27.3.2.2	M		Meets the requirements of Figure 27-9
SD9	100BASE-T2 Transmit state diagram for Port X	27.3.2.2	T2P:M		Meets the requirements of Figure 27-10

**27.7.4.11 Repeater electrical**

Item	Feature	Subclause	Status	Support	Value/Comment
EL1	Port-to-port isolation	27.4.1	M		Satisfies isolation and grounding requirements for attached network segments
EL2	Safety	27.5.1	M		IEC 60950: 1991
EL3	Installation practices	27.5.2.1	M		Sound, as defined by local code and regulations
EL4	Grounding	27.5.2.2	M		Chassis ground provided through ac mains cord
EL5	2-port repeater set MTBF	27.5.4	M		At least 50 000 hours
EL6	Additional port effect on MTBF	27.5.4	M		No more than $3.46 \times 10^{-6}$ increase in failures per hour
EL7	Electromagnetic interference	27.5.5.1	M		Comply with local or national codes

**27.7.4.12 Repeater labeling**

Item	Feature	Subclause	Status	Support	Value/Comment
LB1	Crossover ports	27.6	M		Marked with an X
LB2	Class I repeater	27.6	CLI:M		Marked with a Roman numeral I centered within a circle
LB3	Class II repeater	27.6	CLII:M		Marked with Roman numerals II centered within a circle
LB4	Data Rate	27.6	O		100 Mb/s
LB5	Safety warnings	27.6	O		Any applicable
LB6	Port Types	27.6	O		100BASE-FX or 100BASE-TX or 100BASE-T4 or 100BASE-T2
LB7	Worse-case start-of-packet propagation delay	27.6	O		Value in Bit Times
LB8	Worse-case start-of-collision-Jam propagation delay	27.6	O		Value in Bit Times
LB9	Worse-case cessation-of-collision Jam propagation delay	27.6	O		Value in Bit Times

## 28. Physical Layer link signaling for 10 Mb/s, 100 Mb/s, and 1000 Mb/s Auto-Negotiation on twisted pair

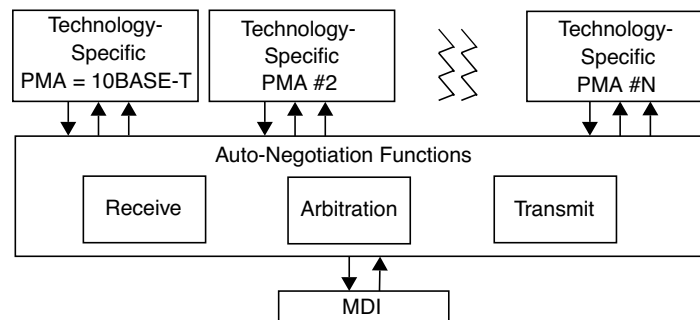
### 28.1 Overview

#### 28.1.1 Scope

Clause 28 describes the Auto-Negotiation function that allows a device to advertise enhanced modes of operation it possesses to a device at the remote end of a link segment and to detect corresponding enhanced operational modes that the other device may be advertising. The normative definitions for all extensions to Auto-Negotiation and all related register assignments for this standard are documented in Annex 28D.

The objective of the Auto-Negotiation function is to provide the means to exchange information between two devices that share a link segment and to automatically configure both devices to take maximum advantage of their abilities. Auto-Negotiation is performed using a modified 10BASE-T link integrity test pulse sequence, such that no packet or upper layer protocol overhead is added to the network devices (see Figure 28–1). Auto-Negotiation does not test the link segment characteristics (see 28.1.4).

The function allows the devices at both ends of a link segment to advertise abilities, acknowledge receipt and understanding of the common mode(s) of operation that both devices share, and to reject the use of operational modes that are not shared by both devices. Where more than one common mode exists between the two devices, a mechanism is provided to allow the devices to resolve to a single mode of operation using a predetermined priority resolution function. The Auto-Negotiation function allows the devices to switch between the various operational modes in an ordered fashion, permits management to disable or enable the Auto-Negotiation function, and allows management to select a specific operational mode. The Auto-Negotiation function also provides a Parallel Detection function to allow 10BASE-T, 100BASE-TX, and 100BASE-T4 compatible devices to be recognized, even though they may not provide Auto-Negotiation.



**Figure 28–1 — High-level model**

The basic mechanism to achieve Auto-Negotiation is to pass information encapsulated within a burst of closely spaced link integrity test pulses that individually meet the 10BASE-T Transmitter Waveform for Link Test Pulse (Figure 14–12). This burst of pulses is referred to as a Fast Link Pulse (FLP) Burst. Each device capable of Auto-Negotiation issues FLP Bursts at power up, on command from management, or due to user interaction. The FLP Burst consists of a series of link integrity test pulses that form an alternating clock/data sequence. Extraction of the data bits from the FLP Burst yields a Link Code Word that identifies the operational modes supported by the remote device, as well as some information used for the Auto-Negotiation function's handshake mechanism.

To maintain interoperability with existing 10BASE-T devices, the function also supports the reception of 10BASE-T compliant link integrity test pulses. 10BASE-T link pulse activity is referred to as the Normal Link Pulse (NLP) sequence and is defined in 14.2.1.1. A device that fails to respond to the FLP Burst sequence by returning only the NLP sequence is treated as a 10BASE-T compatible device.

### 28.1.2 Application perspective/objectives

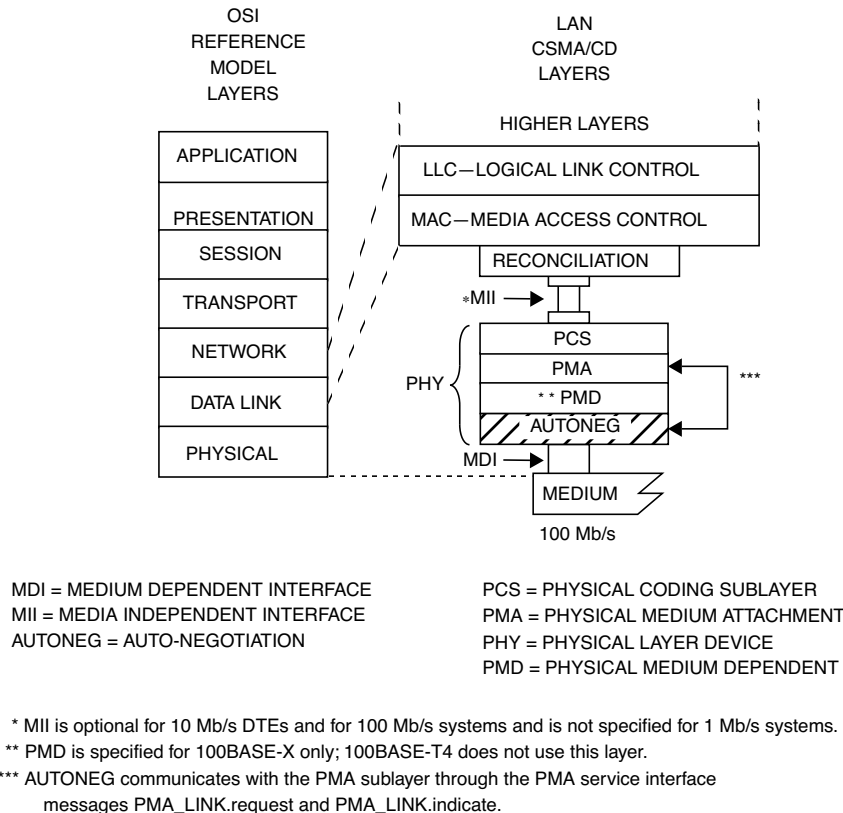
The Auto-Negotiation function is designed to be expandable and allow IEEE 802.3<sup>®</sup> compatible devices using an eight-pin modular connector to self-configure a jointly compatible operating mode. Implementation of the Auto-Negotiation function is optional. However, it is highly recommended that this method alone be utilized to perform the negotiation of the link operation.

The following are the objectives of Auto-Negotiation:

- a) Must interoperate with the IEEE 802.3<sup>®</sup> 10BASE-T installed base.
- b) Must allow automatic upgrade from the 10BASE-T mode to the desired “High-Performance Mode.”
- c) Requires that the 10BASE-T data service is the Lowest Common Denominator (LCD) that can be resolved. A 10BASE-T PMA is not required to be implemented, however. Only the NLP Receive Link Integrity Test function is required.
- d) Reasonable and cost-effective to implement.
- e) Must provide a sufficiently extensible code space to
  - 1) Meet existing and future requirements.
  - 2) Allow simple extension without impacting the installed base.
  - 3) Accommodate remote fault signals.
  - 4) Accommodate link partner ability detection.
- f) Must allow manual or Network Management configuration to override the Auto-Negotiation.
- g) Must be capable of operation in the absence of Network Management.
- h) Must not preclude the ability to negotiate “back” to the 10BASE-T operational mode.
- i) Must operate when
  - 1) The link is initially electrically connected.
  - 2) A device at either end of the link is powered up, reset, or a renegotiation request is made.
- j) The Auto-Negotiation function may be enabled by automatic, manual, or Network Management intervention.
- k) Completes the base page Auto-Negotiation function in a bounded time period.
- l) Will provide the basis for the link establishment process in future CSMA/CD compatible LAN standards that use an eight-pin modular connector.
- m) Must not cause corruption of IEEE 802.3<sup>®</sup> Layer Management statistics.
- n) Operates using a peer-to-peer exchange of information with no requirement for a master device (not master-slave).
- o) Must be robust in the UTP cable noise environment.
- p) Must not significantly impact EMI/RFI emissions.

### 28.1.3 Relationship to ISO/IEC 8802-3

The Auto-Negotiation function is provided at the Physical Layer of the OSI reference model as shown in Figure 28–2. Devices that support multiple modes of operation may advertise this fact using this function. The actual transfer of information of ability is observable only at the MDI or on the medium. Auto-Negotiation signaling does not occur across either the AUI or MII. Control of the Auto-Negotiation function may be supported through the Management Interface of the MII or equivalent. If an explicit embodiment of the MII is supported, the control and status registers to support the Auto-Negotiation function shall be implemented in accordance with the definitions in Clause 22 and 28.2.4. If a physical embodiment of the MII management is not present, then it is strongly recommended that the implementation provide control and status mechanisms equivalent to those described in Clause 22 and 28.2.4 for manual and/or management interaction.



**Figure 28–2—Location of Auto-Negotiation function within the ISO/IEC OSI reference model**

### 28.1.4 Compatibility considerations

The Auto-Negotiation function is designed to be completely backwards compatible and interoperable with 10BASE-T compliant devices. In order to achieve this, a device supporting the Auto-Negotiation function must provide the NLP Receive Link Integrity Test function as defined in Figure 28–17. The Auto-Negotiation function also supports connection to 100BASE-TX and 100BASE-T4 devices without Auto-Negotiation through the Parallel Detection function. Connection to technologies other than 10BASE-T, 100BASE-TX, or 100BASE-T4 that do not incorporate Auto-Negotiation is not supported.

Implementation of the Auto-Negotiation function is optional. For CSMA/CD compatible devices that use the eight-pin modular connector of ISO/IEC 8877: 1992 and that also encompass multiple operational modes, if a signaling method is used to automatically configure the preferred mode of operation, then the Auto-Negotiation function shall be used in compliance with Clause 28. If the device uses 10BASE-T compatible link signaling to advertise non-CSMA/CD abilities, the device shall implement the Auto-Negotiation function as administered by this specification. All future CSMA/CD implementations that use an eight-pin modular connector shall be interoperable with devices supporting Clause 28. If the implementor of a non-CSMA/CD eight-pin modular device wishes to assure that its operation does not conflict with CSMA/CD devices, then adherence to Clause 28 is recommended.

While this Auto-Negotiation function must be implemented in CSMA/CD compatible devices that utilize the eight-pin modular connector, encompass multiple operational modes, and offer an Auto-Negotiation mechanism, the use of this function does not mandate that the 10BASE-T packet data communication service must exist. A device that employs this function must support the 10BASE-T Link Integrity Test function through

the NLP Receive Link Integrity Test state diagram. The device may also need to support other technology-dependent link test functions depending on the modes supported. Auto-Negotiation does not perform cable tests, such as detect number of conductor pairs (if more than two pairs are required) or cable performance measurements. Some PHYs that explicitly require use of high-performance cables, may require knowledge of the cable type, or additional robustness tests (such as monitoring CRC or framing errors) to determine if the link segment is adequate.

#### **28.1.4.1 Interoperability with existing 10BASE-T devices**

During Auto-Negotiation, FLP Bursts separated by  $16 \pm 8$  ms are transmitted. The FLP Burst itself is a series of pulses separated by  $62.5 \pm 7 \mu\text{s}$ . The timing of FLP Bursts will cause a 10BASE-T device that is in the LINK TEST PASS state to remain in the LINK TEST PASS state while receiving FLP Bursts. An Auto-Negotiation able device must recognize the NLP sequence from a 10BASE-T Link Partner, cease transmission of FLP Bursts, and enable the 10BASE-T PMA, if present. If the NLP sequence is detected and if the Auto-Negotiation able device does not have a 10BASE-T PMA, it will cease transmission of FLP Bursts, forcing the 10BASE-T Link Partner into the LINK TEST FAIL state(s) as indicated in Figure 14–6.

NOTE—Auto-Negotiation does not support the transmission of the NLP sequence. The 10BASE-T PMA provides this function if it is connected to the MDI. In the case where an Auto-Negotiation able device without a 10BASE-T PMA is connected to a 10BASE-T device without Auto-Negotiation, the NLP sequence is not transmitted because the Auto-Negotiation function has no 10BASE-T PMA to enable that can transmit the NLP sequence.

#### **28.1.4.2 Interoperability with Auto-Negotiation compatible devices**

An Auto-Negotiation compatible device decodes the base Link Code Word from the FLP Burst, and examines the contents for the highest common ability that both devices share. Both devices acknowledge correct receipt of each other's base Link Code Words by responding with FLP Bursts containing the Acknowledge Bit set. After both devices complete acknowledgment, and optionally, Next Page exchange, both devices enable the highest common mode negotiated. The highest common mode is resolved using the priority resolution hierarchy specified in Annex 28B. It may subsequently be the responsibility of a technology-dependent link integrity test function to verify operation of the link prior to enabling the data service.

#### **28.1.4.3 Cabling compatibility with Auto-Negotiation**

Provision has been made within Auto-Negotiation to limit the resulting link configuration in situations where the cabling may not support the highest common capability of the two end points. The system administrator/installer must take the cabling capability into consideration when configuring a hub port's advertised capability. That is, the advertised capability of a hub port should not result in an operational mode that is not compatible with the cabling.

### **28.2 Functional specifications**

The Auto-Negotiation function provides a mechanism to control connection of a single MDI to a single PMA type, where more than one PMA type may exist. Management may provide additional control of Auto-Negotiation through the Management function, but the presence of a management agent is not required.

The Auto-Negotiation function shall provide the Auto-Negotiation Transmit, Receive, Arbitration, and NLP Receive Link Integrity Test functions and comply with the state diagrams of Figures 28–14 to 28–17. The Auto-Negotiation functions shall interact with the technology-dependent PMAs through the Technology-Dependent Interface. Technology-dependent PMAs include, but are not limited to, 100BASE-TX and 100BASE-T4. Technology-dependent link integrity test functions shall be implemented and interfaced to only if the device supports the given technology. For example, a 10BASE-T and 100BASE-TX Auto-Negotiation able device must implement and interface to the 100BASE-TX PMA/link integrity test function, but



does not need to include the 100BASE-T4 PMA/Link Integrity Test function. The Auto-Negotiation function shall provide an optional Management function that provides a control and status mechanism.

### 28.2.1 Transmit function requirements

The Transmit function provides the ability to transmit FLP Bursts. The first FLP Bursts exchanged by the Local Device and its Link Partner after Power-On, link restart, or renegotiation contain the base Link Code Word defined in 28.2.1.2. The Local Device may modify the Link Code Word to disable an ability it possesses, but will not transmit an ability it does not possess. This makes possible the distinction between local abilities and advertised abilities so that multimode devices may Auto-Negotiate to a mode lower in priority than the highest common local ability.

#### 28.2.1.1 Link pulse transmission

Auto-Negotiation's method of communication builds upon the link pulse mechanism employed by 10BASE-T MAUs to detect the status of the link. Compliant 10BASE-T MAUs transmit link integrity test pulses as a mechanism to determine if the link segment is operational in the absence of packet data. The 10BASE-T NLP sequence is a pulse (Figure 14–12) transmitted every  $16 \pm 8$  ms while the data transmitter is idle.

Auto-Negotiation substitutes the FLP Burst in place of the single 10BASE-T link integrity test pulse within the NLP sequence (Figure 28–3). The FLP Burst encodes the data that is used to control the Auto-Negotiation function. FLP Bursts shall not be transmitted when Auto-Negotiation is complete and the highest common denominator PMA has been enabled.

FLP Bursts were designed to allow use beyond initial link Auto-Negotiation, such as for a link monitor type function. However, use of FLP Bursts beyond the current definition for link startup shall be prohibited. Definition of the use of FLP Bursts while in the FLP LINK GOOD state is reserved.

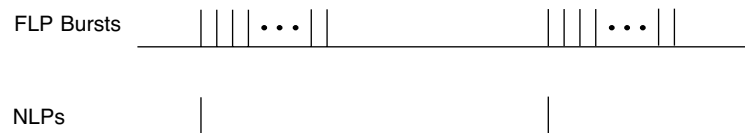


Figure 28–3—FLP Burst sequence to NLP sequence mapping

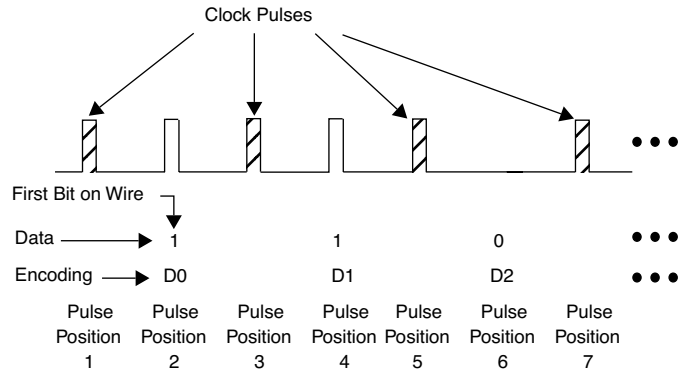
#### 28.2.1.1.1 FLP burst encoding

FLP Bursts shall be composed of link pulses meeting the requirements of Figure 14–12. A Fast Link Pulse Burst consists of 33 pulse positions. The 17 odd-numbered pulse positions shall contain a link pulse and represent clock information. The 16 even-numbered pulse positions shall represent data information as follows: a link pulse present in an even-numbered pulse position represents a logic one, and a link pulse absent from an even-numbered pulse position represents a logic zero. Clock pulses are differentiated from data pulses by the spacing between pulses as shown in Figure 28–5 and enumerated in Table 28–1.

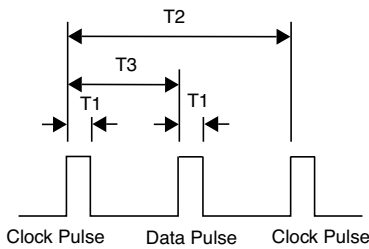
The encoding of data using pulses in an FLP Burst is illustrated in Figure 28–4.

#### 28.2.1.1.2 Transmit timing

The first pulse in an FLP Burst shall be defined as a clock pulse. Clock pulses within an FLP Burst shall be spaced at  $125 \pm 14 \mu\text{s}$ . If the data bit representation of logic one is to be transmitted, a pulse shall occur  $62.5 \pm 7 \mu\text{s}$  after the preceding clock pulse. If a data bit representing logic zero is to be transmitted, there shall be no link integrity test pulses within  $111 \mu\text{s}$  of the preceding clock pulse.

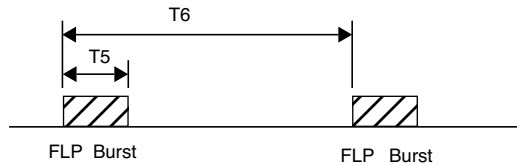


**Figure 28-4—Data bit encoding within FLP Bursts**



**Figure 28-5—FLP Burst pulse-to-pulse timing**

The first link pulse in consecutive FLP Bursts shall occur at a  $16 \pm 8$  ms interval (Figure 28-6).



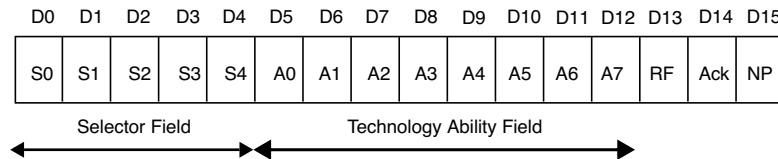
**Figure 28-6—FLP Burst to FLP Burst timing**

**Table 28-1—FLP Burst timing summary**

#	Parameter	Min.	Typ.	Max.	Units
T1	Clock/Data Pulse Width (Figure 14-12)		100		ns
T2	Clock Pulse to Clock Pulse	111	125	139	μs
T3	Clock Pulse to Data Pulse (Data = 1)	55.5	62.5	69.5	μs
T4	Pulses in a Burst	17		33	#
T5	Burst Width		2		ms
T6	FLP Burst to FLP Burst	8	16	24	ms

### 28.2.1.2 Link Code Word encoding

The base Link Code Word (base page) transmitted within an FLP Burst shall convey the encoding shown in Figure 28–7. The Auto-Negotiation function may support additional pages using the Next Page function. Encodings for the Link Code Word(s) used in Next Page exchange are defined in 28.2.3.4. In an FLP Burst, D0 shall be the first bit transmitted.



**Figure 28–7—Base page encoding**

#### 28.2.1.2.1 Selector Field

Selector Field (S[4:0]) is a five bit wide field, encoding 32 possible messages. Selector Field encoding definitions are shown in Annex 28A. Combinations not specified are reserved for future use. Reserved combinations of the Selector Field shall not be transmitted.

#### 28.2.1.2.2 Technology Ability Field

Technology Ability Field (A[7:0]) is an eight bit wide field containing information indicating supported technologies specific to the selector field value. These bits are mapped to individual technologies such that abilities are advertised in parallel for a single selector field value. The Technology Ability Field encoding for the IEEE 802.3<sup>®</sup> selector is described in Annex 28B.2 and in Annex 28D. Multiple technologies may be advertised in the Link Code Word. A device shall support the data service ability for a technology it advertises. It is the responsibility of the Arbitration function to determine the common mode of operation shared by a Link Partner and to resolve multiple common modes.

NOTE—While devices using a Selector Field value other than the IEEE 802.3 Selector Field value are free to define the Technology Ability Field bits, it is recommended that the 10BASE-T bit be encoded in the same bit position as in the IEEE 802.3<sup>®</sup> selector. A common bit position can be important if the technology using the other selector will ever coexist on a device that also offers a 10BASE-T mode.

#### 28.2.1.2.3 Remote Fault

Remote Fault (RF) is encoded in bit D13 of the base Link Code Word. The default value is logic zero. The Remote Fault bit provides a standard transport mechanism for the transmission of simple fault information. When the RF bit in the Auto-Negotiation advertisement register (Register 4) is set to logic one, the RF bit in the transmitted base Link Code Word is set to logic one. When the RF bit in the received base Link Code Word is set to logic one, the Remote Fault bit in the MII status register (Register 1) will be set to logic one, if the MII management function is present.

The Remote Fault bit shall be used in accordance with the Remote Fault function specifications (28.2.3.5).

#### 28.2.1.2.4 Acknowledge

Acknowledge (Ack) is used by the Auto-Negotiation function to indicate that a device has successfully received its Link Partner's Link Code Word. The Acknowledge Bit is encoded in bit D14 regardless of the value of the Selector Field or Link Code Word encoding. If no Next Page information is to be sent, this bit

shall be set to logic one in the Link Code Word after the reception of at least three consecutive and consistent FLP Bursts (ignoring the Acknowledge bit value). If Next Page information is to be sent, this bit shall be set to logic one after the device has successfully received at least three consecutive and matching FLP Bursts (ignoring the Acknowledge bit value), and will remain set until the Next Page information has been loaded into the Auto-Negotiation Next Page transmit register (Register 7). In order to save the current received Link Code Word, it must be read from the Auto-Negotiation link partner ability register (Register 6) before the Next Page of transmit information is loaded into the Auto-Negotiation Next Page transmit register. After the COMPLETE ACKNOWLEDGE state has been entered, the Link Code Word shall be transmitted six to eight (inclusive) times.

### 28.2.1.2.5 Next Page

Next Page (NP) is encoded in bit D15 regardless of the Selector Field value or Link Code Word encoding. Support for transmission and reception of additional Link Code Word encodings is optional. If Next Page ability is not supported, the NP bit shall always be set to logic zero. If a device implements Next Page ability and wishes to engage in Next Page exchange, it shall set the NP bit to logic one. A device may implement Next Page ability and choose not to engage in Next Page exchange by setting the NP bit to a logic zero. The Next Page function is defined in 28.2.3.4.

### 28.2.1.3 Transmit Switch function

The Transmit Switch function shall enable the transmit path from a single technology-dependent PMA to the MDI once a highest common denominator choice has been made and Auto-Negotiation has completed.

During Auto-Negotiation, the Transmit Switch function shall connect only the FLP Burst generator controlled by the Transmit State Diagram, Figure 28–14, to the MDI.

When a PMA is connected to the MDI through the Transmit Switch function, the signals at the MDI shall conform to all of the PHY's specifications.

## 28.2.2 Receive function requirements

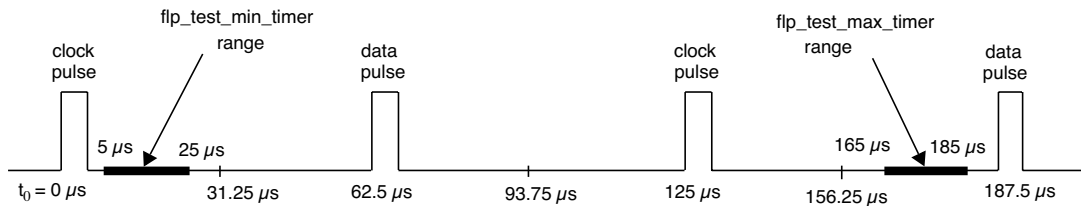
The Receive function detects the NLP sequence using the NLP Receive Link Integrity Test function of Figure 28–17. The NLP Receive Link Integrity Test function will not detect link pass based on carrier sense.

The Receive function detects the FLP Burst sequence, decodes the information contained within, and stores the data in `rx_link_code_word[16:1]`. The Receive function incorporates a receive switch to control connection to the 100BASE-TX or 100BASE-T4 PMAs in addition to the NLP Receive Link Integrity Test function, excluding the 10BASE-T Link Integrity Test function present in a 10BASE-T PMA. If Auto-Negotiation detects `link_status=READY` from any of the technology-dependent PMAs prior to FLP Burst detection, the `autoneg_wait_timer` (28.3.2) is started. If any other technology-dependent PMA indicates `link_status=READY` when the `autoneg_wait_timer` expires, Auto-Negotiation will not allow any data service to be enabled and may signal this as a remote fault to the Link Partner using the base page and will flag this in the Local Device by setting the Parallel Detection Fault bit (6.4) in the Auto-Negotiation expansion register. If a 10BASE-T PMA exists above the Auto-Negotiation function, it is not permitted to receive MDI activity in parallel with the NLP Receive Link Integrity Test function or any other technology-dependent function.

### 28.2.2.1 FLP Burst ability detection and decoding

In Figures 28–8 to 28–10, the symbol “ $t_0=0$ ” indicates the event that caused the timers described to start, and all subsequent times given are referenced from that point. All timers referenced shall expire within the range specified in Table 28–9 in 28.3.2.

The Receive function shall identify the Link Partner as Auto-Negotiation able if it receives 6 to 17 (inclusive) consecutive link pulses that are separated by at least `flp_test_min_timer` time (5–25  $\mu\text{s}$ ) but less than `flp_test_max_timer` time (165–185  $\mu\text{s}$ ) as shown in Figure 28–8. The information contained in the FLP Burst that identifies the Link Partner as Auto-Negotiation able shall not be passed to the Arbitration function if the FLP Burst is not complete. The Receive function may use the FLP Burst that identifies the Link Partner as Auto-Negotiation able for ability matching if the FLP Burst is complete. However, it is not required to use this FLP Burst for any purpose other than identification of the Link Partner as Auto-Negotiation able. Implementations may ignore multiple FLP Bursts before identifying the Link Partner as Auto-Negotiation able to allow for potential receive equalization time.

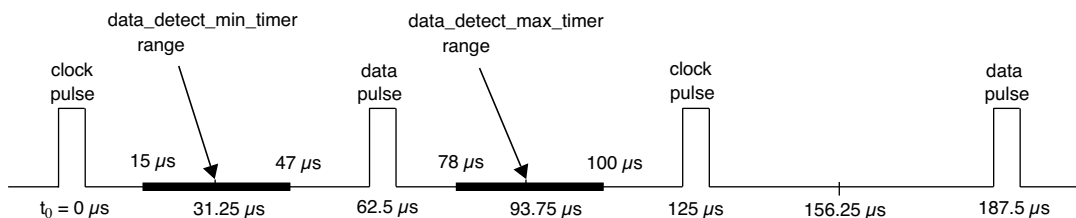


**Figure 28–8—FLP detect timers (`flp_test_min/max_timers`)**

The Receive function captures and decodes link pulses received in FLP Bursts. The first link pulse in an FLP Burst shall be interpreted as a clock link pulse. Detection of a clock link pulse shall restart the `data_detect_min_timer` and `data_detect_max_timer`. The `data_detect_min/max_timers` enable the receiver to distinguish data pulses from clock pulses and logic one data from logic zero data, as follows:

- a) If, during an FLP Burst, a link pulse is received when the `data_detect_min_timer` has expired while the `data_detect_max_timer` has not expired, the data bit shall be interpreted as a logic one (Figure 28–9).
- b) If, during an FLP Burst, a link pulse is received after the `data_detect_max_timer` has expired, the data bit shall be interpreted as a logic zero (Figure 28–9) and that link pulse shall be interpreted as a clock link pulse.

As each data bit is identified it is stored in the appropriate `rx_link_code_word[16:1]` element.

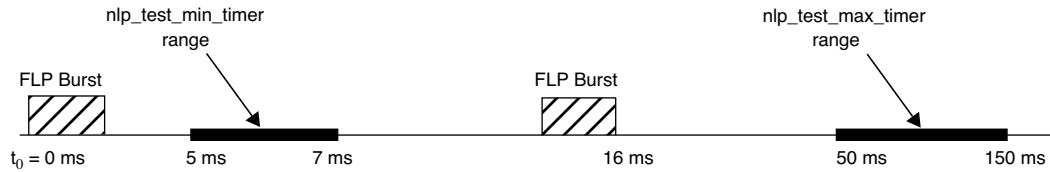


**Figure 28–9—FLP data detect timers (`data_detect_min/max_timers`)**

FLP Bursts conforming to the `nlp_test_min_timer` and `nlp_test_max_timer` timing as shown in Figure 28–10 shall be considered to have valid separation.

### 28.2.2.2 NLP detection

NLP detection is accomplished via the NLP Receive Link Integrity Test function in Figure 28–17. The NLP Receive Link Integrity Test function is a modification of the original 10BASE-T Link Integrity Test function (Figure 14–6), where the detection of receive activity will not cause a transition to the LINK TEST



NOTE—The reference for the starting of the `nlp_test_min_timer` is from the beginning of the FLP Burst, as shown by  $t_0$ , while the reference for the starting of the `nlp_test_max_timer` is from the expiration of the `nlp_test_min_timer`.

**Figure 28-10—FLP Burst timer (`nlp_test_min/max_timers`)**

PASS state during Auto-Negotiation. The NLP Receive Link Integrity Test function also incorporates the Technology-Dependent Interface requirements.

### 28.2.2.3 Receive Switch function

The Receive Switch function shall enable the receive path from the MDI to a single technology-dependent PMA once a highest common denominator choice has been made and Auto-Negotiation has completed.

During Auto-Negotiation, the Receive Switch function shall connect both the FLP Burst receiver controlled by the Receive state diagram, Figure 28-15, and the NLP Receive Link Integrity Test state diagram, Figure 28-17, to the MDI. During Auto-Negotiation, the Receive Switch function shall also connect the 100BASE-TX and 100BASE-T4 PMA receivers to the MDI if the 100BASE-TX and/or 100BASE-T4 PMAs are present.

When a PMA is connected to the MDI through the Receive Switch function, the signals at the PMA shall conform to all of the PHY's specifications.

### 28.2.2.4 Link Code Word matching

The Receive function shall generate `ability_match`, `acknowledge_match`, and `consistency_match` variables as defined in 28.3.1.

## 28.2.3 Arbitration function requirements

The Arbitration function ensures proper sequencing of the Auto-Negotiation function using the Transmit function and Receive function. The Arbitration function enables the Transmit function to advertise and acknowledge abilities. Upon indication of acknowledgment, the Arbitration function determines the highest common denominator using the priority resolution function and enables the appropriate technology-dependent PMA via the Technology-Dependent Interface (28.2.6).

### 28.2.3.1 Parallel detection function

The Local Device detects a Link Partner that supports Auto-Negotiation by FLP Burst detection. The Parallel Detection function allows detection of Link Partners that support 100BASE-TX, 100BASE-T4, and/or 10BASE-T, but do not support Auto-Negotiation. Prior to detection of FLP Bursts, the Receive Switch shall direct MDI receive activity to the NLP Receive Link Integrity Test state diagram, 100BASE-TX and 100BASE-T4 PMAs, if present, but shall not direct MDI receive activity to the 10BASE-T or any other PMA. If at least one of the 100BASE-TX, 100BASE-T4, or NLP Receive Link Integrity Test functions establishes `link_status=READY`, the LINK STATUS CHECK state is entered and the `autoneg_wait_timer` is started. If exactly one `link_status=READY` indication is present when the `autoneg_wait_timer` expires, then Auto-Negotiation shall set `link_control=ENABLE` for the PMA indicating `link_status=READY`. If a PMA is enabled, the Arbitration function shall set `link_control=DISABLE` to all other PMAs and indicate that

Auto-Negotiation has completed. On transition to the FLP LINK GOOD CHECK state from the LINK STATUS CHECK state the Parallel Detection function shall set the bit in the link partner ability register (Register 5) corresponding to the technology detected by the Parallel Detection function.

NOTE 1—Native 10BASE-T devices will be detected by the NLP Receive Link Integrity Test function, an integrated part of the Auto-Negotiation function. Hence, Parallel Detection for the 10BASE-T PMA is not required or allowed.

NOTE 2—When selecting the highest common denominator through the Parallel Detection function, only the half-duplex mode corresponding to the selected PMA may automatically be detected.

### 28.2.3.2 Renegotiation function

A renegotiation request from any entity, such as a management agent, shall cause the Arbitration function to disable all technology-dependent PMAs and halt any transmit data and link pulse activity until the break\_link\_timer expires (28.3.2). Consequently, the Link Partner will go into link fail and normal Auto-Negotiation resumes. The Local Device shall resume Auto-Negotiation after the break\_link\_timer has expired by issuing FLP Bursts with the base page valid in tx\_link\_code\_word[16:1].

Once Auto-Negotiation has completed, renegotiation will take place if the Highest Common Denominator technology that receives link\_control=ENABLE returns link\_status=FAIL. To allow the PMA an opportunity to determine link integrity using its own link integrity test function, the link\_fail\_inhibit\_timer qualifies the link\_status=FAIL indication such that renegotiation takes place if the link\_fail\_inhibit\_timer has expired and the PMA still indicates link\_status=FAIL or link\_status=READY.

### 28.2.3.3 Priority Resolution function

Since a Local Device and a Link Partner may have multiple common abilities, a mechanism to resolve which mode to configure is required. The mechanism used by Auto-Negotiation is a Priority Resolution function that predefines the hierarchy of supported technologies. The single PMA enabled to connect to the MDI by Auto-Negotiation shall be the technology corresponding to the bit in the Technology Ability Field common to the Local Device and Link Partner that has the highest priority as defined in Annex 28B. This technology is referred to as the Highest Common Denominator, or HCD, technology. If the Local Device receives a Technology Ability Field with a bit set that is reserved, the Local Device shall ignore that bit for priority resolution. Determination of the HCD technology occurs on entrance to the FLP LINK GOOD CHECK state. In the event that a technology is chosen through the Parallel Detection function, that technology shall be considered the highest common denominator (HCD) technology. In the event that there is no common technology, HCD shall have a value of “NULL,” indicating that no PMA receives link\_control=ENABLE, and link\_status\_[HCD]=FAIL.

### 28.2.3.4 Next Page function

The Next Page function uses the standard Auto-Negotiation arbitration mechanisms to allow exchange of arbitrary pieces of data. Data is carried by optional Next Pages of information, which follow the transmission and acknowledgment procedures used for the base Link Code Word. Two types of Next Page encodings are defined: Message Pages and Unformatted Pages.

A dual acknowledgment system is used. Acknowledge (Ack) is used to acknowledge receipt of the information; Acknowledge 2 (Ack2) is used to indicate that the receiver is able to act on the information (or perform the task) defined in the message.

Next Page operation is controlled by the same two mandatory control bits, Next Page and Acknowledge, used in the Base Link Code Word. Setting the NP bit in the Base Link Code Word to logic one indicates that the device is Next Page Able. If both a device and its Link Partner are Next Page Able, then Next Page exchange may occur. If one or both devices are not Next Page Able, then Next Page exchange will not occur.

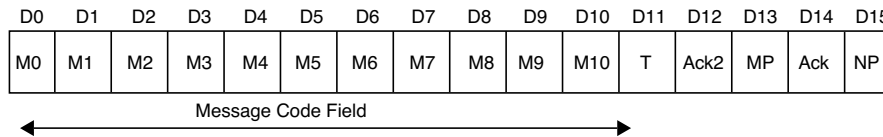
and, after the base Link Code Words have been exchanged, the FLP LINK GOOD CHECK state will be entered. The Toggle bit is used to ensure proper synchronization between the Local Device and the Link Partner.

Next Page exchange occurs after the base Link Code Words have been exchanged. Next Page exchange consists of using the normal Auto-Negotiation arbitration process to send Next Page messages. Two message encodings are defined: Message Pages, which contain predefined 11 bit codes, and Unformatted Pages. Unformatted Pages can be combined to send extended messages. If the Selector Field values do not match, then each series of Unformatted Pages shall be preceded by a Message Page containing a message code that defines how the following Unformatted Pages will be interpreted. If the Selector Field values match, then the convention governing the use of Message Pages shall be as defined by the Selector Field value definition. Any number of Next Pages may be sent in any order; however, it is recommended that the total number of Next Pages sent be kept small to minimize the link startup time.

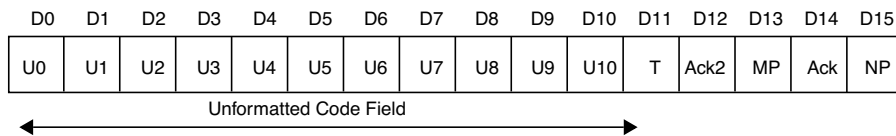
Next Page transmission ends when both ends of a link segment set their Next Page bits to logic zero, indicating that neither has anything additional to transmit. It is possible for one device to have more pages to transmit than the other device. Once a device has completed transmission of its Next Page information, it shall transmit Message Pages with Null message codes and the NP bit set to logic zero while its Link Partner continues to transmit valid Next Pages. An Auto-Negotiation able device shall recognize reception of Message Pages with Null message codes as the end of its Link Partner's Next Page information.

**28.2.3.4.1 Next Page encodings**

The Next Page shall use the encoding shown in Figure 28–11 and Figure 28–12 for the NP, Ack, MP, Ack2, and T bits. The 11-bit field D10–D0 shall be encoded as a Message Code Field if the MP bit is logic one and an Unformatted Code Field if MP is set to logic zero.



**Figure 28–11 — Message Page encoding**



**Figure 28–12 — Unformatted Page encoding**

**28.2.3.4.2 Next Page**

Next Page (NP) is used by the Next Page function to indicate whether or not this is the last Next Page to be transmitted. NP shall be set as follows:

- logic zero = last page.
- logic one = additional Next Page(s) will follow.



### **28.2.3.4.3 Acknowledge**

As defined in 28.2.1.2.4.

### **28.2.3.4.4 Message Page**

Message Page (MP) is used by the Next Page function to differentiate a Message Page from an Unformatted Page. MP shall be set as follows:

logic zero = Unformatted Page.  
logic one = Message Page.

### **28.2.3.4.5 Acknowledge 2**

Acknowledge 2 (Ack2) is used by the Next Page function to indicate that a device has the ability to comply with the message. Ack2 shall be set as follows:

logic zero = cannot comply with message.  
logic one = will comply with message.

### **28.2.3.4.6 Toggle**

Toggle (T) is used by the Arbitration function to ensure synchronization with the Link Partner during Next Page exchange. This bit shall always take the opposite value of the Toggle bit in the previously exchanged Link Code Word. The initial value of the Toggle bit in the first Next Page transmitted is the inverse of bit 11 in the base Link Code Word and, therefore, may assume a value of logic one or zero. The Toggle bit shall be set as follows:

logic zero = previous value of the transmitted Link Code Word equalled logic one.  
logic one = previous value of the transmitted Link Code Word equalled logic zero.

### **28.2.3.4.7 Message Page encoding**

Message Pages are formatted pages that carry a single predefined Message Code, which is enumerated in Annex 28C. Two-thousand and forty-eight Message Codes are available. The allocation of these codes will be controlled by the contents of Annex 28C. If the Message Page bit is set to logic one, then the bit encoding of the Link Code Word shall be interpreted as a Message Page.

### **28.2.3.4.8 Message Code Field**

Message Code Field (M[10:0]) is an eleven bit wide field, encoding 2048 possible messages. Message Code Field definitions are shown in Annex 28C. Combinations not specified are reserved for future use. Reserved combinations of the Message Code Field shall not be transmitted.

### **28.2.3.4.9 Unformatted Page encoding**

Unformatted Pages carry the messages indicated by Message Pages. Five control bits are predefined, the remaining 11 bits may take on an arbitrary value. If the Message Page bit is set to logic zero, then the bit encoding of the Link Code Word shall be interpreted as an Unformatted Page.

### **28.2.3.4.10 Unformatted Code Field**

Unformatted Code Field (U[10:0]) is an eleven bit wide field, which may contain an arbitrary value.

#### 28.2.3.4.11 Use of Next Pages

- a) Both devices must indicate Next Page ability for either to commence exchange of Next Pages.
- b) If both devices are Next Page able, then both devices shall send at least one Next Page.
- c) Next Page exchange shall continue until neither device on a link has more pages to transmit as indicated by the NP bit. A Message Page with a Null Message Code Field value shall be sent if the device has no other information to transmit.
- d) A Message Code can carry either a specific message or information that defines how following Unformatted Page(s) should be interpreted.
- e) If a Message Code references Unformatted Pages, the Unformatted Pages shall immediately follow the referencing Message Code in the order specified by the Message Code.
- f) Unformatted Page users are responsible for controlling the format and sequencing for their Unformatted Pages.

#### 28.2.3.4.12 MII register requirements

The Next Page Transmit register defined in 28.2.4.1.6 shall hold the Next Page to be sent by Auto-Negotiation. Received Next Pages may be stored in the Auto-Negotiation link partner ability register.

#### 28.2.3.5 Remote fault sensing function

The Remote Fault function may indicate to the Link Partner that a fault condition has occurred using the Remote Fault bit and, optionally, the Next Page function.

Sensing of faults in a device as well as subsequent association of faults with the Remote Fault bit shall be optional. If the Local Device has no mechanism to detect a fault or associate a fault condition with the received Remote Fault bit indication, then it shall transmit the Remote Fault bit with the value contained in the Auto-Negotiation advertisement register bit (4.13).

A Local Device may indicate it has sensed a fault to its Link Partner by setting the Remote Fault bit in the Auto-Negotiation advertisement register and renegotiating.

If the Local Device sets the Remote Fault bit to logic one, it may also use the Next Page function to specify information about the fault that has occurred. Remote Fault Message Page Codes have been specified for this purpose.

The Remote Fault bit shall remain set until after successful negotiation with the base Link Code Word, at which time the Remote Fault bit shall be reset to a logic zero. On receipt of a base Link Code Word with the Remote Fault bit set to logic one, the device shall set the Remote Fault bit in the MII status register (1.4) to logic one if the MII management function is present.

### 28.2.4 Management function requirements

The management interface is used to communicate Auto-Negotiation information to the management entity. If an MII is physically implemented, then management access is via the MII Management interface. Where no physical embodiment of the MII exists, an equivalent to MII Registers 0, 1, 4, 5, 6, and 7 (Clause 22) are recommended to be provided.

#### 28.2.4.1 Media Independent Interface

The Auto-Negotiation function shall have five dedicated registers:

- a) MII control register (Register 0).
- b) MII status register (Register 1).

- c) Auto-Negotiation advertisement register (Register 4).
- d) Auto-Negotiation link partner ability register (Register 5).
- e) Auto-Negotiation expansion register (Register 6).

If the Next Page function is implemented, the Auto-Negotiation next page transmit register (Register 7) shall be implemented.

#### **28.2.4.1.1 MII control register**

MII control register (Register 0) provides the mechanism to disable/enable and/or restart Auto-Negotiation. The definition for this register is provided in 28.2.4.1.

The Auto-Negotiation function shall be enabled by setting bit 0.12 to a logic one. If bit 0.12 is set to a logic one, then bits 0.13 and 0.8 shall have no effect on the link configuration, and the Auto-Negotiation process will determine the link configuration. If bit 0.12 is cleared to logic zero, then bits 0.13 and 0.8 will determine the link configuration regardless of the prior state of the link configuration and the Auto-Negotiation process.

A PHY shall return a value of one in bit 0.9 until the Auto-Negotiation process has been initiated. The Auto-Negotiation process shall be initiated by setting bit 0.9 to a logic one. If Auto-Negotiation was completed prior to this bit being set, the process shall be reinitiated. If a PHY reports via bit 1.3 that it lacks the ability to perform Auto-Negotiation, then this bit will have no meaning, and should be written as zero. This bit is self-clearing. The Auto-Negotiation process shall not be affected by clearing this bit to logic zero.

#### **28.2.4.1.2 MII status register**

The MII status register (Register 1) includes information about all modes of operations supported by the Local Device's PHY, the status of Auto-Negotiation, and whether the Auto-Negotiation function is supported by the PHY or not. The definition for this register is provided in 28.2.4.2.

When read as a logic one, bit 1.5 indicates that the Auto-Negotiation process has been completed, and that the contents of Registers 4, 5, and 6 are valid. When read as a logic zero, bit 1.5 indicates that the Auto-Negotiation process has not been completed, and that the contents of Registers 4, 5, and 6 are meaningless. A PHY shall return a value of zero in bit 1.5 if Auto-Negotiation is disabled by clearing bit 0.12. A PHY shall also return a value of zero in bit 1.5 if it lacks the ability to perform Auto-Negotiation.

When read as logic one, bit 1.4 indicates that a remote fault condition has been detected. The type of fault as well as the criteria and method of fault detection is PHY specific. The Remote Fault bit shall be implemented with a latching function, such that the occurrence of a remote fault will cause the Remote Fault bit to become set and remain set until it is cleared. The Remote Fault bit shall be cleared each time Register 1 is read via the management interface, and shall also be cleared by a PHY reset.

When read as a one, bit 1.3 indicates that the PHY has the ability to perform Auto-Negotiation. When read as a logic zero, bit 1.3 indicates that the PHY lacks the ability to perform Auto-Negotiation.

#### **28.2.4.1.3 Auto-Negotiation advertisement register (Register 4) (R/W)**

This register contains the Advertised Ability of the PHY. (See Table 28–2). The bit definition for the base page is defined in 28.2.1.2. On power-up, before Auto-Negotiation starts, this register shall have the following configuration: The Selector Field (4.4:0) is set to an appropriate code as specified in Annex 28A. The Acknowledge bit (4.14) is set to logic zero. The Technology Ability Field (4.12:5) is set based on the values set in the MII status register (Register 1) (1.15:11) or equivalent. See also Annex 28D.

Only the bits in the Technology Ability Field that represent the technologies supported by the Local Device may be set. Any of the Technology Ability Field bits that may be set can also be cleared by management

**Table 28–2—Advertisement register bit definitions**

Bit(s)	Name	Description	R/W
4.15	Next Page	See 28.2.1.2	R/W
4.14	Reserved	Write as zero, ignore on read	RO
4.13	Remote Fault	See 28.2.1.2	R/W
4.12:5	Technology Ability Field	See 28.2.1.2	R/W
4.4:0	Selector Field	See 28.2.1.2	R/W

before a renegotiation. This can be used to enable management to Auto-Negotiate to an alternate common mode.

The management entity may initiate renegotiation with the Link Partner using alternate abilities by setting the Selector Field (4.4:0) and Technology Ability Field (4.12:5) to indicate the preferred mode of operation and setting the Restart Auto-Negotiation bit (0.9) in the control register (Register 0) to logic one.

Any writes to this register prior to completion of Auto-Negotiation as indicated by bit 1.5 should be followed by a renegotiation for the new values to be properly used for Auto-Negotiation. Once Auto-Negotiation has completed, this register value may be examined by software to determine the highest common denominator technology.

#### **28.2.4.1.4 Auto-Negotiation link partner ability register (Register 5) (RO)**

All of the bits in the Auto-Negotiation link partner ability register are read only. A write to the Auto-Negotiation link partner ability register shall have no effect.

This register contains the Advertised Ability of the Link Partner's PHY. (See Tables 28–3 and 28–4.) The bit definitions shall be a direct representation of the received Link Code Word (Figure 28–7). Upon successful completion of Auto-Negotiation, status register (Register 1) Auto-Negotiation Complete bit (1.5) shall be set to logic one. If the Next Page function is supported, the Auto-Negotiation link partner ability register may be used to store Link Partner Next Pages.

**Table 28–3—Link partner ability register bit definitions (Base Page)**

Bit(s)	Name	Description	R/W
5.15	Next Page	See 28.2.1.2	RO
5.14	Acknowledge	See 28.2.1.2	RO
5.13	Remote Fault	See 28.2.1.2	RO
5.12:5	Technology Ability Field	See 28.2.1.2	RO
5.4:0	Selector Field	See 28.2.1.2	RO

The values contained in this register are only guaranteed to be valid once Auto-Negotiation has successfully completed, as indicated by bit 1.5 or, if used with Next Page exchange, after the Page Received bit (6.1) has been set to logic one.

**Table 28–4—Link partner ability register bit definitions (Next Page)**

Bit(s)	Name	Description	R/W
5.15	Next Page	See 28.2.3.4	RO
5.14	Acknowledge	See 28.2.3.4	RO
5.13	Message Page	See 28.2.3.4	RO
5.12	Acknowledge 2	See 28.2.3.4	RO
5.11	Toggle	See 28.2.3.4	RO
5.10:0	Message/Unformatted Code Field	See 28.2.3.4	RO

NOTE—If this register is used to store Link Partner Next Pages, the previous value of this register is assumed to be stored by a management entity that needs the information overwritten by subsequent Link Partner Next Pages.

#### 28.2.4.1.5 Auto-Negotiation expansion register (Register 6) (RO)

All of the bits in the Auto-Negotiation expansion register are read only; a write to the Auto-Negotiation expansion register shall have no effect. (See Table 28–5.)

**Table 28–5—Expansion register bit definitions**

Bit(s)	Name	Description	R/W	Default
6.15:5	Reserved	Write as zero, ignore on read	RO	0
6.4	Parallel Detection Fault	1 = A fault has been detected via the Parallel Detection function. 0 = A fault has not been detected via the Parallel Detection function.	RO/ LH	0
6.3	Link Partner Next Page Able	1 = Link Partner is Next Page able 0 = Link Partner is not Next Page able	RO	0
6.2	Next Page Able	1 = Local Device is Next Page able 0 = Local Device is not Next Page able	RO	0
6.1	Page Received	1 = A New Page has been received 0 = A New Page has not been received	RO/ LH	0
6.0	Link Partner Auto-Negotiation Able	1 = Link Partner is Auto-Negotiation able 0 = Link Partner is not Auto-Negotiation able	RO	0

Bits 6.15:5 are reserved for future Auto-Negotiation expansion.

The Parallel Detection Fault bit (6.4) shall be set to logic one to indicate that zero or more than one of the NLP Receive Link Integrity Test function, 100BASE-TX, or 100BASE-T4 PMAs have indicated link\_status=READY when the autoneg\_wait\_timer expires. The Parallel Detection Fault bit shall be reset to logic zero on a read of the Auto-Negotiation expansion register (Register 6).

The Link Partner Next Page Able bit (6.3) shall be set to logic one to indicate that the Link Partner supports the Next Page function. This bit shall be reset to logic zero to indicate that the Link Partner does not support the Next Page function.

The Next Page Able bit (6.2) shall be set to logic one to indicate that the Local Device supports the Next Page function. The Next Page Able bit (6.2) shall be set to logic zero if the Next Page function is not supported.

The Page Received bit (6.1) shall be set to logic one to indicate that a new Link Code Word has been received and stored in the Auto-Negotiation link partner ability register. The Page Received bit shall be reset to logic zero on a read of the Auto-Negotiation expansion register (Register 6).

The Link Partner Auto-Negotiation Able bit (6.0) shall be set to logic one to indicate that the Link Partner is able to participate in the Auto-Negotiation function. This bit shall be reset to logic zero if the Link Partner is not Auto-Negotiation able.

#### 28.2.4.1.6 Auto-Negotiation Next Page transmit register (Register 7) (R/W)

The Auto-Negotiation Next Page Transmit register contains the Next Page Link Code Word to be transmitted when Next Page ability is supported. (See Table 28–6.) The contents are defined in 28.2.3.4. On power-up, this register shall contain the default value of 2001H, which represents a Message Page with the Message Code set to Null Message. This value may be replaced by any valid Next Page Message Code that the device wishes to transmit. Writing to this register shall set `mr_next_page_loaded` to true.

**Table 28–6—Next Page transmit register bit definitions**

Bit(s)	Name	Description	R/W
7.15	Next Page	See 28.2.3.4	R/W
7.14	Reserved	Write as 0, ignore on read	RO
7.13	Message Page	See 28.2.3.4	R/W
7.12	Acknowledge 2	See 28.2.3.4	R/W
7.11	Toggle	See 28.2.3.4	RO
7.10:0	Message/Unformatted Code field	See 28.2.3.4	R/W

#### 28.2.4.1.7 Auto-Negotiation Link Partner Ability register (Register 8) (RO)

Support for 100BASE-T2 and 1000BASE-T requires support for Next Page and the provision of an Auto-Negotiation Link Partner Next Page Ability register (register 8) to store Link Partner Next Pages as shown in Table 28–7. All of the bits in the Auto-Negotiation Link Partner Next Page Ability register are read only. A write to the Auto-Negotiation Link Partner Next Page Ability register shall have no effect.

The values contained in this register are only guaranteed to be valid after the Page Received bit (6.1) has been set to logical one or once Auto-Negotiation has successfully completed, as indicated by bit 1.5.

NOTE—If this register is used to store multiple Link Partner Next Pages, the previous value of this register is assumed to be stored by a management entity that needs the information overwritten by subsequent Link Partner Next Pages.

**Table 28–7—Link Partner Next Page Ability register bit definitions**

Bit(s)	Name	Description	R/W
8.15	Next Page	see 28.2.3.4	RO
8.14	Acknowledge	see 28.2.3.4	RO
8.13	Message Page	see 28.2.3.4	RO
8.12	Acknowledge 2	see 28.2.3.4	RO
8.11	Toggle	see 28.2.3.4	RO
8.10:0	Message/Unformatted Code Field	see 28.2.3.4	RO

**28.2.4.1.8 State diagram variable to MII register mapping**

The state diagrams of Figures 28–14 to 28–17 generate and accept variables of the form “mr\_x,” where x is an individual signal name. These variables comprise a management interface that may be connected to the MII management function or other equivalent function. Table 28–8 describes how the MII registers map to the management function interface signals.

**Table 28–8—State diagram variable to MII register mapping**

State diagram variable	MII register
mr_adv_ability[16:1]	4.15:0 Auto-Negotiation advertisement register
mr_autoneg_complete	1.5 Auto-Negotiation Complete
mr_autoneg_enable	0.12 Auto-Negotiation Enable
mr_lp_adv_ability[16:1]	5.15:0 Auto-Negotiation link partner ability register
mr_lp_autoneg_able	6.0 Link Partner Auto-Negotiation Able
mr_lp_np_able	6.3 Link Partner Next Page Able
mr_main_reset	0.15 Reset
mr_next_page_loaded	Set on write to Auto-Negotiation Next Page Transmit register; cleared by Arbitration state diagram
mr_np_able	6.2 Next Page Able
mr_np_tx[16:1]	7.15:0 Auto-Negotiation Next Page Transmit Register
mr_page_rx	6.1 Page Received
mr_parallel_detection_fault	6.4 Parallel Detection Fault
mr_restart_negotiation	0.9 Auto-Negotiation Restart
set if Auto-Negotiation is available	1.3 Auto-Negotiation Ability

### 28.2.4.2 Auto-Negotiation managed object class

The Auto-Negotiation Managed Object Class is defined in Clause 30.

### 28.2.5 Absence of management function

In the absence of any management function, the advertised abilities shall be provided through a logical equivalent of `mr_adv_ability[16:1]`. A device shall comply with all Next Page function requirements, including the provision of the `mr_np_able`, `mr_lp_np_able`, and `mr_next_page_loaded` variables (or their logical equivalents), in order to permit the NP bit to be set to logic one in the transmitted Link Code Word.

NOTE—Storage of a valid base Link Code Word is required to prevent a deadlock situation where negotiation must start again while Next Pages are being transmitted. If a shared transmit register were used, then renegotiation could not occur when Next Pages were being transmitted because the base Link Code Word would not be available. This requirement can be met using a number of different implementations, including use of temporary registers or register stacks.

### 28.2.6 Technology-Dependent Interface

The Technology-Dependent Interface is the communication mechanism between each technology's PMA and the Auto-Negotiation function. Auto-Negotiation can support multiple technologies, all of which need not be implemented in a given device. Each of these technologies may utilize its own technology-dependent link integrity test function.

#### 28.2.6.1 PMA\_LINK.indicate

This primitive is generated by the PMA to indicate the status of the underlying medium. The purpose of this primitive is to give the PCS, repeater client, or Auto-Negotiation function a means of determining the validity of received code elements.

##### 28.2.6.1.1 Semantics of the service primitive

`PMA_LINK.indicate(link_status)`

The `link_status` parameter shall assume one of three values: `READY`, `OK`, or `FAIL`, indicating whether the underlying receive channel is intact and ready to be enabled (`READY`), intact and enabled (`OK`), or not intact (`FAIL`). When `link_status=FAIL` or `link_status=READY`, the `PMA_CARRIER.indicate` and `PMA_UNITDATA.indicate` primitives are undefined.

##### 28.2.6.1.2 When generated

A technology-dependent PMA and the NLP Receive Link Integrity Test state diagram (Figure 28–17) shall generate this primitive to indicate the value of `link_status`.

##### 28.2.6.1.3 Effect of receipt

The effect of receipt of this primitive shall be governed by the state diagrams of Figure 28–16.

#### 28.2.6.2 PMA\_LINK.request

This primitive is generated by Auto-Negotiation to allow it to enable and disable operation of the PMA.

##### 28.2.6.2.1 Semantics of the service primitive

`PMA_LINK.request(link_control)`



The `link_control` parameter shall assume one of three values: `SCAN_FOR_CARRIER`, `DISABLE`, or `ENABLE`.

The `link_control=SCAN_FOR_CARRIER` mode is used by the Auto-Negotiation function prior to receiving any FLP Bursts or `link_status=READY` indications. During this mode, the PMA shall search for carrier and report `link_status=READY` when carrier is received, but no other actions shall be enabled.

The `link_control=DISABLE` mode shall be used by the Auto-Negotiation function to disable PMA processing.

The `link_control=ENABLE` mode shall be used by Auto-Negotiation to turn control over to a single PMA for all normal processing functions.

#### **28.2.6.2.2 When generated**

The Auto-Negotiation function shall generate this primitive to indicate to the PHY how to respond, in accordance with the state diagrams of Figure 28–15 and Figure 28–16.

Upon power-on or reset, if the Auto-Negotiation function is enabled (`mr_autoneg_enable=true`) the `PMA_LINK.request(DISABLE)` message shall be issued to all technology-dependent PMAs. If Auto-Negotiation is disabled at any time including at power-on or reset, the state of `PMA_LINK.request(link_control)` is implementation dependent.

#### **28.2.6.2.3 Effect of receipt**

The effect of receipt of this primitive shall be governed by the NLP Receive Link Integrity Test state diagram (Figure 28–17) and the receiving technology-dependent link integrity test function, based on the intent specified in the primitive semantics.

#### **28.2.6.3 PMA\_LINKPULSE.request**

This primitive is generated by Auto-Negotiation to indicate that a valid Link Pulse, as transmitted in compliance with Figure 14–12, has been received.

##### **28.2.6.3.1 Semantics of the service primitive**

`PMA_LINKPULSE.request(linkpulse)`

The `linkpulse` parameter shall assume one of two values: `TRUE` or `FALSE`.

The `linkpulse=FALSE` mode shall be used by the Auto-Negotiation function to indicate that the Receive State Diagram has performed a state transition.

The `linkpulse=TRUE` mode shall be used by the Auto-Negotiation function to indicate that a valid Link Pulse has been received.

##### **28.2.6.3.2 When generated**

The Auto-Negotiation function shall generate this primitive to indicate to the PHY how to respond, in accordance with the state diagram of Figure 28–15.

Upon power-on or reset, if the Auto-Negotiation function is enabled (`mr_autoneg_enable=true`) the `PMA_LINKPULSE.request(FALSE)` message shall be issued to all technology-dependent PMAs. If Auto-

Negotiation is disabled at any time including at power-on or reset, the state of PMA\_LINKPULSE.request (linkpulse) is implementation dependent.

**28.2.6.3.3 Effect of receipt**

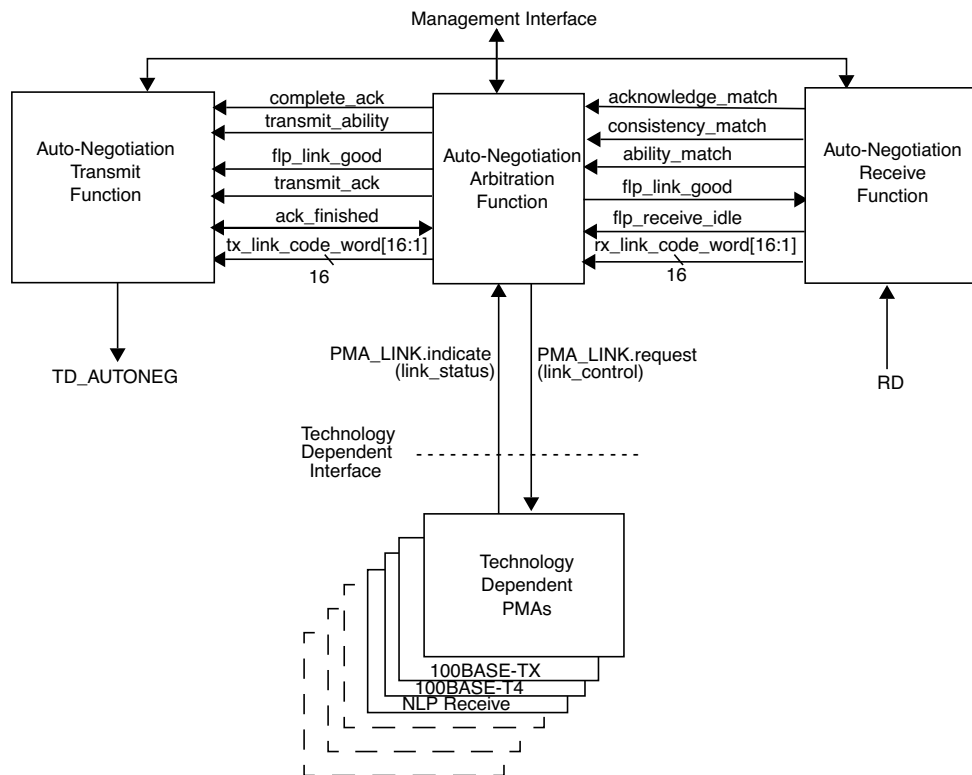
The effect of receipt of this primitive shall be governed by the receiving technology-dependent PMA function, based on the intent specified in the primitive semantics.

**28.3 State diagrams and variable definitions**

The notation used in the state diagrams (Figures 28–14 to 28–17) follows the conventions in 21.5. State diagram variables follow the conventions of 21.5.2 except when the variable has a default value. Variables in a state diagram with default values evaluate to the variable default in each state where the variable value is not explicitly set. Variables using the “mr\_x” notation do not have state diagram defaults; however, their appropriate initialization conditions when mapped to the MII interface are covered in 28.2.4 and 22.2.4. The variables, timers, and counters used in the state diagrams are defined in 28.3, 14.2.3, and 28.2.6.

Auto-Negotiation shall implement the Transmit state diagram, Receive state diagram, Arbitration state diagram, and NLP Receive Link Integrity Test state diagram as depicted in 28.3. Additional requirements to these state diagrams are made in the respective functional requirements sections. Options to these state diagrams clearly stated as such in the functional requirements sections or state diagrams shall be allowed. In the case of any ambiguity between stated requirements and the state diagrams, the state diagrams shall take precedence.

The functional reference diagram (Figure 28–13) provides a generic example, illustrated with initial PMA implementations and showing the mechanism for expansion. New PMAs are documented in Annex 28D.



**Figure 28–13—Functional reference diagram**

### 28.3.1 State diagram variables

A variable with “\_[x]” appended to the end of the variable name indicates a variable or set of variables as defined by “x”. “x” may be as follows:

- all; represents all specific technology-dependent PMAs supported in the Local Device and the NLP Receive Link Integrity Test state diagram.
- 1GigT; represents that the 1000BASE-T PMA is the signal source.
- HCD; represents the single technology-dependent PMA chosen by Auto-Negotiation as the highest common denominator technology through the Priority Resolution or Parallel Detection function. To select 10BASE-T, LIT is used instead of NLP to enable the full 10BASE-T Link Integrity Test function state diagram.
- notHCD; represents all technology-dependent PMAs not chosen by Auto-Negotiation as the highest common denominator technology through the Priority Resolution or Parallel Detection function.
- TX; represents that the 100BASE-TX PMA is the signal source.
- T4; represents that the 100BASE-T4 PMA is the signal source.
- NLP; represents that the NLP Receive Link Integrity Test function is the signal source.
- PD; represents all of the following that are present: 100BASE-TX PMA, 100BASE-T4 PMA, and the NLP Receive Link Integrity Test state diagram.
- LIT; represents the 10BASE-T Link Integrity Test function state diagram is the signal source or destination.

Variables with [16:1] appended to the end of the variable name indicate arrays that can be directly mapped to 16-bit registers. For these variables, “[x]” indexes an element or set of elements in the array, where “[x]” may be as follows:

- Any integer.
- Any variable that takes on integer values.
- NP; represents the index of the Next Page bit.
- ACK; represents the index of the Acknowledge bit.
- RF; represents the index of the Remote Fault bit.

Variables of the form “mr\_x”, where x is a label, comprise a management interface that is intended to be connected to the MII Management function. However, an implementation-specific management interface may provide the control and status function of these bits.

#### ability\_match

Indicates that three consecutive Link Code Words match, ignoring the Acknowledge bit. Three consecutive words are any three words received one after the other, regardless of whether the word has already been used in a word-match comparison or not.

Values: false; three matching consecutive Link Code Words have not been received, ignoring the Acknowledge bit (default).  
 true; three matching consecutive Link Code Words have been received, ignoring the Acknowledge bit.

NOTE—This variable is set by this variable definition; it is not set explicitly in the state diagrams.

**ability\_match\_word** [16:1]

A 16-bit array that contains the last Link Code Word that caused `ability_match = true`. For each element in the array:

Values: zero; data bit is logical zero.  
 one; data bit is logical one.

**ack\_finished**

Status indicating that the final remaining\_ack\_cnt Link Code Words with the Ack bit set have been transmitted.

Values: false; more Link Code Words with the Ack bit set to logic one must be transmitted.  
 true; all remaining Link Code Words with the Ack bit set to logic one have been transmitted.

**acknowledge\_match**

Indicates that three consecutive Link Code Words match and have the Acknowledge bit set. Three consecutive words are any three words received one after the other, regardless of whether the word has already been used in a word match comparison or not.

Values: false; three matching and consecutive Link Code Words have not been received with the Acknowledge bit set (default).  
 true; three matching and consecutive Link Code Words have been received with the Acknowledge bit set.

NOTE—This variable is set by this variable definition; it is not set explicitly in the state diagrams.

**base\_page**

Status indicating that the page currently being transmitted by Auto-Negotiation is the initial Link Code Word encoding used to communicate the device's abilities.

Values: false; a page other than base Link Code Word is being transmitted.  
 true; the base Link Code Word is being transmitted.

**complete\_ack**

Controls the counting of transmitted Link Code Words that have their Acknowledge bit set.

Values: false; transmitted Link Code Words with the Acknowledge bit set are not counted (default).  
 true; transmitted Link Code Words with the Acknowledge bit set are counted.

**consistency\_match**

Indicates that the Link Code Word that caused `ability_match` to be set is the same as the Link Code Word that caused `acknowledge_match` to be set.

Values: false; the Link Code Word that caused `ability_match` to be set is not the same as the Link Code Word that caused `acknowledge_match` to be set, ignoring the Acknowledge bit value.  
 true; the Link Code Word that caused `ability_match` to be set is the same as the Link Code Word that caused `acknowledge_match` to be set, independent of the Acknowledge bit value.

NOTE—This variable is set by this variable definition; it is not set explicitly in the state diagrams.

**desire\_np**

Status indicating that the Local Device desires to engage in Next Page exchange. This information comes from the setting of the NP bit in the base Link Code Word stored in the Auto-Negotiation advertisement register (Register 4).

Values: false; Next Page exchange is not desired.  
 true; Next Page exchange is desired.

**flp\_link\_good**

Indicates that Auto-Negotiation has completed.

Values: false; negotiation is in progress (default).

true; negotiation is complete, forcing the Transmit and Receive functions to IDLE.

#### flp\_receive\_idle

Indicates that the Receive state diagram is in the IDLE, LINK PULSE DETECT, or LINK PULSE COUNT state.

Values: false; the Receive state diagram is not in the IDLE, LINK PULSE DETECT, or LINK PULSE COUNT state (default).  
true; the Receive state diagram is in the IDLE, LINK PULSE DETECT, or LINK PULSE COUNT state.

#### incompatible\_link

Parameter used following Priority Resolution to indicate the resolved link is incompatible with the Local Device settings. A device's ability to set this variable to true is optional.

Values: false; A compatible link exists between the Local Device and Link Partner (default).  
true; Optional indication that Priority Resolution has determined no highest common denominator exists following the most recent negotiation.

NOTE—This variable is set by this variable definition; it is not set explicitly in the state diagrams.

#### link\_control

This variable is defined in 28.2.6.2.1.

#### link\_status

This variable is defined in 28.2.6.1.1.

#### linkpulse

This variable is defined in 28.2.6.3.1.

Values: false; linkpulse is set to false after any Receive State Diagram state transition (default).  
true; linkpulse is set to true when a valid Link Pulse is received.

#### mr\_autoneg\_complete

Status indicating whether Auto-Negotiation has completed or not.

Values: false; Auto-Negotiation has not completed.  
true; Auto-Negotiation has completed.

#### mr\_autoneg\_enable

Controls the enabling and disabling of the Auto-Negotiation function.

Values: false; Auto-Negotiation is disabled.  
true; Auto-Negotiation is enabled.

#### mr\_adv\_ability[16:1]

A 16-bit array that contains the Advertised Abilities Link Code Word.  
For each element within the array:

Values: zero; data bit is logical zero.  
one; data bit is logical one.

#### mr\_lp\_adv\_ability[16:1]

A 16-bit array that contains the Link Partner's Advertised Abilities Link Code Word.  
For each element within the array:

Values: zero; data bit is logical zero.  
one; data bit is logical one.

#### mr\_lp\_np\_able

Status indicating whether the Link Partner supports Next Page exchange.

Values: false; the Link Partner does not support Next Page exchange.  
true; the Link Partner supports Next Page exchange.

#### mr\_np\_able

Status indicating whether the Local Device supports Next Page exchange.

Values: false; the Local Device does not support Next Page exchange.

true; the Local Device supports Next Page exchange.

#### mr\_lp\_autoneg\_able

Status indicating whether the Link Partner supports Auto-Negotiation.

Values: false; the Link Partner does not support Auto-Negotiation.  
true; the Link Partner supports Auto-Negotiation.

#### mr\_main\_reset

Controls the resetting of the Auto-Negotiation state diagrams.

Values: false; do not reset the Auto-Negotiation state diagrams.  
true; reset the Auto-Negotiation state diagrams.

#### mr\_next\_page\_loaded

Status indicating whether a new page has been loaded into the Auto-Negotiation Next Page Transmit register (Register 7).

Values: false; a New Page has not been loaded.  
true; a New Page has been loaded.

#### mr\_np\_tx[16:1]

A 16-bit array that contains the new Next Page to transmit.

For each element within the array:

Values: zero; data bit is logical zero.  
one; data bit is logical one.

#### mr\_page\_rx

Status indicating whether a New Page has been received. A New Page has been successfully received when `acknowledge_match=true` and `consistency_match=true` and the Link Code Word has been written to `mr_lp_adv_ability[16:1]`.

Values: false; a New Page has not been received.  
true; a New Page has been received.

#### mr\_parallel\_detection\_fault

Error condition indicating that while performing Parallel Detection, either `flp_receive_idle = false`, or zero or more than one of the following indications were present when the `autoneg_wait_timer` expired. This signal is cleared on read of the Auto-Negotiation expansion register.

- 1) `link_status_[NLP] = READY`
- 2) `link_status_[TX] = READY`
- 3) `link_status_[T4] = READY`

Values: false; Exactly one of the above three indications was true when the `autoneg_wait_timer` expired, and `flp_receive_idle = true`.  
true; either zero or more than one of the above three indications was true when the `autoneg_wait_timer` expired, or `flp_receive_idle = false`.

#### mr\_restart\_negotiation

Controls the entrance to the TRANSMIT DISABLE state to break the link before Auto-Negotiation is allowed to renegotiate via management control.

Values: false; renegotiation is not taking place.  
true; renegotiation is started.

#### np\_rx

Flag to hold the value of `rx_link_code_word[NP]` upon entry to the COMPLETE ACKNOWLEDGE state. This value is associated with the value of `rx_link_code_word[NP]` when `acknowledge_match` was last set.

Values: zero; local device `np_rx` bit equals a logical zero.  
one; local device `np_rx` bit equals a logical one.

#### power\_on

Condition that is true until such time as the power supply for the device that contains the Auto-

Negotiation state diagrams has reached the operating region or the device has low power mode set via MII control register bit 0.11.

Values: false; the device is completely powered (default).  
true; the device has not been completely powered.

rx\_link\_code\_word[16:1]

A 16-bit array that contains the data bits to be received from an FLP Burst.

For each element within the array:

Values: zero; data bit is a logical zero.  
one; data bit is a logical one.

single\_link\_ready

Status indicating that flp\_receive\_idle = true and only one the of the following indications is being received:

- 1) link\_status\_[NLP] = READY
- 2) link\_status\_[TX] = READY
- 3) link\_status\_[T4] = READY

Values: false; either zero or more than one of the above three indications are true or flp\_receive\_idle = false.  
true; Exactly one of the above three indications is true and flp\_receive\_idle = true.

NOTE—This variable is set by this variable definition; it is not set explicitly in the state diagrams.

TD\_AUTONEG

Controls the signal sent by Auto-Negotiation on the TD\_AUTONEG circuit.

Values: idle; Auto-Negotiation prevents transmission of all link pulses on the MDI.  
link\_test\_pulse; Auto-Negotiation causes a single link pulse as defined by Figure 14–12 to be transmitted on the MDI.

toggle\_rx

Flag to keep track of the state of the Link Partner's Toggle bit.

Values: 0; Link Partner's Toggle bit equals logic zero.  
1; Link Partner's Toggle bit equals logic one.

toggle\_tx

Flag to keep track of the state of the Local Device's Toggle bit.

Values: 0; Local Device's Toggle bit equals logic zero.  
1; Local Device's Toggle bit equals logic one.

transmit\_ability

Controls the transmission of the Link Code Word containing tx\_link\_code\_word[16:1].

Values: false; any transmission of tx\_link\_code\_word[16:1] is halted (default).  
true; the transmit state diagram begins sending tx\_link\_code\_word[16:1].

transmit\_ack

Controls the setting of the Acknowledge bit in the tx\_link\_code\_word[16:1] to be transmitted.

Values: false; sets the Acknowledge bit in the transmitted tx\_link\_code\_word[16:1] to a logic zero (default).  
true; sets the Acknowledge bit in the transmitted tx\_link\_code\_word[16:1] to a logic one.

transmit\_disable

Controls the transmission of tx\_link\_code\_word[16:1].

Values: false; tx\_link\_code\_word[16:1] transmission is allowed (default).  
true; tx\_link\_code\_word[16:1] transmission is halted.

**tx\_link\_code\_word[16:1]**

A 16-bit array that contains the data bits to be transmitted in an FLP Burst. This array may be loaded from `mr_adv_ability` or `mr_np_tx`.  
For each element within the array:

Values: Zero; data bit is logical zero.  
One; data bit is logical one.

**28.3.2 State diagram timers**

All timers operate in the manner described in 14.2.3.2.

**autoneg\_wait\_timer**

Timer for the amount of time to wait before evaluating the number of link integrity test functions with `link_status=READY` asserted. The `autoneg_wait_timer` shall expire 500–1000 ms from the assertion of `link_status=READY` from the 100BASE-TX PMA, 100BASE-T4 PMA, or the NLP Receive State diagram.

**break\_link\_timer**

Timer for the amount of time to wait in order to assure that the Link Partner enters a Link Fail state. The timer shall expire 1200–1500 ms after being started.

**data\_detect\_max\_timer**

Timer for the maximum time between a clock pulse and the next link pulse. This timer is used in conjunction with the `data_detect_min_timer` to detect whether the data bit between two clock pulses is a logic zero or a logic one. The `data_detect_max_timer` shall expire 78–100  $\mu$ s from the last clock pulse.

**data\_detect\_min\_timer**

Timer for the minimum time between a clock pulse and the next link pulse. This timer is used in conjunction with the `data_detect_max_timer` to detect whether the data bit between two clock pulses is a logic zero or a logic one. The `data_detect_min_timer` shall expire 15–47  $\mu$ s from the last clock pulse.

**flp\_test\_max\_timer**

Timer for the maximum time between two link pulses within an FLP Burst. This timer is used in conjunction with the `flp_test_min_timer` to detect whether the Link Partner is transmitting FLP Bursts. The `flp_test_max_timer` shall expire 165–185  $\mu$ s from the last link pulse.

**flp\_test\_min\_timer**

Timer for the minimum time between two link pulses within an FLP Burst. This timer is used in conjunction with the `flp_test_max_timer` to detect whether the Link Partner is transmitting FLP Bursts. The `flp_test_min_timer` shall expire 5–25  $\mu$ s from the last link pulse.

**interval\_timer**

Timer for the separation of a transmitted clock pulse from a data bit. The `interval_timer` shall expire 55.5–69.5  $\mu$ s from each clock pulse and data bit.

**link\_fail\_inhibit\_timer**

Timer for qualifying a `link_status=FAIL` indication or a `link_status=READY` indication when a specific technology link is first being established. A link will only be considered “failed” if the `link_fail_inhibit_timer` has expired and the link has still not gone into the `link_status=OK` state. The `link_fail_inhibit_timer` shall expire 750–1000 ms after entering the FLP LINK GOOD CHECK state.

NOTE—The `link_fail_inhibit_timer` expiration value must be greater than the time required for the Link Partner to complete Auto-Negotiation after the Local Device has completed Auto-Negotiation plus the time required for the specific technology to enter the `link_status=OK` state. The maximum time difference between a Local Device and its Link Partner completing Auto-Negotiation is



(Maximum FLP Burst to FLP Burst separation) × (Maximum number of FLP Bursts needed to complete acknowledgment) = (24 ms) × (8 bursts) = 192 ms.

For example, 100BASE-T4 requires approximately 460 ms to enter link\_status=OK for a total minimum link\_fail\_inhibit\_timer time of 652 ms. The lower bound for the link\_fail\_inhibit\_timer was chosen to provide adequate margin for the current technologies and any future PMAs.

#### nlp\_test\_max\_timer

Timer for the maximum time that no FLP Burst may be seen before forcing the receive state diagram to the IDLE state. The nlp\_test\_max\_timer shall expire 50–150 ms after being started or restarted.

#### nlp\_test\_min\_timer

Timer for the minimum time between two consecutive FLP Bursts. The nlp\_test\_min\_timer shall expire 5–7 ms after being started or restarted.

#### transmit\_link\_burst\_timer

Timer for the separation of a transmitted FLP Burst from the next FLP Burst. The transmit\_link\_burst\_timer shall expire 5.7-22.3 ms after the last transmitted link pulse in an FLP Burst.

**Table 28–9—Timer min./max. value summary**

Parameter	Min.	Typ.	Max.	Units
autoneg_wait_timer	500		1000	ms
break_link_timer	1200		1500	ms
data_detect_min_timer	15		47	$\mu$ s
data_detect_max_timer	78		100	$\mu$ s
flp_test_min_timer	5		25	$\mu$ s
flp_test_max_timer	165		185	$\mu$ s
interval_timer	55.5	62.5	69.5	$\mu$ s
link_fail_inhibit_timer	750		1000	ms
nlp_test_max_timer	50		150	ms
nlp_test_min_timer	5		7	ms
transmit_link_burst_timer	5.7	14	22.3	ms

### 28.3.3 State diagram counters

#### flp\_cnt

A counter that may take on integer values from 0 to 17. This counter is used to keep a count of the number of FLPs detected to enable the determination of whether the Link Partner supports Auto-Negotiation.

Values: not\_done; 0 to 5 inclusive.  
done; 6 to 17 inclusive.  
init; counter is reset to zero.

#### remaining\_ack\_cnt

A counter that may take on integer values from 0 to 8. The number of additional Link Code Words with the Acknowledge Bit set to logic one to be sent to ensure that the Link Partner receives the acknowledgment.

Values: not\_done; positive integers between 0 and 5 inclusive.  
done; positive integers 6 to 8 inclusive (default).  
init; counter is reset to zero.

#### rx\_bit\_cnt

A counter that may take on integer values from 0 to 17. This counter is used to keep a count of data bits received from an FLP Burst and to ensure that when erroneous extra pulses are received, the first 16 bits are kept while the rest are ignored. When this variable reaches 16 or 17, enough data bits have been received. This counter does not increment beyond 17 and does not return to 0 until it is reinitialized.

Values: not\_done; 1 to 15 inclusive.  
done; 16 or 17  
init; counter is reset to zero.  
rx\_bit\_cnt\_check; 10 to 17 inclusive.

#### tx\_bit\_cnt

A counter that may take on integer values from 1 to 17. This counter is used to keep a count of data bits sent within an FLP Burst. When this variable reaches 17, all data bits have been sent.

Values: not\_done; 1 to 16 inclusive.  
done; 17.  
init; counter is initialized to 1.

28.3.4 State diagrams

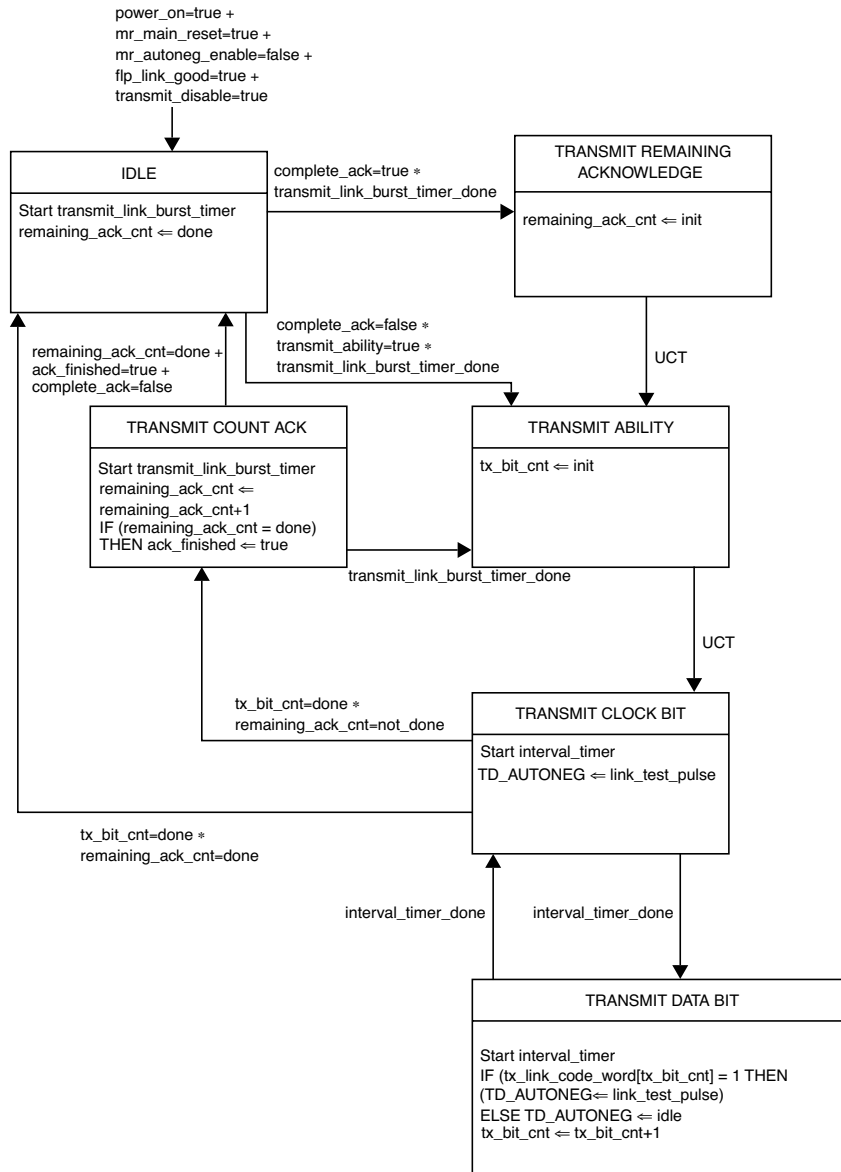


Figure 28–14—Transmit state diagram

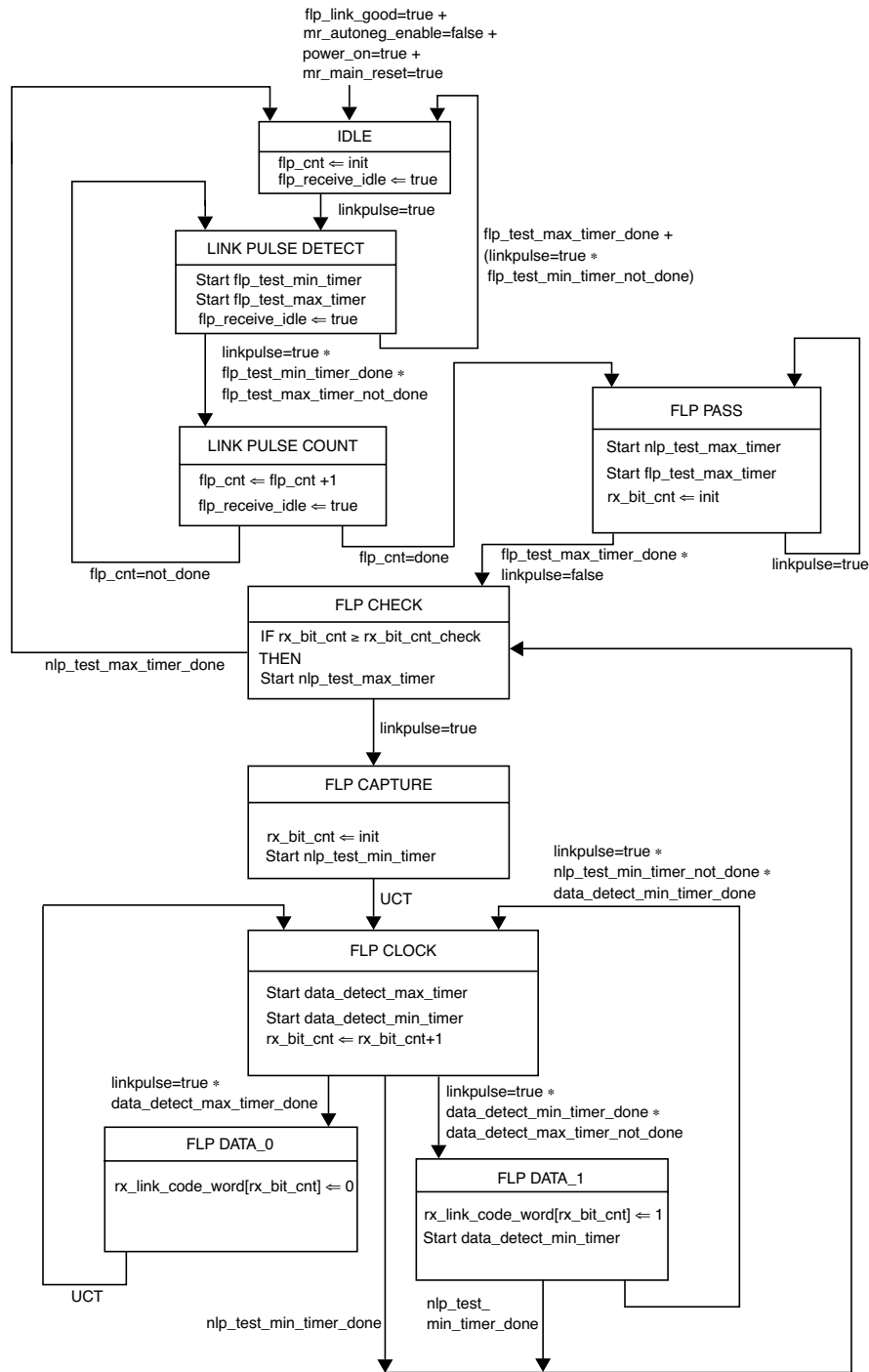
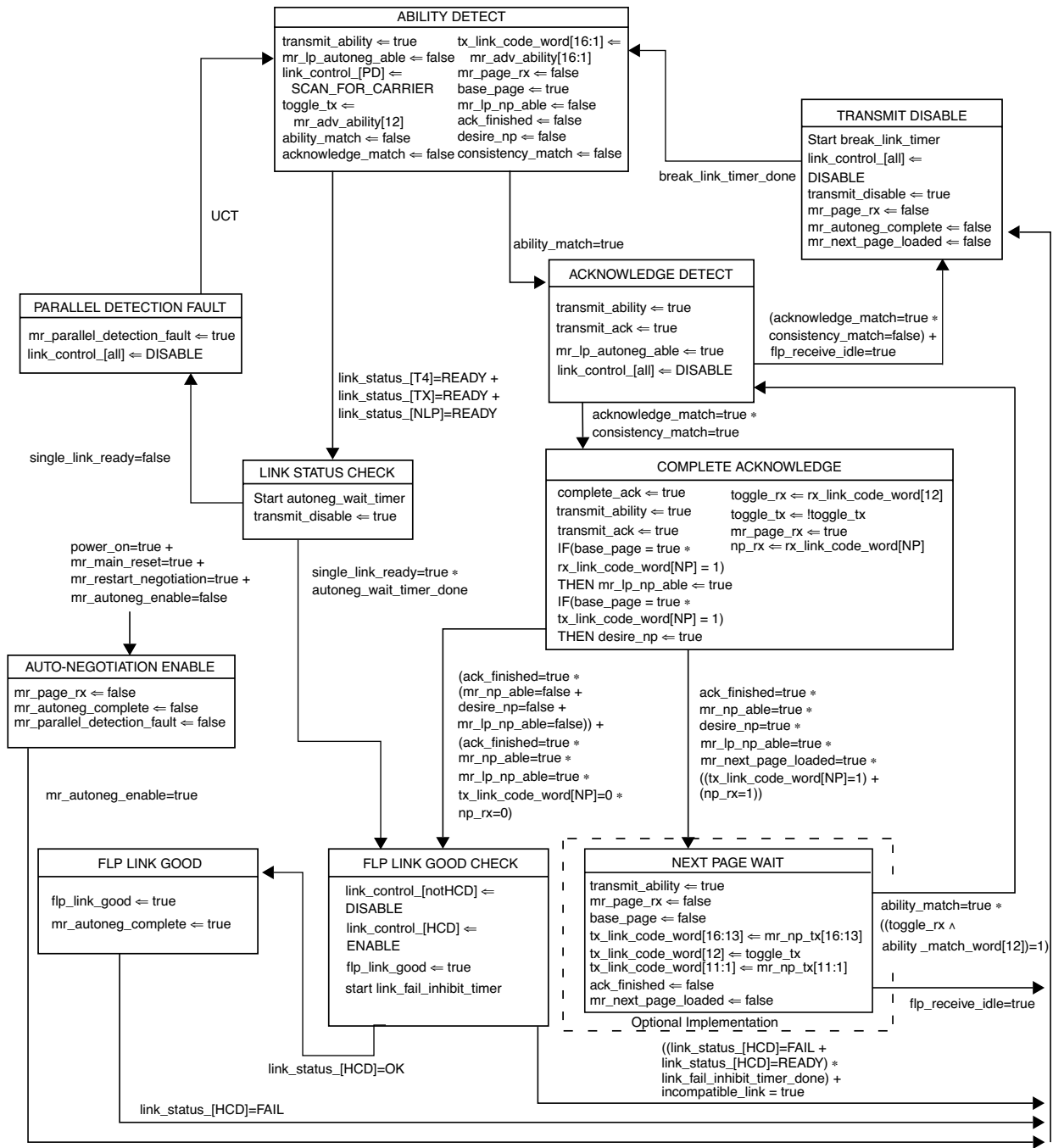


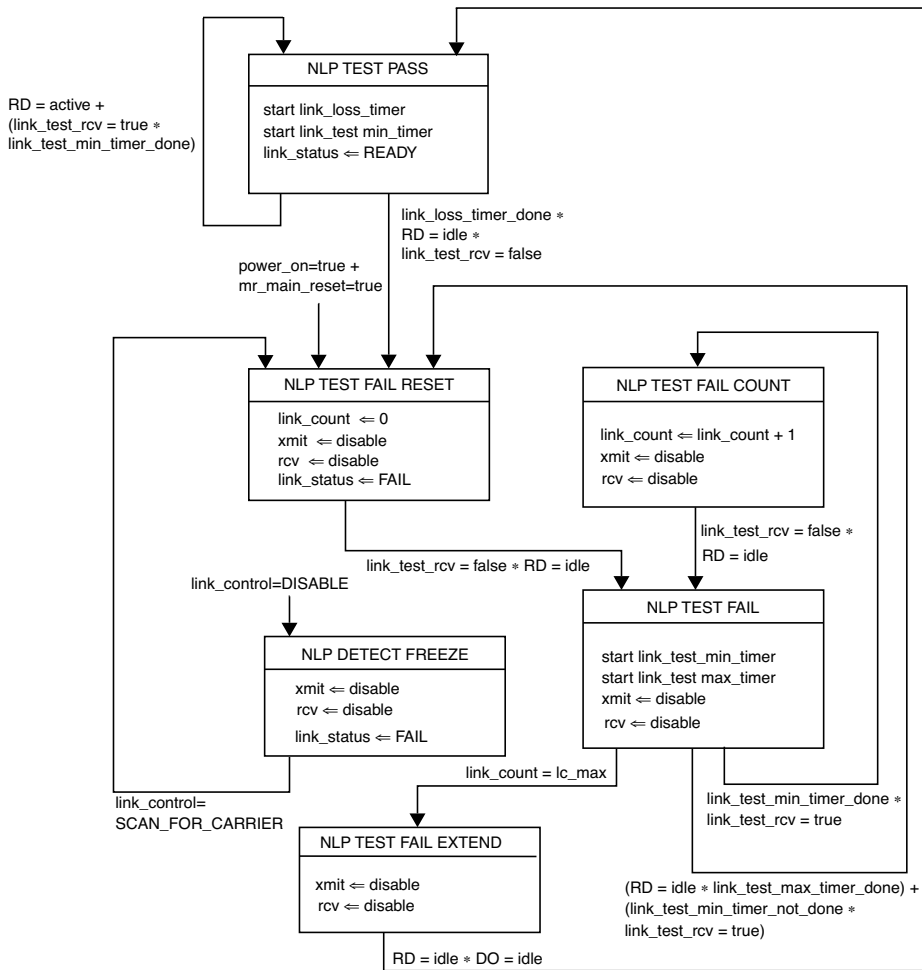
Figure 28–15—Receive state diagram



NOTE—The transition from COMPLETE ACKNOWLEDGE to FLP LINK GOOD CHECK can be simplified to “ack\_finished=true” if the optional Next Page function is not supported.

NOTE—ability\_match, acknowledge\_match, single\_link\_ready, consistency\_match, and incompatible\_link are set according to the variable definitions and are not set explicitly in the state diagrams.

Figure 28–16—Arbitration state diagram



NOTE—The variables link\_control and link\_status are viewed as dedicated signals by the NLP Receive Link integrity Test state diagram, but are viewed as link\_control\_[NLP] and link\_status\_[NLP] by the Auto-Negotiation Arbitration state diagram, Figure 28-16.

Figure 28-17—NLP Receive Link Integrity Test state diagram

### 28.4 Electrical specifications

The electrical characteristics of pulses within FLP Bursts shall be identical to the characteristics of NLPs and shall meet the requirements of Figure 14-12.

It is the responsibility of the technology-specific Transmit and Receive functions to interface to the MDI correctly.

NOTE—The requirements relative to the interface to the MDI are specified via the Transmit Switch and Receive Switch functions.

## 28.5 Protocol Implementation Conformance Statement (PICS) proforma for Clause 28, Physical Layer link signaling for 10 Mb/s, 100 Mb/s, and 1000 Mb/s Auto-Negotiation on twisted pair<sup>11</sup>

### 28.5.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 28, Physical Layer link signaling for 10 Mb/s, 100 Mb/s, and 1000 Mb/s Auto-Negotiation on twisted pair, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

### 28.5.2 Identification

#### 28.5.2.1 Implementation identification

Supplier	
Contact point for enquiries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s)	
NOTE 1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.	
NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

#### 28.5.2.2 Protocol summary

Identification of protocol standard	IEEE Std 802.3-2002 <sup>®</sup> , Clause 28, Physical Layer link signaling for 10 Mb/s, 100 Mb/s, and 1000 Mb/s Auto-Negotiation on twisted pair
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
Have any Exception items been required? No [ ] Yes [ ] (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3-2002 <sup>®</sup> .)	
Date of Statement	

<sup>11</sup>Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

**28.5.3 Major capabilities/options**

Item	Feature	Subclause	Status	Support	Value/comment
10BT	Implementation supports a 10BASE-T data service	28.1.2	O		N/A
*NP	Implementation supports Next Page function	28.1.2	O		N/A
*MII	Implementation supports the MII Management Interface	28.1.2	O/1		N/A
MGMT	Implementation supports a non-MII Management Interface	28.1.2	O/1		N/A
*NOM	Implementation does not support management	28.1.2	O/1		N/A
*RF	Implementation supports Remote Fault Sensing	28.2.3.5	O		N/A

**28.5.4 PICS proforma tables for Physical Layer link signaling for 10 Mb/s, 100 Mb/s, and 1000 Mb/s Auto-Negotiation on twisted pair****28.5.4.1 Scope**

Item	Feature	Subclause	Status	Support	Value/comment
1	MII Management Interface control and status registers	28.1.3	MII:M		Implemented in accordance with the definitions in Clause 22 and 28.2.4
2	CSMA/CD compatible devices using an eight-pin modular connector and using a signaling method to automatically configure the preferred mode of operation	28.1.4	M		Auto-Negotiation function implemented in compliance with Clause 28
3	Device uses 10BASE-T compatible link signaling to advertise non-CSMA/CD abilities	28.1.4	M		Auto-Negotiation function implemented in compliance with Clause 28
4	Future CSMA/CD implementations that use an eight-pin modular connector	28.1.4	M		Interoperable with devices compliant with Clause 28



**28.5.4.2 Auto-Negotiation functions**

Item	Feature	Subclause	Status	Support	Value/comment
1	Transmit	28.2	M		Complies with Figure 28–14
2	Receive	28.2	M		Complies with Figure 28–15
3	Arbitration	28.2	M		Complies with Figure 28–16
4	NLP Receive Link Integrity Test	28.2	M		Complies with Figure 28–17
5	Technology-Dependent Interface	28.2	M		Complies with 28.2.6
6	Technology-dependent link integrity test	28.2	M		Implemented and interfaced to for those technologies supported by device
7	Management	28.2	O		MII based or alternate management

**28.5.4.3 Transmit function requirements**

Item	Feature	Subclause	Status	Support	Value/comment
1	FLP Burst transmission	28.2.1.1	M		Not transmitted once Auto-Negotiation is complete and highest common denominator PMA has been enabled. Prohibited other than for link start-up
2	FLP Burst composition	28.2.1.1.1	M		Pulses in FLP Bursts meet the requirements of Figure 14–12
3	FLP Burst pulse definition	28.2.1.1.1	M		17 odd-numbered pulse positions represent clock information; 16 even-numbered pulse positions represent data information
4	The first pulse in an FLP Burst	28.2.1.1.2	M		Defined as a clock pulse for timing purposes
5	FLP Burst clock pulse spacing	28.2.1.1.2	M		Within an FLP Burst, spacing is $125 \pm 14 \mu\text{s}$
6	Logic one data bit representation	28.2.1.1.2	M		Pulse transmitted $62.5 \pm 7 \mu\text{s}$ after the preceding clock pulse
7	Logic zero data bit representation	28.2.1.1.2	M		No link integrity test pulses within $111 \mu\text{s}$ of the preceding clock pulse
8	Consecutive FLP Bursts	28.2.1.1.2	M		The first link pulse in each FLP Burst is separated by $16 \pm 8 \text{ ms}$
9	FLP Burst base page	28.2.1.2	M		Conforms to Figure 28–7
10	FLP Burst bit transmission order	28.2.1.2	M		Transmission is D0 first to D15 last

**28.5.4.3 Transmit function requirements (continued)**

Item	Feature	Subclause	Status	Support	Value/comment
11	Selector Field values	28.2.1.2.1	M		Only defined values transmitted
12	Technology Ability Field values	28.2.1.2.2	M		Implementation supports a data service for each ability set in the Technology Ability Field
13	Remote Fault bit	28.2.1.2.3	M		Used in accordance with the Remote Fault function specifications
14	Acknowledge bit set, no Next Page to be sent	28.2.1.2.4	M		Set to logic one in the Link Code Word after the reception of at least three consecutive and consistent FLP Bursts
15	Acknowledge bit set, Next Page to be sent	28.2.1.2.4	NP:M		Set to logic one in the transmitted Link Code Word after the reception of at least three consecutive and consistent FLP Bursts and the current receive Link Code Word is saved
16	Number of Link Code Words sent with Acknowledge bit set	28.2.1.2.4	M		6 to 8 inclusive after COMPLETE ACKNOWLEDGE state entered
17	Device does not implement optional Next Page ability	28.2.1.2.5	M		NP=0 in base Link Code Word
18	Device implements optional Next Page ability and wishes to engage in Next Page exchange	28.2.1.2.5	NP:M		NP=1 in base Link Code Word
19	Transmit Switch function on completion of Auto-Negotiation	28.2.1.3	M		Enables the transmit path from a single technology-dependent PMA to the MDI once the highest common denominator has been selected
20	Transmit Switch function during Auto-Negotiation	28.2.1.3	M		Connects FLP Burst generator governed by Figure 28–14 to the MDI
21	Signals presented at MDI after connection through Transmit Switch from PMA	28.2.1.3	M		Conform to appropriate PHY specifications

#### 28.5.4.4 Receive function requirements

Item	Feature	Subclause	Status	Support	Value/comment
1	Timer expiration	28.2.2.1	M		Timer definition in 28.3.2, values shown in Table 28–9
2	Identification of Link Partner as Auto-Negotiation able	28.2.2.1	M		Reception of 6 to 17 (inclusive) consecutive link pulses separated by at least flp_test_min_timer time but less than flp_test_max_timer time
3	First FLP Burst identifying Link Partner as Auto-Negotiation able	28.2.2.1	M		Data recovered is discarded if FLP Burst is incomplete
4	First link pulse in an FLP Burst	28.2.2.1	M		Interpreted as a clock link pulse
5	Restart of the data_detect_min_timer and data_detect_max_timer	28.2.2.1	M		Detection of a clock link pulse (Figure 28–9)
6	Reception of logic one	28.2.2.1	M		Link pulse received between greater than data_detect_min_timer time and less than data_detect_max_timer time after a clock pulse (Figure 28–9)
7	Reception of logic zero	28.2.2.1	M		Link pulse received after greater than data_detect_max_timer time after clock pulse, is treated as clock pulse (Figure 28–9)
8	FLP Bursts separation	28.2.2.1	M		Conforms to the nlp_test_min_timer and nlp_test_max_timer timing (Figure 28–10)
9	Receive Switch function on completion of Auto-Negotiation	28.2.2.3	M		Enables the receive path from the MDI to a single technology-dependent PMA once the highest common denominator has been selected
10	Receive Switch function during Auto-Negotiation	28.2.2.3	M		Connects the MDI to the FLP and NLP receivers governed by Figures 28–15 and 28–17, and to the 100BASE-TX and 100BASE-T4 receivers if present
11	Signals presented to PMA after connection through Receive Switch from MDI	28.2.2.3	M		Conform to appropriate PHY specifications
12	Generation of ability_match, acknowledge_match, and consistency_match	28.2.2.4	M		Responsibility of Receive function in accordance with 28.3.1

**28.5.4.5 Arbitration functions**

Item	Feature	Subclause	Status	Support	Value/comment
1	MDI receive connection during Auto-Negotiation, prior to FLP detection	28.2.3.1	M		Connected to the NLP Receive Link Integrity Test state diagram, and the link integrity test functions of 100BASE-TX and/or 100BASE-T4. Not connected to the 10BASE-T or any other PMA
2	Parallel detection operational mode selection	28.2.3.1	M		Set link_control=ENABLE for the single PMA indicating link_status=READY when the autoneg_wait_timer expires
3	Parallel detection PMA control	28.2.3.1	M		Set link_control=DISABLE to all PMAs except the selected operational PMA and indicate Auto-Negotiation has completed
4	Parallel detection setting of link partner ability register	28.2.3.1	M		On transition to the FLP LINK GOOD CHECK state from the LINK STATUS CHECK state the Parallel Detection function shall set the bit in the link partner ability register (Register 5) corresponding to the technology detected by the Parallel Detection function
5	Response to renegotiation request	28.2.3.2	M		Disable all technology-dependent link integrity test functions and halt transmit activity until break_link_timer expires
6	Auto-Negotiation resumption	28.2.3.2	M		Issue FLP Bursts with base page valid in tx_link_code_word[16:1] after break_link_timer expires
7	Priority resolution	28.2.3.3	M		Single PMA connected to MDI is enabled corresponding to Technology Ability Field bit common to both Local/Link Partner Device and that has highest priority as defined by Annex 28B
8	Effect of receipt of reserved Technology Ability Field bit on priority resolution	28.2.3.3	M		Local Device ignores during priority resolution
9	Effect of parallel detection on priority resolution	28.2.3.3	M		Local Device considers technology identified by parallel detection as HCD
10	Values for HCD and link_status_[HCD] in the event there is no common technology	28.2.3.3	M		HCD=NULL link_status_[HCD]=FAIL

**28.5.4.5 Arbitration functions (continued)**

Item	Feature	Subclause	Status	Support	Value/comment
11	Message Page to Unformatted Page relationship for non-matching Selector Fields	28.2.3.4	NP:M		Each series of Unformatted Pages is preceded by an Message Page containing a message code that defines how the following Unformatted Page(s) will be interpreted
12	Message Page to Unformatted Page relationship for matching Selector Fields	28.2.3.4	NP:M		Use of Message Pages is specified by the Selector Field value
13	Transmission of Null message codes	28.2.3.4	NP:M		Sent with NP=0 on completion of all Next Pages while Link Partner continues to transmit valid Next Page information
14	Reception of Null message codes	28.2.3.4	NP:M		Recognized as indicating end of Link Partner's Next Page information
15	Next Page encoding	28.2.3.4.1	NP:M		Comply with Figures 28–11 and 28–12 for the NP, Ack, MP, Ack2, and T bits
16	Message/Unformatted Code Field	28.2.3.4.1	NP:M		D10-D0 encoded as Message Code Field if MP=1 or Unformatted Code Field if MP=0
17	NP bit encoding	28.2.3.4.2	NP:M		Logic 0=last page, logic 1=additional Next Page(s) follow
18	Message Page bit encoding	28.2.3.4.4	NP:M		Logic 0=Unformatted Page, logic 1=Message Page
19	Ack2 bit encoding	28.2.3.4.5	NP:M		Logic 0=cannot comply with message; logic 1= will comply with message
20	Toggle	28.2.3.4.6	NP:M		Takes the opposite value of the Toggle bit in the previously exchanged Link Code Word
21	Toggle encoding	28.2.3.4.6	NP:M		Logic zero = previous value of the transmitted Link Code Word equalled logic one Logic one = previous value of the transmitted Link Code Word equalled logic zero
22	Message Page encoding	28.2.3.4.7	NP:M		If MP=1, Link Code Word interpreted as Message Page
23	Message Code Field	28.2.3.4.8	NP:M		Combinations not shown in Annex 28B are reserved and may not be transmitted
24	Unformatted Page encoding	28.2.3.4.9	NP:M		If MP=0, Link Code Word interpreted as Unformatted Page

**28.5.4.5 Arbitration functions (continued)**

Item	Feature	Subclause	Status	Support	Value/comment
25	Minimum Next Page exchange	28.2.3.4.11	NP:M		If both devices indicate Next Page able, both send a minimum of one Next Page
26	Multiple Next Page exchange	28.2.3.4.11	NP:M		If both devices indicate Next Page able, exchange continues until neither Local/Remote Device has additional information; device sends Next Page with Null Message Code if it has no information to transmit
27	Unformatted Page ordering	28.2.3.4.11	NP:M		Unformatted Pages immediately follow the referencing Message Code in the order specified by the Message Code
28	Next Page Transmit register	28.2.3.4.12	NP:M		Defined in 28.2.4.1.6
29	Next Page receive data	28.2.3.4.12	NP:O		May be stored in Auto-Negotiation link partner ability register
30	Remote Fault sensing	28.2.3.5	RF:M		Optional
31	Transmission of RF bit by Local Device	28.2.3.5	M		If Local Device has no method to set RF bit, it must transmit RF bit with value of RF bit in Auto-Negotiation advertisement register (4.13)
32	RF bit reset	28.2.3.5	M		Once set, the RF bit remains set until successful renegotiation with the base Link Code Word
33	Receipt of Remote Fault indication in Base Link Code Word	28.2.3.5	MII:M		Device sets the Remote Fault bit in the MII status register (1.4) to logic one if MII is present

**28.5.4.6 Management function requirements**

Item	Feature	Subclause	Status	Support	Value/comment
1	Mandatory MII registers for Auto-Negotiation	28.2.4.1	MII:M		Registers 0, 1, 4, 5, 6
2	Optional MII register for Auto-Negotiation	28.2.4.1	MII* NP:M		Register 7
3	Auto-Negotiation enable	28.2.4.1.1	MII:M		Set control register Auto-Negotiation Enable bit (0.12)

**28.5.4.6 Management function requirements (continued)**

Item	Feature	Subclause	Status	Support	Value/comment
4	Manual Speed/Duplex settings	28.2.4.1.1	MII:M		When bit 0.12 set, control register Speed Detection (0.13) and Duplex Mode (0.8) are ignored, and the Auto-Negotiation function determines link configuration
5	Control register (Register 0) Restart Auto-Negotiation (0.9) default	28.2.4.1.1	MII:M		PHY returns value of one in 0.9 until Auto-Negotiation has been initiated
6	Control register (Register 0) Restart Auto-Negotiation (0.9) set	28.2.4.1.1	MII:M		When 0.9 set, Auto-Negotiation will (re)initiate. On completion, 0.9 will be reset by the PHY device. Writing a zero to 0.9 at any time has no effect
7	Control register (Register 0) Restart Auto-Negotiation (0.9) reset	28.2.4.1.1	MII:M		0.9 is self-clearing; writing a zero to 0.9 at any time has no effect
8	Status register (Register 1) Auto-Negotiation Complete (1.5) reset	28.2.4.1.2	MII:M		If bit 0.12 reset, or a PHY lacks the ability to perform Auto-Negotiation, (1.5) is reset
9	Status register (Register 1) Remote Fault (1.4)	28.2.4.1.2	MII:M		Set by the PHY and remains set until either the status register is read or the PHY is reset
10	Advertisement register power on default	28.2.4.1.3	MII:M		Selector field as defined in Annex 28A; Ack=0; Technology Ability Field based on MII status register (1.15:11) or logical equivalent
11	Link partner ability register read/write	28.2.4.1.4	MII:M		Read only; write has no effect
12	Link partner ability register bit definitions	28.2.4.1.4	MII:M		Direct representation of the received Link Code Word (Figure 28–7)
13	Status register (Register 1) Auto-Negotiation Complete (1.5) set	28.2.4.1.4	MII:M		Set to logic one upon successful completion of Auto-Negotiation
14	Auto-Negotiation expansion register (Register 6)	28.2.4.1.5	MII:M		Read only; write has no effect
15	Link Partner Auto-Negotiation Able bit (6.0)	28.2.4.1.5	MII:M		Set to indicate that the Link Partner is able to participate in the Auto-Negotiation function
16	Page Received bit (6.1) set	28.2.4.1.5	MII:M		Set to indicate that a new Link Code Word has been received and stored in the Auto-Negotiation link partner ability register
17	Page Received bit (6.1) reset	28.2.4.1.5	MII:M		Reset on a read of the Auto-Negotiation expansion register (Register 6)

**28.5.4.6 Management function requirements (continued)**

Item	Feature	Subclause	Status	Support	Value/comment
18	The Next Page Able bit (6.2) set	28.2.4.1.5	NP* MII:M		Set to indicate that the Local Device supports the Next Page function
19	The Link Partner Next Page Able bit (6.3) set	28.2.4.1.5	MII:M		Set to indicate that the Link Partner supports the Next Page function
20	Parallel Detection Fault bit (6.4) set	28.2.4.1.5	MII:M		Set to indicate that zero or more than one of the NLP Receive Link Integrity Test function, 100BASE-TX, or 100BASE-T4 PMAs have indicated link_status=READY when the autoneg_wait_timer expires
21	Parallel Detection Fault bit (6.4) reset	28.2.4.1.5	MII:M		Reset on a read of the Auto-Negotiation expansion register (Register 6)
22	Next Page Transmit register default	28.2.4.1.6	NP* MII:M		On power-up, contains value of 2001 H
23	Write to Next Page Transmit register	28.2.4.1.6	NP* MII:M		mr_next_page_loaded set to true
24	Absence of management function	28.2.5	NOM:M		Advertised abilities provided through a logical equivalent of mr_adv_ability[16:1]
25	Next Page support in absence of MII management	28.2.5	NOM:M		Device must provide logical equivalent of mr_np_able, mr_lp_np_able, or mr_next_page_loaded variables in order to set NP bit in transmitted Link Code Word

**28.5.4.7 Technology-dependent interface**

Item	Feature	Subclause	Status	Support	Value/comment
1	PMA_LINK.indicate(link_status) values	28.2.6.1.1	M		link_status set to READY, OK or FAIL
2	PMA_LINK.indicate(link_status) generation	28.2.6.1.2	M		Technology-dependent PMA and NLP Receive Link Integrity Test state diagram (Figure 28–17) responsibility
3	PMA_LINK.indicate(link_status), effect of receipt	28.2.6.1.3	M		Governed by the state diagram of Figure 28–16
4	PMA_LINK.request(link_control) values	28.2.6.1.3	M		link_control set to SCAN_FOR_CARRIER, DISABLE, or ENABLE



**28.5.4.7 Technology-dependent interface (continued)**

Item	Feature	Subclause	Status	Support	Value/comment
5	Effect of link_control=SCAN_FOR_CARRIER	28.2.6.2.1	M		PMA to search for carrier and report link_status=READY when carrier is received, but no other actions are enabled
6	Effect of link_control=DISABLE	28.2.6.2.1	M		Disables PMA processing
7	Effect of link_control=ENABLE	28.2.6.2.1	M		Control passed to a single PMA for normal processing functions
8	PMA_LINK.request(link_control) generation	28.2.6.2.2	M		Auto-Negotiation function responsibility in accordance with Figures 28–15 and 28–16
9	PMA_LINK.request(link_control) default upon power-on, reset, or release from power-down	28.2.6.2.2	M		link_control = DISABLE state to all technology-dependent PMAs
10	PMA_LINK.request(link_control) effect of receipt	28.2.6.2.3	M		Governed by Figure 28–17 and the receiving technology-dependent link integrity test function
11	The linkpulse parameter shall	28.2.6.3.1	M	Yes [ ]	TRUE or FALSE.
12	The linkpulse=FALSE shall be used	28.2.6.3.1	M	Yes [ ]	By the Auto-Negotiation function to indicate that the Receive State Diagram has performed a state transition.
13	The linkpulse=TRUE shall be used	28.2.6.3.1	M	Yes [ ]	By the Auto-Negotiation function to indicate that a valid Link Pulse has been received.
14	The Auto-Negotiation function shall generate linkpulse	28.2.6.3.2	M	Yes [ ]	To indicate to the PHY how to respond, in accordance with the state diagram of Figure 28–15.
15	Upon power-on or reset, if Auto-Negotiation is enabled (mr_autoneg_enable=true) the PMA_LINKPULSE.request(FALSE) message shall be	28.2.6.3.2	M	Yes [ ]	Issued to all technology-dependent PMAs.
16	The effect of the receipt of linkpulse shall be governed	28.2.6.3.3	M	Yes [ ]	By the receiving technology-dependent PMA function, based on the intent specified in the primitive semantics.

**28.5.4.8 State diagrams**

Item	Feature	Subclause	Status	Support	Value/comment
1	Adherence to state diagrams	28.3	M		Implement all features of Figures 28–14 to 28–17. Identified options to Figures 28–14 to 28–17 are permitted
3	Ambiguous requirements	28.3	M		State diagrams take precedence in defining functional operation
4	autoneg_wait_timer	28.3.1	M		Expires between 500–1000 ms after being started
5	break_link_timer	28.3.2	M		Expires between 1200–1500 ms after being started
6	data_detect_min_timer	28.3.2	M		Expires between 15–47 $\mu$ s from the last clock pulse
7	data_detect_max_timer	28.3.2	M		Expire between 78–100 $\mu$ s from the last clock pulse
8	flp_test_max_timer	28.3.2	M		Expires between 165–185 $\mu$ s from the last link pulse
9	flp_test_min_timer	28.3.2	M		Expires between 5–25 $\mu$ s from the last link pulse
10	interval_timer	28.3.2	M		Expires 55.5–69.5 $\mu$ s from each clock pulse and data bit
11	link_fail_inhibit_timer	28.3.2	M		Expires 750–1000 ms after entering the FLP LINK GOOD CHECK state
12	nlp_test_max_timer	28.3.2	M		Expires between 50–150 ms after being started if not restarted
13	nlp_test_min_timer	28.3.2	M		Expires between 5–7 ms after being started if not restarted
14	transmit_link_burst_timer	28.3.1	M		Expires 5.7–22.3 ms after the last transmitted link pulse in an FLP Burst

**28.5.4.9 Electrical characteristics**

Item	Feature	Subclause	Status	Support	Value/comment
1	Pulses within FLP Bursts	28.4	M		Identical to the characteristics of NLPs and meet the requirements of Figure 14–12

**28.5.4.10 Auto-Negotiation annexes**

Item	Feature	Annex	Status	Support	Value/comment
1	Selector field, S[4:0] values in the Link Code Word	Annex 28A	M		Identifies base message type as defined by Table 28A–1
2	Selector field reserved combinations	Annex 28A	M		Transmission not permitted
3	Relative priorities of the technologies supported by the IEEE 802.3 Selector Field value	28B.3	M		Defined in Annex 28B.3
4	Relative order of the technologies supported by IEEE 802.3 Selector Field	28B.3	M		Remain unchanged
5	Addition of new technology	28B.3	M		Inserted into its appropriate place in the priority resolution hierarchy, shifting technologies of lesser priority lower in priority
6	Addition of vendor-specific technology	28B.3	M		Priority of IEEE 802.3 <sup>®</sup> standard topologies maintained, vendor-specific technologies to be inserted into an appropriate location
7	Message Code Field	Annex 28C	NP:M		Defines how following Unformatted Pages (if applicable) are interpreted
8	Message Code Field reserved combinations	Annex 28C	NP:M		Transmission not permitted
9	Auto-Negotiation reserved code 1	28C.1	NP:M		Transmission of M10 to M0 equals 0, not permitted
10	Null Message Code	28C.2	NP:M		Transmitted during Next Page exchange when the Local Device has no information to transmit and Link Partner has additional pages to transmit
11	Remote Fault Identifier Message Code	28C.5	NP:M		Followed by single Unformatted Page to identify fault type with types defined in 28C.5

**28.5.4.10 Auto-Negotiation annexes (continued)**

Item	Feature	Annex	Status	Support	Value/comment
12	Organizationally Unique Identifier Message Code	28C.6	NP:M		Followed by 4 Unformatted Pages. First Unformatted Page contains most significant 11 bits of OUI (bits 23:13) with MSB in U10; Second Unformatted Page contains next most significant 11 bits of OUI (bits 12:2), with MSB in U10; Third Unformatted Page contains the least significant 2 bits of OUI (bits 1:0) with MSB in U10, bits U8:0 contains user-defined code specific to OUI; Fourth Unformatted Page contains user-defined code specific to OUI
13	PHY Identifier Message Code	28C.7	NP:M		Followed by 4 Unformatted Pages. First Unformatted Page contains most significant 11 bits of PHY ID (2.15:5) with MSB in U10; Second Unformatted Page contains PHY ID bits 2.4:0 to 3.15:10, with MSB in U10; Third Unformatted Page contains PHY ID bits 3.9:0, with MSB in U10, and U0 contains user-defined code specific to PHY ID; Fourth Unformatted Page contains user-defined code specific to PHY ID
14	Auto-Negotiation reserved code 2	28C.8	NP:M		Transmission of M10 to M0 equals 1, not permitted

**28.6 Auto-Negotiation expansion**

Auto-Negotiation is designed in a way that allows it to be easily expanded as new technologies are developed. When a new technology is developed, the following things must be done to allow Auto-Negotiation to support it:

- a) The appropriate Selector Field value to contain the new technology must be selected and allocated.
- b) A Technology bit must be allocated for the new technology within the chosen Selector Field value.
- c) The new technology's relative priority within the technologies supported within a Selector Field value must be established.

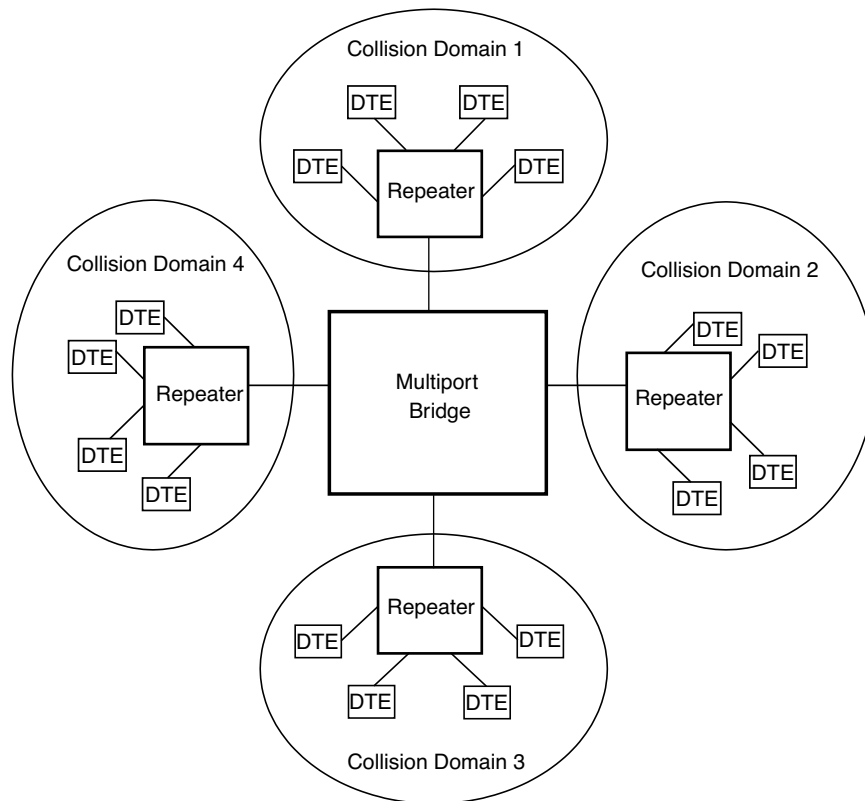
Code space allocations are enumerated in Annex 28A, Annex 28B, and Annex 28C. Additions and insertions to the annexes are allowed. No changes to existing bits already defined are allowed.

## 29. System considerations for multisegment 100BASE-T networks

### 29.1 Overview

This clause provides information on building 100BASE-T networks. The 100BASE-T technology is designed to be deployed in both homogenous 100 Mb/s networks and heterogeneous 10/100 Mb/s mixed CSMA/CD networks. Network topologies can be developed within a single 100BASE-T collision domain, but maximum flexibility is achieved by designing multiple collision domain networks that are joined by bridges and/or routers configured to provide a range of service levels to DTEs. For example, a combined 100BASE-T/10BASE-T system built with repeaters and bridges can deliver dedicated 100 Mb/s, shared 100 Mb/s, dedicated 10 Mb/s, and shared 10 Mb/s service to DTEs. The effective bandwidth of shared services is controlled by the number of DTEs that share the service.

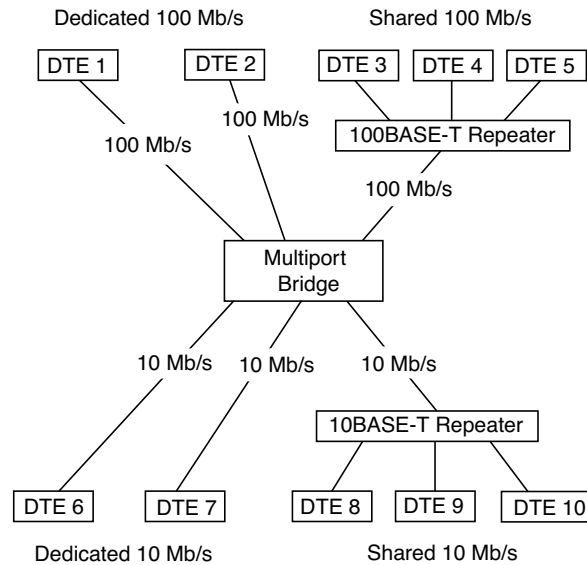
Linking multiple 100BASE-T collision domains with bridges maximizes flexibility. Bridged topology designs can provide single bandwidth (Figure 29–1) or multiple bandwidth (Figure 29–2) services.



**Figure 29–1 — 100 Mb/s multiple collision domain topology using multiport bridge**

Individual collision domains can be linked by single devices (as shown in Figure 29–1 and Figure 29–2) or by multiple devices from any of several transmission systems. The design of multiple-collision-domain networks is governed by the rules defining each of the transmission systems incorporated into the design.

The design of shared bandwidth 10 Mb/s collision domains is defined in 13.1 through 13.4; the design of shared bandwidth 100 Mb/s CSMA/CD collision domains is defined in 29.1.1 through 29.3.1.2. The design of 10BASE full duplex LANs is defined in 13.5; the design of full duplex 100BASE-X LANs is defined in 29.4.



**Figure 29–2—Multiple bandwidth, multiple collision domain topology using multiport bridge**

### 29.1.1 Single collision domain multisegment networks

This clause provides information on building 100 Mb/s CSMA/CD multisegment networks within a single collision domain. The proper operation of a CSMA/CD network requires the physical size and number of repeaters to be limited in order to meet the round-trip propagation delay requirements of 4.2.3.2.3 and 4.4.2.1 and IPG requirements specified in 4.4.2.1.

This clause provides two network models. Transmission System Model 1 is a set of configurations that have been validated under conservative rules and have been qualified as meeting the requirements set forth above. Transmission System Model 2 is a set of calculation aids that allow those configuring a network to test a proposed configuration against a simple set of criteria that allows it to be qualified. Transmission System Model 2 validates an additional broad set of topologies that are fully functional and do not fit within the simpler, but more restrictive rules of Model 1.

The physical size of a CSMA/CD network is limited by the characteristics of individual network components. These characteristics include the following:

- a) Media lengths and their associated propagation time delay
- b) Delay of repeater units (start-up, steady-state, and end of event)
- c) Delay of MAUs and PHYs (start-up, steady-state, and end of event)
- d) Interpacket gap shrinkage due to repeater units
- e) Delays within the DTE associated with the CSMA/CD access method
- f) Collision detect and deassertion times associated with the MAUs and PHYs

Table 29–1 summarizes the delays for 100BASE-T media segments. For more detailed information on the delays associated with individual 100BASE-T components, see

MII: Annex 22A  
 100BASE-T2: 32.12  
 100BASE-T4: 23.11

100BASE-TX: 24.6  
 100BASE-FX: ISO/IEC 9314-3: 1990  
 Repeater: 27.3

**Table 29–1—Delays for network media segments Model 1**

Media type	Maximum number of PHYs per segment	Maximum segment length (m)	Maximum medium round-trip delay per segment (BT)
Balanced cable Link Segment 100BASE-T	2	100	114
Fiber Link Segment	2	412	412

### 29.1.2 Repeater usage

Repeaters are the means used to connect segments of a network medium together into a single collision domain. Different signaling systems (i.e., 100BASE-T2, 100BASE-T4, 100BASE-TX, 100BASE-FX) can be joined into a common collision domain using repeaters. Bridges can also be used to connect different signaling systems; however, if a bridge is so used, each system connected to the bridge will be a separate collision domain.

Two types of repeaters are defined for 100BASE-T (see Clause 27). Class I repeaters are principally used to connect unlike physical signaling systems and have internal delays such that only one Class I repeater can reside within a single collision domain when maximum cable lengths are used (see Figure 29–4). Class II repeaters typically provide ports for only one physical signaling system type (e.g., 100BASE-TX but not 100BASE-T4) and have smaller internal delays so that two such repeaters may reside within a given collision domain when maximum cable lengths are used (see Figure 29–6). Cable length can be sacrificed to add additional repeaters in a collision domain (see 29.3).

### 29.2 Transmission System Model 1

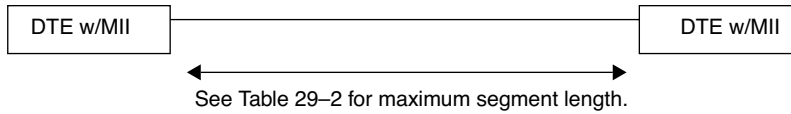
The following network topology constraints apply to networks using Transmission System Model 1.

- a) All balanced cable (copper) segments less than or equal to 100 m each.
- b) Fiber segments less than or equal to 412 m each.
- c) MII cables for 100BASE-T shall not exceed 0.5 m each. When evaluating system topology, MII cable delays need not be accounted for separately. Delays attributable to the MII are incorporated into DTE and repeater component delays.

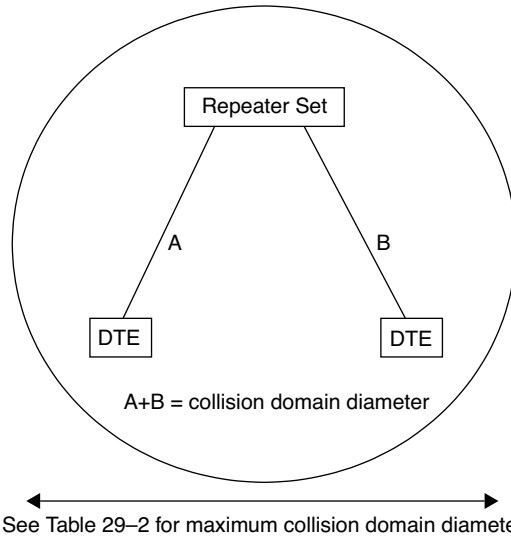
### 29.3 Transmission System Model 2

The physical size and number of topological elements in a 100BASE-T network is limited primarily by round-trip collision delay. A network configuration must be validated against collision delay using a network model. Since there are a limited number of topology models for any 100BASE-T collision domain, the modeling process is quite straightforward and can easily be done either manually or with a spreadsheet.

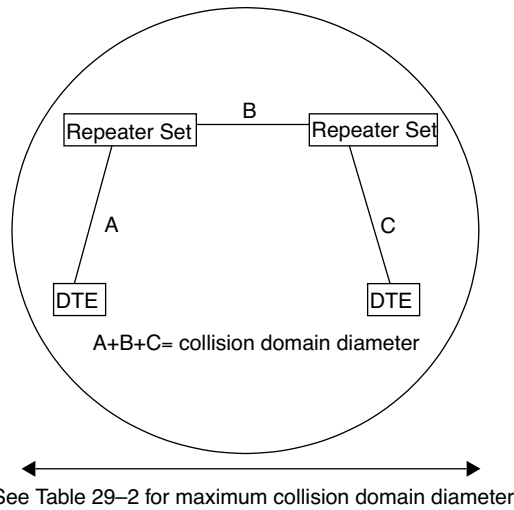
The model proposed here is derived from the one presented in 13.4. Modifications have been made to accommodate adjustments for DTE, repeater, and cable speeds.



**Figure 29-3—Model 1: Two DTEs, no repeater**



**Figure 29-4—Model 1: Single repeater**



**Figure 29-5—System Model 1: Two Class II repeaters**



**Table 29–2—Maximum Model 1 collision domain diameter<sup>a</sup>**

Model	Balanced cabling (copper)	Fiber	Balanced cabling & fiber (T2 or T4 and FX)	Balanced cabling & fiber (TX and FX)
DTE-DTE (see Figure 29–3)	100	412	NA	NA
One Class I repeater (see Figure 29–4)	200	272	231 <sup>a</sup>	260.8 <sup>a</sup>
One Class II repeater (see Figure 29–4)	200	320	304 <sup>b</sup>	308.8 <sup>b</sup>
Two Class II repeaters (see Figure 29–5)	205	228	236.3 <sup>c</sup>	216.2 <sup>c</sup>

<sup>a</sup>Assumes 100 m of balanced cabling and one fiber link.

<sup>b</sup>This entry included for completeness. It may be impractical to construct a T4 to FX class II repeater.

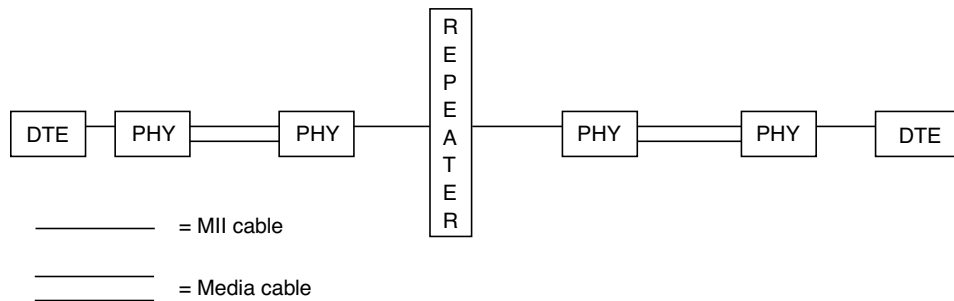
<sup>c</sup>Assumes 105 m of balanced cabling and one fiber link.

**29.3.1 Round-trip collision delay**

For a network to be valid, it must be possible for any two DTEs on the network to contend for the network at the same time. Each station attempting to transmit must be notified of the contention by the returned “collision” signal within the “collision window” (see 4.1.2.2 and 5.2.2.1.2). Additionally, the maximum length fragment created must contain less than 512 bits after the start-of-frame delimiter (SFD). These requirements limit the physical diameter (maximum distance between DTEs) of a network. The maximum round-trip delay must be qualified between all pairs of DTEs in the network. In practice this means that the qualification must be done between those that, by inspection of the topology, are candidates for the longest delay. The following network modeling methodology is provided to assist that calculation.

**29.3.1.1 Worst-case path delay value (PDV) selection**

The worst-case path through a network to be validated shall be identified by examination of aggregate DTE delays, cable delays, and repeater delays. The worst case consists of the path between the two DTEs at opposite ends of the network that have the longest round-trip time. Figure 29–6 and Figure 29–7 show schematic representatins of one-repeater and two-repeater paths.

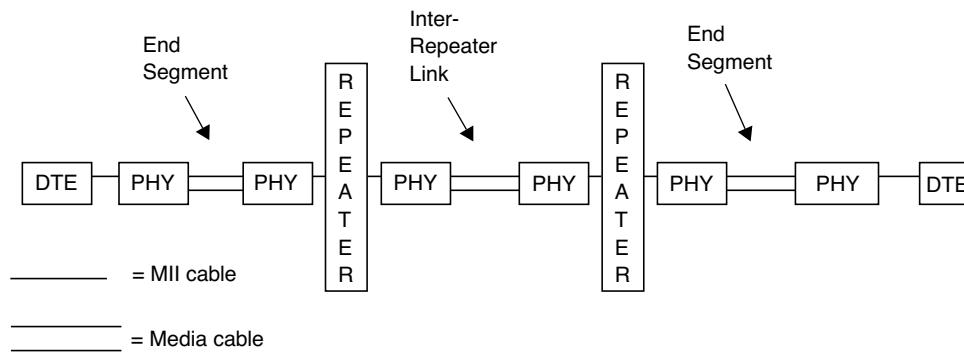


**Figure 29–6—System Model 2: Single repeater**

**29.3.1.2 Worst-case PDV calculation**

Once a set of paths is chosen for calculation, each shall be checked for validity against the following formula:

$$PDV = \sum \text{link delays (LSDV)} + \sum \text{repeater delays} + \text{DTE delays} + \text{safety margin}$$



**Figure 29-7—System Model 2-2: Two repeaters**

Values for the formula variables are determined by the following method:

- a) Determine the delay for each link segment (Link Segment Delay Value, or LSDV), including inter-repeater links, using the formula

$$\text{LSDV} = 2 \text{ (for round-trip delay)} \times \text{segment length} \times \text{cable delay for this segment}$$

NOTE 1—Length is the sum of the cable lengths between the PHY interfaces at the repeater and the farthest DTE for End Segments plus the sum of the cable lengths between the repeater PHY interfaces for Inter-Repeater Links. All measurements are in meters.

NOTE 2—Cable delay is the delay specified by the manufacturer or the maximum value for the type of cable used as shown in Table 29-3. For this calculation, cable delay must be specified in bit times per meter (BT/m). Table 29-4 can be used to convert values specified relative to the speed of light (%c) or nanoseconds per meter (ns/m).

NOTE 3—When actual cable lengths or propagation delays are not known, use the Max delay in bit times as specified in Table 29-3 for copper cables. Delays for fiber should be calculated, as the value found in Table 29-3 will be too large for most applications.

- b) Sum together the LSDVs for all segments in the path.
- c) Determine the delay for each repeater in the path. If model-specific data are not available from the manufacturer, determine the class of each repeater (I or II) and enter the appropriate default value from Table 29-3.
- d) MII cables for 100BASE-T shall not exceed 0.5 m each. When evaluating system topology, MII cable delays need not be accounted for separately. Delays attributable to the MII are incorporated into DTE and repeater component delays.
- e) Use the DTE delay value shown in Table 29-3 unless your equipment manufacturer defines a different value.
- f) Decide on appropriate safety margin—0 to 5 bit times—for the PDV calculation. Safety margin is used to provide additional margin to accommodate unanticipated delay elements, such as extra-long connecting cable runs between wall jacks and DTEs. (A safety margin of 4 BT is recommended.)
- g) Insert the values obtained through the calculations above into the following formula to calculate the PDV. (Some configurations may not use all the elements of the formula.)

$$\text{PDV} = \sum \text{link delays (LSDV)} + \sum \text{repeater delays} + \text{DTE delay} + \text{safety margin}$$

- h) If the PDV is less than 512, the path is qualified in terms of worst-case delay.
- i) Late collisions and/or CRC errors are indicators that path delays exceed 512 BT.

**Table 29–3—Network component delays, Transmission System Model 2**

Component	Round trip delay in bit times per meter	Maximum round trip delay in bit times
Two TX/FX DTEs		100
Two T4 DTEs		138
Two T2 DTEs		96
One T2 or T4 and one TX/FX DTE <sup>a</sup>		127
Cat 3 cabling segment	1.14	114 (100 m)
Cat 4 cabling segment	1.14	114 (100 m)
Cat 5 cabling segment	1.112	111.2 (100 m)
STP cabling segment	1.112	111.2 (100 m)
Fiber optic cabling segment	1.0	412 (412 m)
Class I repeater		140
Class II repeater with all ports TX/FX		92
Class II repeater with any port T4		67
Class II repeater with any port T2		90

<sup>a</sup> Worst-case values are used (TX/FX values for MAC transmit start and MDI input to collision detect; T4 value for MDI input to MDI output).

**Table 29–4—Conversion table for cable delays**

Speed relative to c	ns/m	BT/m
0.4	8.34	0.834
0.5	6.67	0.667
0.51	6.54	0.654
0.52	6.41	0.641
0.53	6.29	0.629
0.54	6.18	0.618
0.55	6.06	0.606
0.56	5.96	0.596
0.57	5.85	0.585
0.58	5.75	0.575
0.5852	5.70	0.570
0.59	5.65	0.565
0.6	5.56	0.556
0.61	5.47	0.547
0.62	5.38	0.538

**Table 29–4—Conversion table for cable delays (continued)**

Speed relative to c	ns/m	BT/m
0.63	5.29	0.529
0.64	5.21	0.521
0.65	5.13	0.513
0.654	5.10	0.510
0.66	5.05	0.505
0.666	5.01	0.501
0.67	4.98	0.498
0.68	4.91	0.491
0.69	4.83	0.483
0.7	4.77	0.477
0.8	4.17	0.417
0.9	3.71	0.371

#### 29.4 Full duplex 100 Mb/s topology limitations

Unlike half duplex CSMA/CD networks, the physical size of full duplex 100 Mb/s CSMA/CD networks is not limited by the round-trip collision propagation delay. Instead, the maximum link length between DTEs is limited only by the signal transmission characteristics of the specific cable, as specified in Table 29–5.

**Table 29–5—Link segment length limits; 100 Mb/s full duplex segments**

Cable Type	Maximum link segment length
100BASE-TX (UTP, STP per Clause 25)	<b>100 m</b>
100BASE-FX (multimode fiber per Clause 26)	<b>2000 m</b>
100BASE-T2 (UTP per Clause 32)	<b>100 m</b>

## 30. 10 Mb/s, 100 Mb/s, 1000 Mb/s, MAC Control, and Link Aggregation Management

### 30.1 Overview

This clause provides the Layer Management specification for DTEs, repeaters, and MAUs based on the CSMA/CD access method. The clause is produced from the ISO framework additions to Clause 5, Layer Management; Clause 19, Repeater Management; and Clause 20, MAU Management. It incorporates additions to the objects, attributes, and behaviors to support 100 and 1000 Mb/s CSMA/CD, full duplex operation, MAC Control, and Link Aggregation.

The layout of this clause takes the same form as 5.1, 5.2, and Clauses 19 and 20, although with equivalent subclauses grouped together. It identifies a common management model and framework applicable to IEEE 802.3<sup>®</sup> managed elements, identifies those elements and defines their managed objects, attributes, and behaviours in a protocol-independent language. It also includes a formal GDMO definition of the protocol encodings for CMIP and ISO/IEC 15802-2: 1995 [ANSI/IEEE Std 802.1B<sup>™</sup> and 802.1k<sup>™</sup>, 1995 Edition].

NOTE—The arcs (that is, object identifier values) defined in Annex 30A, the formal GDMO definitions, deprecate the arcs previously defined in Annexes H.1 (Layer Management), H.2 (Repeater Management), and H.3 (MAU Management). See IEEE Std 802.1F-1993<sup>™</sup>, Annex C.4.

This clause provides the Layer Management specification for DTEs, repeaters, and MAUs based on the CSMA/CD access method. It defines facilities comprised of a set of statistics and actions needed to provide IEEE 802.3<sup>®</sup> Management services. The information in this clause should be used in conjunction with the Procedural Model defined in 4.2.7 through 4.2.10. The Procedural Model provides a formal description of the relationship between the CSMA/CD Layer Entities and the Layer Management facilities.

This management specification has been developed in accordance with the OSI management architecture as specified in the ISO Management Framework document, ISO/IEC 7498-4: 1989. It is independent of any particular management application or management protocol.

The management facilities defined in this standard may be accessed both locally and remotely. Thus, the Layer Management specification provides facilities that can be accessed from within a station or can be accessed remotely by means of a peer-management protocol operating between application entities.

In CSMA/CD no peer management facilities are necessary for initiating or terminating normal protocol operations or for handling abnormal protocol conditions. Since these activities are subsumed by the normal operation of the protocol, they are not considered to be a function of Layer Management and are, therefore, not discussed in this clause.

Implementation of part or all of 10 Mb/s, 100 Mb/s and 1000 Mb/s Management is not a requirement for conformance to Clauses 4, 7, 9, 22, 23, 24, 25, 26, 27, 28, 31, 32, 35, 36, 37, 38, 39, 40, and 41.

The intent of this standard is to furnish a management specification that can be used by the wide variety of different devices that may be attached to a network specified by ISO/IEC 8802-3. Thus, a comprehensive list of management facilities is provided.

The improper use of some of the facilities described in this clause may cause serious disruption of the network. In accordance with ISO management architecture, any necessary security provisions should be provided by the Agent in the Local System Environment. This can be in the form of specific security features or in the form of security features provided by the peer communication facilities.

### 30.1.1 Scope

This clause includes selections from Clauses 5, 19, and 20. It is intended to be an entirely equivalent specification for the management of 10 Mb/s DTEs, 10 Mb/s baseband repeater units, and 10 Mb/s integrated MAUs. It also includes the additions for management of full duplex operation, MAC Control, 100 Mb/s and 1000 Mb/s DTEs and repeater units, embedded MAUs, and external PHYs connected with the MII or GMII. Implementations of management for 10 Mb/s DTEs, repeater units, and embedded MAUs should follow the requirements of this clause (e.g., a 10 Mb/s implementation should incorporate the attributes to indicate that it is not capable of 100 or 1000 Mb/s operation, a half duplex DTE should incorporate the attributes to indicate that it is not capable of full duplex operation, etc.).

This clause defines a set of mechanisms that enable management of ISO/IEC 8802-3 10 Mb/s, 100 Mb/s and 1000 Mb/s DTEs, baseband repeater units, and integrated Medium Attachment Units (MAUs). In addition, for ports without integral MAUs, attributes are provided for characteristics observable from the AUI of the connected DTE or repeater. Direct management of AUI MAUs that are external to their respective DTEs or repeaters is beyond the scope of this standard. The managed objects within this standard are defined in terms of their behaviour, attributes, actions, notifications, and packages in accordance with IEEE 802.1<sup>™</sup> and ISO standards for network management. Managed objects are grouped into mandatory and optional packages.

This specification is defined to be independent of any particular management application or management protocol. The means by which the managed objects defined in this standard are accessed is beyond the scope of this standard.

### 30.1.2 Relationship to objects in IEEE 802.1F<sup>®</sup>

The following managed object classes, if supported by an implementation, shall be as specified in IEEE Std 802.1F-1993<sup>™</sup>: ResourceTypeID, EWMAMetricMonitor.

#### **oResourceTypeID**

This object class is mandatory and shall be implemented as defined in IEEE 802.1F<sup>®</sup>. This object is bound to oMAC-Entity, oRepeater, and oMAU as defined by the NAMEBINDINGS in 30A.8.1. Note that the binding to oMAU is mandatory only when MII is present. The Entity Relationship Diagram, Figure 30–3, shows these bindings pictorially.

#### **oEWMAMetricMonitor**

This object class is optional. When implemented, it shall be implemented as defined in IEEE 802.1F<sup>®</sup>, subject to the specific requirements described below. This object is bound to system as defined by the NAMEBINDINGS in 30A.1.1, 30A.3.1, and 30A.2.1.

Implementations of IEEE 802.3<sup>®</sup> Management that support the oEWMAMetricMonitor managed object class are required to support values of granularity period as small as one second. Implementations are required to support at least one sequence of low and high thresholds. The granularity period may be set to equal to the moving time period as a minimal conformant implementation.

### 30.1.3 Systems management overview

Within the ISO Open Systems Interconnection (OSI) architecture, the need to handle the special problems of initializing, terminating, and monitoring ongoing activities and assisting in their operations, as well as handling abnormal conditions, is recognized. These needs are collectively addressed by the systems management component of the OSI architecture.

A management protocol is required for the exchange of information between systems on a network. This management standard is independent of any particular management protocol.

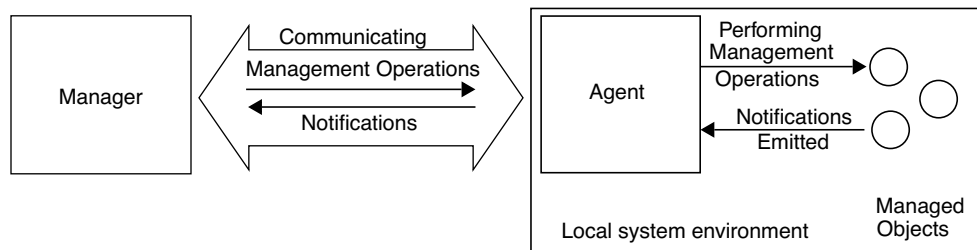
This management standard, in conjunction with the management standards of other layers, provides the means to perform various management functions. IEEE 802.3<sup>®</sup> Management collects information needed from the MAC and Physical Layers and the devices defined in IEEE 802.3<sup>®</sup>. It also provides a means to exercise control over those elements.

The relationship between the various management entities and the layer entities according to the ISO model is shown in Figure 30–1.

### 30.1.4 Management model

This standard describes management of DTEs, repeaters, and integrated MAUs in terms of a general model of management of resources within the open systems environment. The model, which is described in ISO/IEC 10040: 1992, is briefly summarized here.

Management is viewed as a distributed application modeled as a set of interacting management processes. These processes are executed by systems within the open environment. A managing system executes a managing process that invokes management operations. A managed system executes a process that is receptive to these management operations and provides an interface to the resources to be managed. A managed object is the abstraction of a resource that represents its properties as seen by (and for the purpose of) management. Managed objects respond to a defined set of management operations. Managed objects are also capable of emitting a defined set of notifications. This interaction of processes is shown in Figure 30–1.



NOTE—This figure is drawn from Figure 1 of ISO/IEC 10040: 1992, Information technology—Open Systems Interconnection—Systems management overview. In the event of any conflict, the depiction in ISO/IEC 10040: 1992 takes precedence.

**Figure 30–1—Interaction between manager, agent, and objects**

A managed object is a management view of a resource. The resource may be a logical construct, function, physical device, or anything subject to management. Managed objects are defined in terms of four types of elements:

- a) *Attributes*. Data-like properties (as seen by management) of a managed object.
- b) *Actions*. Operations that a managing process may perform on an object or its attributes.
- c) *Notifications*. Unsolicited reports of events that may be generated by an object.
- d) *Behaviour*. The way in which managed objects, attributes, and actions interact with the actual resources they model and with each other.

The above items are defined in 30.3, 30.4, 30.5, and 30.6 of this clause in terms of the template requirements of ISO/IEC 10165-4: 1991.

Some of the functions and resources within 802.3<sup>®</sup> devices are appropriate targets for management. They have been identified by specifying managed objects that provide a management view of the functions or resources. Within this general model, the 802.3<sup>®</sup> device is viewed as a managed device. It performs functions as defined by the applicable standard for such a device. Managed objects providing a view of those functions and resources appropriate to the management of the device are specified. The purpose of this standard is to define the object classes associated with the devices in terms of their attributes, operations, notifications, and behaviour.

## 30.2 Managed objects

### 30.2.1 Introduction

This clause identifies the Managed Object classes for IEEE 802.3<sup>®</sup> components within a managed system. It also identifies which managed objects and packages are applicable to which components.

All counters defined in this specification are assumed to be wrap-around counters. Wrap-around counters are those that automatically go from their maximum value (or final value) to zero and continue to operate. These unsigned counters do not provide for any explicit means to return them to their minimum (zero), i.e., reset. Because of their nature, wrap-around counters should be read frequently enough to avoid loss of information. Counters in 30.3, 30.4, 30.5, and 30.6 that have maximum increment rates specified for 10 Mb/s operation, and are appropriate to 100 Mb/s operation, have ten times the stated maximum increment rate for 100 Mb/s operation unless otherwise indicated. Counters that are appropriate to 1000 Mb/s operation have one hundred times the stated maximum increment rate for 1000 Mb/s operation unless otherwise indicated.

### 30.2.2 Overview of managed objects

Managed objects provide a means to

- Identify a resource
- Control a resource
- Monitor a resource

#### 30.2.2.1 Text description of managed objects

In case of conflict, the formal behavior definitions in 30.3, 30.4, 30.5, 30.6, and 30.7 take precedence over the text descriptions in this subclause.

**oAggregator**                      If implemented, oAggregator is the top-most managed object class of the DTE portion of the containment tree shown in Figure 30–3. Note that this managed object class may be contained within another superior managed object class. Such containment is expected, but is outside the scope of this International Standard. The oAggregator managed object class provides the management controls necessary to allow an instance of an Aggregator to be managed.

**oAggregationPort**                If oAggregator is implemented, oAggregationPort is contained within oAggregator. An instance of this managed object class is present for each Aggregation Port that is part of the aggregation represented by the oAggregator instance. This managed object class provides the basic management controls necessary to allow an instance of an Aggregation Port to be managed, for the purposes of Link Aggregation.



- oAggPortStats** If oAggregator is implemented, a single instance of oAggPortStats may be contained within oAggregationPort. This managed object class provides optional additional statistics related to LACP and Marker protocol activity on an instance of an Aggregation Port that is involved in Link Aggregation.
- oAggPortDebugInformation** If oAggregator is implemented, a single instance of oAggPortDebugInformation may be contained within oAggregationPort. This managed object class provides optional additional information that can assist with debugging and fault finding in Systems that support Link Aggregation.
- oMACControlEntity** If implemented, and if oAggregator is implemented, oMACControlEntity is contained within oAggregator. Otherwise, if implemented, oMACControlEntity becomes the top-most managed object class of the DTE portion of the containment tree shown in Figure 30–3. Note that this managed object class may be contained within another superior managed object class. Such containment is expected, but is outside the scope of this International Standard.
- oMACControlFunctionEntity** Contained within oMACControlEntity. Each function defined and implemented within the MAC Control sublayer has an associated oMACControlFunctionEntity for the purpose of managing that function.
- oMACEntity** If oMACControlEntity is implemented, oMACEntity is contained within oMACControlEntity. Otherwise, if oAggregator is implemented, oMACEntity is contained within oAggregator. Otherwise, oMACEntity becomes the top-most managed object class of the DTE portion of the containment tree shown in Figure 30–3. Note that this managed object class may be contained within another superior managed object class. Such containment is expected, but is outside the scope of this International Standard.
- oPHYEntity** Contained within oMACEntity. Many instances of oPHYEntity may coexist within one instance of oMACEntity; however, only one PHY may be active for data transfer to and from the MAC at any one time. oPHYEntity is the managed object that contains the MAU managed object in a DTE.
- oRepeater** The top-most managed object class of the repeater portion of the containment tree shown in Figure 30–3. Note that this managed object class may be contained within another superior managed object class. Such containment is expected, but is outside the scope of this standard.
- oRepeaterMonitor** A managed object class called out by IEEE Std 802.1F-1993<sup>™</sup>. See 30.1.2, oEWMAMetricMonitor.
- oGroup** The group managed object class is a view of a collection of repeater ports.

**oRepeaterPort**

The repeater port managed object class provides a view of the functional link between the data transfer service and a single PMA. The attributes associated with repeater port deal with the monitoring of traffic being handled by the repeater from the port and control of the operation of the port. The Port Enable/Disable function as reported by portAdminState is preserved across events involving loss of power. The oRepeaterPort managed object contains the MAU managed object in a repeater set.

NOTE—Attachment to nonstandard PMAs is outside the scope of this standard.

**oMAU**

The managed object of that portion of the containment tree shown in Figure 30–3. The attributes, notifications, and actions defined in this subclause are contained within the MAU managed object. Neither counter values nor the value of MAUAdminState is required to be preserved across events involving the loss of power.

**oAutoNegotiation**

The managed object of that portion of the containment tree shown in Figure 30–3. The attributes, notifications, and actions defined in this subclause are contained within the MAU managed object.

**oResourceTypeID**

A managed object class called out by IEEE Std 802.1F-1993<sup>™</sup>. It is used within this clause to identify manufacturer, product, and revision of managed components that implement functions and interfaces defined within IEEE 802.3<sup>®</sup>. The Clause 22 MII or Clause 35 GMII specifies two registers to carry PHY Identifier (22.2.4.3.1), which provides succinct information sufficient to support oResourceTypeID.

**30.2.2.2 Functions to support management**

Functions are defined in Clauses 5, 7, 22, 23, 24, 25, 26, 27, 28, 31, 32, 35, 36, 37, 38, 39, 40, and 41 both to facilitate unmanaged operation and managed operation. The functions in these clauses that facilitate managed operation are referenced from the text of this management clause.

**30.2.2.2.1 DTE MAC sublayer functions**

For DTE MACs, with regard to reception-related error statistics a hierarchical order has been established such that when multiple error statuses can be associated with one frame, only one status is returned to the LLC. This hierarchy in descending order is as follows:

- frameTooLong
- alignmentError
- frameCheckError
- lengthError

The counters are primarily incremented based on the status returned to the MAC client; therefore, the hierarchical order of the counters is determined by the order of the status. Frame fragments are not included in any of the statistics unless otherwise stated. In implementing any of the specified actions, receptions and transmissions that are in progress are completed before the action takes effect.

### 30.2.2.2.2 Repeater functions

The Repeater Port Object class contains seven functions which are defined in this clause and are used to collect statistics on the activity received by the port. The relationship of the functions to the repeater port and to the port attributes is shown in Figure 30–2.

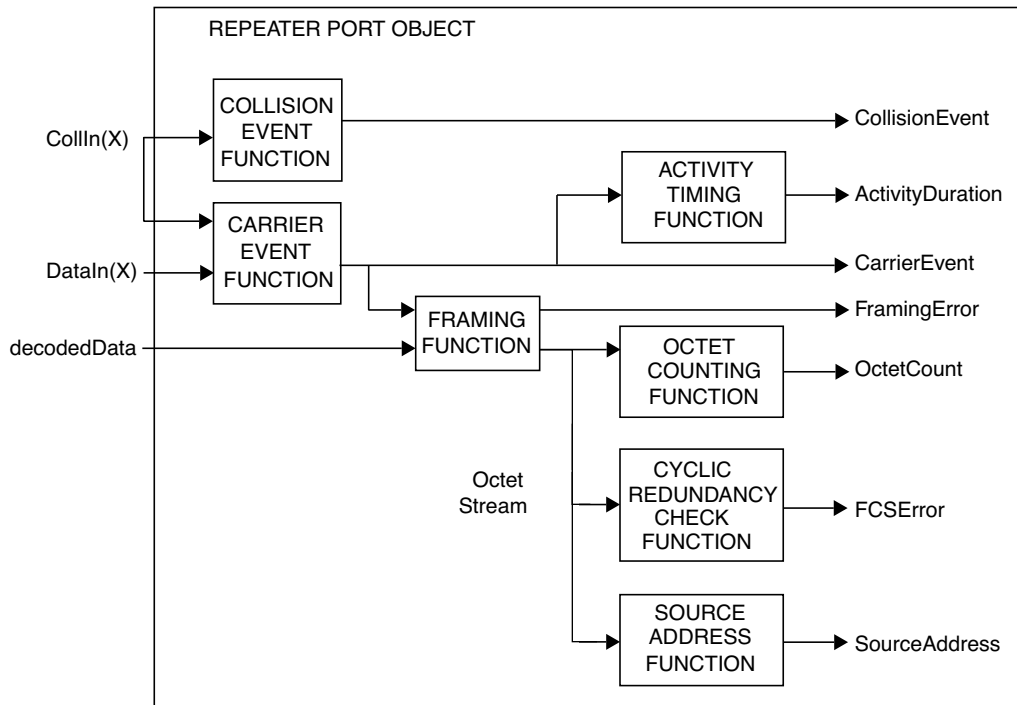


Figure 30–2—Functions relationship

#### Activity Timing function

The Activity Timing function measures the duration of the assertion of the CarrierEvent signal. For 10 Mb/s repeaters, this duration value must be adjusted by removing the value of Carrier Recovery Time (see 9.5.6.5) to obtain the true duration of activity on the network. The output of the Activity Timing function is the ActivityDuration value, which represents the duration of the CarrierEvent signal as expressed in units of bit times.

#### Carrier Event function

For 10 Mb/s repeaters the Carrier Event function for port N asserts the CarrierEvent signal when the repeater exits the IDLE state (see Figure 9–2) and the port has been determined to be port N. It de-asserts the CarrierEvent signal when, for a duration of at least Carrier Recovery Time (see 9.5.6.5), both the DataIn(N) variable has the value II and the CollIn(N) variable has the value –SQE. The value N is the port assigned at the time of transition from the IDLE state. For 100 and 1000 Mb/s repeaters the Carrier Event function for port N asserts the CarrierEvent signal when the repeater exits the IDLE state of the repeater state diagram (Figure 27–2 and Figure 41–2) and the port has been determined to be port N. The Carrier Event function for Port N de-asserts the CarrierEvent signal when the repeater enters the IDLE state of the

repeater state diagram (Figure 27–2 and Figure 41–2).

### **Collision Event function**

The Collision Event function asserts the CollisionEvent signal when a collision is detected at a repeater port. For a 10 Mb/s repeater port this is indicated by the CollIn(X) variable having the value SQE. For a 100 and 1000 Mb/s repeater port this is indicated by entering the COLLISION COUNT INCREMENT state of the partition state diagram (Figure 27–8 and Figure 41–4). The CollisionEvent signal remains asserted until the assertion of any CarrierEvent signal due to the reception of the following event.

### **Cyclic Redundancy Check function**

The Cyclic Redundancy Check function verifies that the sequence of octets output by the Framing function contains a valid Frame Check Sequence Field. The Frame Check Sequence Field is the last four octets received from the output of the Framing function. The algorithm for generating an FCS from the octet stream is specified in 3.2.8. If the FCS generated according to this algorithm is not the same as the last four octets received from the Framing function, then the FCSError signal is asserted. The FCSError signal is cleared and the Cyclic Redundancy Check function is reinitialized upon the assertion of the CarrierEvent signal due to the reception of the following event and, additionally in the case of a frame burst on a 1000 Mb/s network, upon the recognition of a valid Start of Packet delimiter (see 35.2.3.6), once the duration of the CarrierEvent is greater than or equal to slotTime.

### **Framing function**

The Framing function recognizes the boundaries of an incoming frame by monitoring the CarrierEvent signal and the decoded data stream. Data bits are accepted while the CarrierEvent signal is asserted. The framing function delineates frames by stripping the Start of Packet delimiter (see 35.2.3.6), preamble, Start of frame delimiter, End of Packet delimiter (see 35.2.3.6), and any Carrier Extend from the received data stream. The remaining bits of each frame are aligned along octet boundaries. If there is not an integral number of octets, then FramingError shall be asserted. The FramingError signal is cleared upon the assertion of the CarrierEvent signal due to the reception of the following event. For 1000 Mb/s repeaters, the data stream shall continue until the end of the CarrierEvent signal or upon the recognition of a valid Start of Packet delimiter once the duration of the CarrierEvent is greater than or equal to slotTime. Such a Start of Packet delimiter will begin a new data stream.

### **Octet Counting function**

The Octet Counting function counts the number of complete octets received from the output of the framing function. The output of the octet counting function is the OctetCount value. The OctetCount value is reset to zero upon the assertion of the CarrierEvent signal due to the reception of the following event and, additionally in the case of a frame burst on a 1000 Mb/s network, upon the recognition of a valid Start of Packet delimiter (see 35.2.3.6), once the duration of the CarrierEvent is greater than or equal to slotTime.

### Source Address function

The Source Address function extracts octets from the stream output by the framing function. The seventh through twelfth octets shall be extracted from the octet stream and output as the SourceAddress variable. The SourceAddress variable is set to an invalid state upon the assertion of the CarrierEvent signal due to the reception of the following event and, additionally in the case of a frame burst on a 1000 Mb/s network, upon the recognition of a valid Start of Packet delimiter (see 35.2.3.6), once the duration of the CarrierEvent is greater than or equal to slotTime.

### 30.2.3 Containment

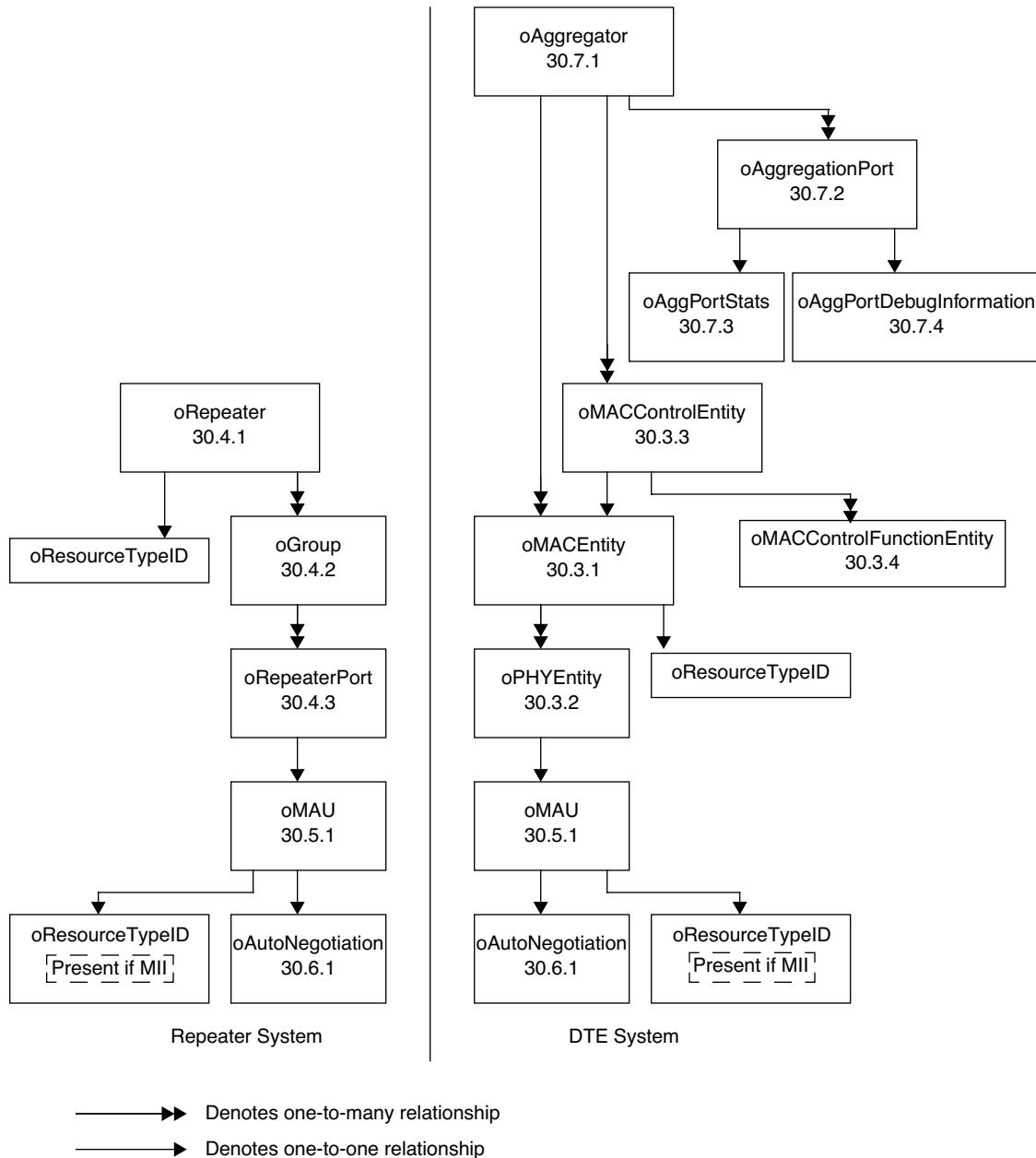
A containment relationship is a structuring relationship for managed objects in which the existence of a managed object is dependent on the existence of a containing managed object. The contained managed object is said to be the subordinate managed object, and the containing managed object the superior managed object. The containment relationship is used for naming managed objects. The local containment relationships among object classes are depicted in the entity relationship diagram, Figure 30–3. This figure shows the names of the object classes and whether a particular containment relationship is one-to-one or one-to-many. For further requirements on this topic, see IEEE Std 802.1F-1993™.

MAU management is only valid in a system that provides management at the next higher containment level, that is, either a DTE or repeater with management.

### 30.2.4 Naming

The name of an individual managed object is hierarchically defined within a managed system. For example, in the context of repeater management, a repeater port might be identified as “repeater 3, group 01, port 13,” that is, port 13 of group 01 of a repeater with repeaterID 3 within the managed system.

In the case of MAU management, this will present itself in one of the two forms that are appropriate for a MAU’s use, that is, as associated with a CSMA/CD interface of a DTE or with a particular port of a managed repeater. For example, a MAU could be identified as “repeater 3, group 01, port 13, MAU 1” or, that is, the MAU associated with port 13 of group 01 of a repeater with repeaterID 3 within the managed system. Examples of this are represented in the relationship of the naming attributes in the entity relationship diagram, Figure 30–3.



**Figure 30–3—10/100/1000 Mb/s MAC, Control, and Link Aggregation entity relationship diagram**

### 30.2.5 Capabilities

This standard makes use of the concept of *packages*, as defined in ISO/IEC 10165-4: 1992, as a means of grouping behavior, attributes, actions, and notifications within a managed object class definition. Packages may be either mandatory or conditional, that is to say, present if a given condition is true. Within this standard *capabilities* are defined, each of which corresponds to a set of packages that are components of a number of managed object class definitions and that share the same condition for presence. Implementation of the appropriate basic and mandatory packages is the minimum requirement for claiming conformance to

IEEE 802.3<sup>®</sup> 10 Mb/s, 100 Mb/s, 1000 Mb/s, MAC Control, and Link Aggregation Management. Implementation of an entire optional capability is required in order to claim conformance to that capability. The capabilities and packages for 10 Mb/s, 100 Mb/s, and 1000 Mb/s Management are specified in Table 30–1a, Table 30–1b, Table 30–1c, Table 30–1d, and Table 30–1e. The capabilities and packages for Link Aggregation Management are specified in Table 30–2.

DTE Management has two packages that are required for management at the minimum conformance configuration—the Basic Package and the Mandatory Package. Systems that implement the optional MAC Control sublayer shall also implement the Basic and Mandatory Packages for the MAC Control Entity managed object class to claim DTE minimum conformance. For systems that include multiple PHY entities per MAC entity and implement the Multiple PHY Package to manage the selection of the active PHY, the optional Recommended Package shall be implemented. Systems that implement the optional Link Aggregation sublayer shall also implement the Basic and Mandatory Packages for the Aggregator and Aggregation Port managed object class to claim minimum DTE conformance.

For managed MAUs, the Basic Package is mandatory; all other packages are optional. For a managed MAU to be conformant to this standard, it shall fully implement the Basic Package. For a MAU to be conformant to an optional package it shall implement that entire package. While nonconformant (reference `aMAUType = "other"`) MAUs may utilize some or all of this clause to specify their management, conformance to this clause requires both a conformant MAU and conformant management. MAU Management is optional with respect to all other CSMA/CD Management. If an MII is present then the conditional MII Capability must be implemented. This provides the means to identify the vendor and type of the externally connected device.

There are two distinct aspects of Repeater Management.

The first aspect provides the means to monitor and control the functions of a repeater. These functions include, but are not limited to: identifying a repeater, testing and initializing a repeater, and enabling/disabling a port. This is encompassed by the mandatory Basic Control Capability.

The second aspect provides the means to monitor traffic from attached segments, and to measure traffic sourced by DTEs connected to these segments. This is done by gathering statistics on packets that enter a repeater and maintaining those statistics on a per port basis. This is encompassed by the optional Performance Monitor Capability. The optional Address Tracking Capability provides the means to identify existence and movement of attached DTEs by their MAC addresses. While nonconformant (reference `aRepeaterType = "other"`) repeaters may utilize some or all of this clause to specify their management, conformance to this clause requires both a conformant repeater and conformant management.

If link Auto-Negotiation is present and managed, the Auto-Negotiation managed object class shall be implemented in its entirety. All attributes and actions are mandatory.

The 1000 Mb/s Burst Monitor Capability provides additional attributes that relate only to 1000 Mb/s operation, while the 100 Mb/s Monitor Capability has attributes that apply to a mixed 100 and 1000 Mb/s operation. These attributes are provided to complement the counter attributes of the optional packages and capabilities that apply to 10 Mb/s and mixed 10, 100, and 1000 Mb/s implementations. It is recommended that when the 100/1000 Mb/s Monitor Capability or 1000 Mb/s Burst Monitor Capability is implemented, the appropriate complementary counter packages and capabilities are also implemented.

**Table 30–1a—Capabilities**

			DTE										Repeater				MAU					
			Basic Package (Mandatory)	Mandatory Package (Mandatory)	Recommended Package (Optional)	Optional Package (Optional)	Array Package (Optional)	Excessive Deferral Package (Optional)	Multiple PHY Package (Optional)	100/1000 Mb/s Monitor Capability (Optional)	Basic Control Capability (Mandatory)	Performance Monitor Capability (Optional)	Address Tracking Capability (Optional)	100/1000 Mb/s Monitor Capability (Optional)	1000 Mb/s Burst Monitor Capability (Optional)	Basic Package (Mandatory)	MAU Control Package (Optional)	Media Loss Tracking Package (Conditional)	Broadband DTE MAU Package (Conditional)	MI1 Capability (Conditional)	100/1000 Mb/s Monitor Capability (Optional)	Auto-Negotiation Package (Mandatory)
<b>oResourceTypeID managed object</b>																						
aResourceTypeIDName	ATTRIBUTE	GET	X											X								X
aResourceInfo	ATTRIBUTE	GET	X											X								X
<b>oMACEntity managed object class (30.3.1)</b>																						
aMACID	ATTRIBUTE	GET	X																			
aFramesTransmittedOK	ATTRIBUTE	GET	X																			
aSingleCollisionFrames	ATTRIBUTE	GET	X																			
aMultipleCollisionFrames	ATTRIBUTE	GET	X																			
aFramesReceivedOK	ATTRIBUTE	GET	X																			
aFrameCheckSequenceErrors	ATTRIBUTE	GET	X																			
aAlignmentErrors	ATTRIBUTE	GET	X																			
aOctetsTransmittedOK	ATTRIBUTE	GET		X																		
aFramesWithDeferredXmissions	ATTRIBUTE	GET		X																		
aLateCollisions	ATTRIBUTE	GET		X																		
aFramesAbortedDueToXSColls	ATTRIBUTE	GET		X																		
aFramesLostDueToIntMACXmitError	ATTRIBUTE	GET		X																		
aCarrierSenseErrors	ATTRIBUTE	GET		X																		
aOctetsReceivedOK	ATTRIBUTE	GET		X																		
aFramesLostDueToIntMACRcvError	ATTRIBUTE	GET		X																		
aPromiscuousStatus	ATTRIBUTE	GET-SET		X																		
aReadMulticastAddressList	ATTRIBUTE	GET		X																		
aMulticastFramesXmittedOK	ATTRIBUTE	GET			X																	
aBroadcastFramesXmittedOK	ATTRIBUTE	GET			X																	
aFramesWithExcessiveDeferral	ATTRIBUTE	GET				X																
aMulticastFramesReceivedOK	ATTRIBUTE	GET			X																	
aBroadcastFramesReceivedOK	ATTRIBUTE	GET			X																	
aInRangeLengthErrors	ATTRIBUTE	GET			X																	
aOutOfRangeLengthField	ATTRIBUTE	GET			X																	
aFrameTooLongErrors	ATTRIBUTE	GET			X																	
aMACEnableStatus	ATTRIBUTE	GET-SET			X																	



**Table 30–1b—Capabilities**

			DTE				Repeater				MAU											
			Basic Package (Mandatory)	Mandatory Package (Mandatory)	Recommended Package (Optional)	Optional Package (Optional)	Array Package (Optional)	Excessive Deferral Package (Optional)	Multiple PHY Package (Optional)	100/1000 Mb/s Monitor Capability (Optional)	Basic Control Capability (Mandatory)	Performance Monitor Capability (Optional)	Address Tracking Capability (Optional)	100/1000 Mb/s Monitor Capability (Optional)	1000 Mb/s Burst Monitor Capability (Optional)	Basic Package (Mandatory)	MAU Control Package (Optional)	Media Loss Tracking Package (Conditional)	Broadband DTE MAU Package (Conditional)	MIU Capability (Conditional)	100/1000 Mb/s Monitor Capability (Optional)	Auto-Negotiation Package (Mandatory)
<b>oMACEntity managed object class (con'd.)</b>																						
aTransmitEnableStatus	ATTRIBUTE	GET-SET			X																	
aMulticastReceiveStatus	ATTRIBUTE	GET-SET			X																	
aReadWriteMACAddress	ATTRIBUTE	GET-SET			X																	
aCollisionFrames	ATTRIBUTE	GET				X																
aMACCapabilities	ATTRIBUTE	GET	X <sup>1</sup>	X																		
aDuplexStatus	ATTRIBUTE	GET-SET	X <sup>1</sup>	X																		
acInitializeMAC	ACTION		X																			
acAddGroupAddress	ACTION			X																		
acDeleteGroupAddress	ACTION			X																		
acExecuteSelfTest	ACTION				X																	
<b>oPHYEntity managed object class (30.3.2)</b>																						
aPHYID	ATTRIBUTE	GET	X																			
aPHYType	ATTRIBUTE	GET	X																			
aPHYTypeList	ATTRIBUTE	GET	X																			
aSQETestErrors	ATTRIBUTE	GET		X																		
aSymbolErrorDuringCarrier	ATTRIBUTE	GET							X													
aMIIDetect	ATTRIBUTE	GET	X																			
aPHYAdminState	ATTRIBUTE	GET	X																			
acPHYAdminControl	ACTION							X														
<b>oMACControlEntity managed object class (30.3.3)</b>																						
aMACControlID	ATTRIBUTE	GET	X																			
aMACControlFunctionsSupported	ATTRIBUTE	GET-SET	X																			
aMACControlFramesTransmitted	ATTRIBUTE	GET		X																		
aMACControlFramesReceived	ATTRIBUTE	GET		X																		
aUnsupportedOpcodesReceived	ATTRIBUTE	GET		X																		
NOTE 1: The aMACCapailities and aDuplexStatus attributes are Mandatory in systems that can operate in full duplex mode. They are Recommended in systems that can operate only in half duplex mode.																						

**Table 30–1c—Capabilities**

				DTE										Repeater				MAU					
				Basic Package (Mandatory)	Mandatory Package (Mandatory)	Recommended Package (Optional)	Optional Package (Optional)	Array Package (Optional)	Excessive Deferral Package (Optional)	Multiple PHY Package (Optional)	100/1000 Mb/s Monitor Capability (Optional)	Basic Control Capability (Mandatory)	Performance Monitor Capability (Optional)	Address Tracking Capability (Optional)	100/1000 Mb/s Monitor Capability (Optional)	1000 Mb/s Burst Monitor Capability (Optional)	Basic Package (Mandatory)	MAU Control Package (Optional)	Media Loss Tracking Package (Conditional)	Broadband DTE MAU Package (Conditional)	MII Capability (Conditional)	100/1000 Mb/s Monitor Capability (Optional)	Auto-Negotiation Package (Mandatory)
<b>oPAUSEEntity managed object class (instance of oMACControlFunctionEntity) (30.3.4)</b>																							
aPAUSELinkDelayAllowance	ATTRIBUTE	GET-SET		X																			
aPAUSEMACCtrlFramesTransmitted	ATTRIBUTE	GET			X																		
aPAUSEMACCtrlFramesReceived	ATTRIBUTE	GET		X																			
<b>oRepeater managed object class (30.4.1)</b>																							
aRepeaterID	ATTRIBUTE	GET										X											
aRepeaterType	ATTRIBUTE	GET										X											
aRepeaterGroupCapacity	ATTRIBUTE	GET										X											
aGroupMap	ATTRIBUTE	GET										X											
aRepeaterHealthState	ATTRIBUTE	GET										X											
aRepeaterHealthText	ATTRIBUTE	GET										X											
aRepeaterHealthData	ATTRIBUTE	GET										X											
aTransmitCollisions	ATTRIBUTE	GET											X										
acResetRepeater	ACTION											X											
acExecuteNonDisruptiveSelfTest	ACTION											X											
nRepeaterHealth	NOTIFICATION											X											
nRepeaterReset	NOTIFICATION											X											
nGroupMapChange	NOTIFICATION											X											
<b>oGroup managed object class (30.4.2)</b>																							
aGroupID	ATTRIBUTE	GET										X											
aGroupPortCapacity	ATTRIBUTE	GET										X											
aPortMap	ATTRIBUTE	GET										X											
nPortMapChange	NOTIFICATION											X											

**Table 30–1d—Capabilities**

			DTE	Repeater	MAU
			Basic Package (Mandatory)	Basic Control Capability (Mandatory)	Basic Package (Mandatory)
			Mandatory Package (Mandatory)	Performance Monitor Capability (Optional)	MAU Control Package (Optional)
			Recommended Package (Optional)	Address Tracking Capability (Optional)	Media Loss Tracking Package (Conditional)
			Optional Package (Optional)	100/1000 Mb/s Monitor Capability (Optional)	Broadband DTE MAU Package (Conditional)
			Array Package (Optional)	1000 Mb/s Burst Monitor Capability (Optional)	MIU Capability (Conditional)
			Excessive Deferral Package (Optional)		100/1000 Mb/s Monitor Capability (Optional)
			Multiple PHY Package (Optional)		Auto-Negotiation Package (Mandatory)
			100/1000 Mb/s Monitor Capability (Optional)		
<b>oRepeaterPort managed object class (30.4.3)</b>					
aPortID	ATTRIBUTE	GET		X	
aPortAdminState	ATTRIBUTE	GET		X	
aAutoPartitionState	ATTRIBUTE	GET		X	
aReadableFrames	ATTRIBUTE	GET		X	
aReadableOctets	ATTRIBUTE	GET		X	
aFrameCheckSequenceErrors	ATTRIBUTE	GET		X	
aAlignmentErrors	ATTRIBUTE	GET		X	
aFramesTooLong	ATTRIBUTE	GET		X	
aShortEvents	ATTRIBUTE	GET		X	
aRunts	ATTRIBUTE	GET		X	
aCollisions	ATTRIBUTE	GET		X	
aLateEvents	ATTRIBUTE	GET		X	
aVeryLongEvents	ATTRIBUTE	GET		X	
aDataRateMismatches	ATTRIBUTE	GET		X	
aAutoPartitions	ATTRIBUTE	GET		X	
alsolates	ATTRIBUTE	GET			X
aSymbolErrorDuringPacket	ATTRIBUTE	GET			X
aLastSourceAddress	ATTRIBUTE	GET		X	
aSourceAddressChanges	ATTRIBUTE	GET		X	
aBursts	ATTRIBUTE	GET			X
acPortAdminControl	ACTION			X	

**Table 30–1e—Capabilities**

			DTE	Repeater	MAU			
			Basic Package (Mandatory)					
			Mandatory Package (Mandatory)					
			Recommended Package (Optional)					
			Optional Package (Optional)					
			Array Package (Optional)					
			Excessive Deferral Package (Optional)					
			Multiple PHY Package (Optional)					
			100/1000 Mb/s Monitor Capability (Optional)					
			Basic Control Capability (Mandatory)					
			Performance Monitor Capability (Optional)					
			Address Tracking Capability (Optional)					
			100/1000 Mb/s Monitor Capability (Optional)					
			1000 Mb/s Burst Monitor Capability (Optional)					
			Basic Package (Mandatory)					
			MAU Control Package (Optional)					
			Media Loss Tracking Package (Conditional)					
			Broadband DTE MAU Package (Conditional)					
			MIU Capability (Conditional)					
			100/1000 Mb/s Monitor Capability (Optional)					
			Auto-Negotiation Package (Mandatory)					
<b>oMAU managed object class (30.5.1)</b>								
aMAUID	ATTRIBUTE	GET				X		
aMAUType	ATTRIBUTE	GET-SET				X		
aMAUTypeList	ATTRIBUTE	GET				X		
aMediaAvailable	ATTRIBUTE	GET				X		
aLoseMediaCounter	ATTRIBUTE	GET					X	
aJabber	ATTRIBUTE	GET				X		
aMAUAdminState	ATTRIBUTE	GET				X		
aBbMAUXmitRcvSplitType	ATTRIBUTE	GET					X	
aBroadbandFrequencies	ATTRIBUTE	GET					X	
aFalseCarriers	ATTRIBUTE	GET						X
aIdleErrorCount	ATTRIBUTE	GET						X
acResetMAU	ACTION					X		
acMAUAdminControl	ACTION					X		
nJabber	NOTIFICATION					X		
<b>oAuto-Negotiation managed object class (30.6.1)</b>								
aAutoNegID	ATTRIBUTE	GET						X
aAutoNegAdminState	ATTRIBUTE	GET						X
aAutoNegRemoteSignaling	ATTRIBUTE	GET						X
aAutoNegAutoConfig	ATTRIBUTE	GET-SET						X
aAutoNegLocalTechnologyAbility	ATTRIBUTE	GET						X
aAutoNegAdvertisedTechnologyAbility	ATTRIBUTE	GET-SET						X
aAutoNegReceivedTechnologyAbility	ATTRIBUTE	GET						X
aAutoNegLocalSelectorAbility	ATTRIBUTE	GET						X
aAutoNegAdvertisedSelectorAbility	ATTRIBUTE	GET-SET						X
aAutoNegReceivedSelectorAbility	ATTRIBUTE	GET						X
acAutoNegRestartAutoConfig	ACTION							X
acAutoNegAdminControl	ACTION							X
<b>Common Attributes Template</b>								
aCMCounter	ATTRIBUTE	GET	X	X	X	X	X	X

Table 30–2—Link Aggregation Capabilities

Object Name	Object Type	Operations Supported	DTE					
			Basic Package (Mandatory)	Mandatory Package (Mandatory)	Recommended Package (Optional)	Optional Package (Optional)	Aggregation Port Statistics (Optional)	Aggregation Port Debug Information (Optional)
oAggregator (30.7.1)								
aAggID	ATTRIBUTE	GET	X					
aAggDescription	ATTRIBUTE	GET		X				
aAggName	ATTRIBUTE	GET-SET		X				
aAggActorSystemID	ATTRIBUTE	GET-SET		X				
aAggActorSystemPriority	ATTRIBUTE	GET-SET		X				
aAggAggregateOrIndividual	ATTRIBUTE	GET		X				
aAggActorAdminKey	ATTRIBUTE	GET-SET		X				
aAggActorOperKey	ATTRIBUTE	GET		X				
aAggMACAddress	ATTRIBUTE	GET		X				
aAggPartnerSystemID	ATTRIBUTE	GET		X				
aAggPartnerSystemPriority	ATTRIBUTE	GET		X				
aAggPartnerOperKey	ATTRIBUTE	GET		X				
aAggAdminState	ATTRIBUTE	GET-SET		X				
aAggOperState	ATTRIBUTE	GET		X				
aAggTimeOfLastOperChange	ATTRIBUTE	GET		X				
aAggDataRate	ATTRIBUTE	GET		X				
aAggOctetsTxOK	ATTRIBUTE	GET			X			
aAggOctetsRxOK	ATTRIBUTE	GET			X			
aAggFramesTxOK	ATTRIBUTE	GET		X				
aAggFramesRxOK	ATTRIBUTE	GET		X				
aAggMulticastFramesTxOK	ATTRIBUTE	GET				X		
aAggMulticastFramesRxOK	ATTRIBUTE	GET				X		
aAggBroadcastFramesTxOK	ATTRIBUTE	GET				X		
aAggBroadcastFramesRxOK	ATTRIBUTE	GET				X		
aAggFramesDiscardedOnTx	ATTRIBUTE	GET			X			

**Table 30–2—Link Aggregation Capabilities (continued)**

Object Name	Object Type	Operations Supported	DTE					
			Basic Package (Mandatory)	Mandatory Package (Mandatory)	Recommended Package (Optional)	Optional Package (Optional)	Aggregation Port Statistics (Optional)	Aggregation Port Debug Information (Optional)
aAggFramesDiscardedOnRx	ATTRIBUTE	GET			X			
aAggFramesWithTxErrors	ATTRIBUTE	GET			X			
aAggFramesWithRxErrors	ATTRIBUTE	GET			X			
aAggUnknownProtocolFrames	ATTRIBUTE	GET			X			
aAggLinkUpDownNotificationEnable	ATTRIBUTE	GET-SET		X				
nAggLinkUpNotification	NOTIFICATION			X				
nAggLinkDownNotification	NOTIFICATION			X				
aAggPortList	ATTRIBUTE	GET			X			
aAggCollectorMaxDelay	ATTRIBUTE	GET-SET		X				
oAggregationPort (30.7.2)								
aAggPortID	ATTRIBUTE	GET	X					
aAggPortActorSystemPriority	ATTRIBUTE	GET-SET		X				
aAggPortActorSystemID	ATTRIBUTE	GET		X				
aAggPortActorAdminKey	ATTRIBUTE	GET-SET		X				
aAggPortActorOperKey	ATTRIBUTE	GET		X				
aAggPortPartnerAdminSystemPriority	ATTRIBUTE	GET-SET		X				
aAggPortPartnerOperSystemPriority	ATTRIBUTE	GET		X				
aAggPortPartnerAdminSystemID	ATTRIBUTE	GET-SET		X				
aAggPortPartnerOperSystemID	ATTRIBUTE	GET		X				
aAggPortPartnerAdminKey	ATTRIBUTE	GET-SET		X				
aAggPortPartnerOperKey	ATTRIBUTE	GET		X				
aAggPortSelectedAggID	ATTRIBUTE	GET		X				
aAggPortAttachedAggID	ATTRIBUTE	GET		X				
aAggPortActorPort	ATTRIBUTE	GET		X				
aAggPortActorPortPriority	ATTRIBUTE	GET-SET		X				
aAggPortPartnerAdminPort	ATTRIBUTE	GET-SET		X				

Table 30–2—Link Aggregation Capabilities (*continued*)

Object Name	Object Type	Operations Supported	DTE					
			Basic Package (Mandatory)	Mandatory Package (Mandatory)	Recommended Package (Optional)	Optional Package (Optional)	Aggregation Port Statistics (Optional)	Aggregation Port Debug Information (Optional)
aAggPortPartnerOperPort	ATTRIBUTE	GET		X				
aAggPortPartnerAdminPortPriority	ATTRIBUTE	GET-SET		X				
aAggPortPartnerOperPortPriority	ATTRIBUTE	GET		X				
aAggPortActorAdminState	ATTRIBUTE	GET-SET		X				
aAggPortActorOperState	ATTRIBUTE	GET		X				
aAggPortPartnerAdminState	ATTRIBUTE	GET-SET		X				
aAggPortPartnerOperState	ATTRIBUTE	GET		X				
aAggPortAggregateOrIndividual	ATTRIBUTE	GET		X				
oAggPortStats (30.7.3)								
aAggPortStatsID	ATTRIBUTE	GET					X	
aAggPortStatsLACPDU <sub>s</sub> Rx	ATTRIBUTE	GET					X	
aAggPortStatsMarkerPDU <sub>s</sub> Rx	ATTRIBUTE	GET					X	
aAggPortStatsMarkerResponsePDU <sub>s</sub> Rx	ATTRIBUTE	GET					X	
aAggPortStatsUnknownRx	ATTRIBUTE	GET					X	
aAggPortStatsIllegalRx	ATTRIBUTE	GET					X	
aAggPortStatsLACPDU <sub>s</sub> Tx	ATTRIBUTE	GET					X	
aAggPortStatsMarkerPDU <sub>s</sub> Tx	ATTRIBUTE	GET					X	
aAggPortStatsMarkerResponsePDU <sub>s</sub> Tx	ATTRIBUTE	GET					X	
oAggPortDebugInformation (30.7.4)								
aAggPortDebugInformationID	ATTRIBUTE	GET						X
aAggPortDebugRxState	ATTRIBUTE	GET						X
aAggPortDebugLastRxTime	ATTRIBUTE	GET						X
aAggPortDebugMuxState	ATTRIBUTE	GET						X
aAggPortDebugMuxReason	ATTRIBUTE	GET						X
aAggPortDebugActorChurnState	ATTRIBUTE	GET						X
aAggPortDebugPartnerChurnState	ATTRIBUTE	GET						X

**Table 30–2—Link Aggregation Capabilities (continued)**

Object Name	Object Type	Operations Supported	DTE					
			Basic Package (Mandatory)	Mandatory Package (Mandatory)	Recommended Package (Optional)	Optional Package (Optional)	Aggregation Port Statistics (Optional)	Aggregation Port Debug Information (Optional)
aAggPortDebugActorChurnCount	ATTRIBUTE	GET						X
aAggPortDebugPartnerChurnCount	ATTRIBUTE	GET						X
aAggPortDebugActorSyncTransitionCount	ATTRIBUTE	GET						X
aAggPortDebugPartnerSyncTransitionCount	ATTRIBUTE	GET						X
aAggPortDebugActorChangeCount	ATTRIBUTE	GET						X
aAggPortDebugPartnerChangeCount	ATTRIBUTE	GET						X
Common Attributes Template								
aCMCounter	ATTRIBUTE	GET		X	X	X	X	X

### 30.3 Layer management for DTEs

#### 30.3.1 MAC entity managed object class

This subclause formally defines the behaviours for the oMACEntity managed object class attributes, actions, and notifications.

##### 30.3.1.1 MAC entity attributes

###### 30.3.1.1.1 aMACID

ATTRIBUTE

APPROPRIATE SYNTAX:  
INTEGER

BEHAVIOUR DEFINED AS:

The value of aMACID is assigned so as to uniquely identify a MAC among the subordinate managed objects of the containing object.;



**30.3.1.1.2 aFramesTransmittedOK**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that are successfully transmitted. This counter is incremented when the TransmitStatus is reported as transmitOK. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

**30.3.1.1.3 aSingleCollisionFrames**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 13 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that are involved in a single collision, and are subsequently transmitted successfully. This counter is incremented when the result of a transmission is reported as transmitOK and the attempt value is 2. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2). The contents of this attribute are undefined for MAC entities operating in full duplex mode.;

**30.3.1.1.4 aMultipleCollisionFrames**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 11 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that are involved in more than one collision and are subsequently transmitted successfully. This counter is incremented when the TransmitStatus is reported as transmitOK and the value of the attempts variable is greater than 2 and less or equal to attemptLimit. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2). The contents of this attribute are undefined for MAC entities operating in full duplex mode.;

**30.3.1.1.5 aFramesReceivedOK**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that are successfully received (receiveOK). This does not include frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error. This counter is incremented when the ReceiveStatus is reported as receiveOK. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

**30.3.1.1.6 aFrameCheckSequenceErrors**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of receive frames that are an integral number of octets in length and do not pass the FCS check. This does not include frames received with frame-too-long, or frame-too-short (frame fragment) error. This counter is incremented when the ReceiveStatus is reported as frameCheckError. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

NOTE—Coding errors detected by the physical layer for speeds above 10 Mb/s will cause the frame to fail the FCS check.;

### **30.3.1.1.7 aAlignmentErrors**

**ATTRIBUTE**

**APPROPRIATE SYNTAX:**

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of frames that are not an integral number of octets in length and do not pass the FCS check. This counter is incremented when the ReceiveStatus is reported as alignmentError. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3). This counter will not increment for 8 bit wide group encoding schemes.;

### **30.3.1.1.8 aOctetsTransmittedOK**

**ATTRIBUTE**

**APPROPRIATE SYNTAX:**

Generalized nonresettable counter. This counter has a maximum increment rate of 1 230 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of data and padding octets of frames that are successfully transmitted. This counter is incremented when the TransmitStatus is reported as transmitOK. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

### **30.3.1.1.9 aFramesWithDeferredXmissions**

**ATTRIBUTE**

**APPROPRIATE SYNTAX:**

Generalized nonresettable counter. This counter has a maximum increment rate of 13 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of frames whose transmission was delayed on its first attempt because the medium was busy. This counter is incremented when the Boolean variable deferred has been asserted by the TransmitLinkMgmt function (4.2.8). Frames involved in any collisions are not counted. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2). The contents of this attribute are undefined for MAC entities operating in full duplex mode.;

### **30.3.1.1.10 aLateCollisions**

**ATTRIBUTE**

**APPROPRIATE SYNTAX:**

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of the times that a collision has been detected later than one slotTime from the start of the packet transmission. A late collision is counted twice, i.e., both as a collision and as a lateCollision. This counter is incremented when the lateCollisionCount variable is nonzero. The actual update is incremented in the LayerMgmtTransmitCounters procedure (5.2.4.2). The contents of this attribute are undefined for MAC entities operating in full duplex mode.;

**30.3.1.1.11 aFramesAbortedDueToXSColls****ATTRIBUTE****APPROPRIATE SYNTAX:**

Generalized nonresetable counter. This counter has a maximum increment rate of 3255 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of the frames that, due to excessive collisions, are not transmitted successfully. This counter is incremented when the value of the attempts variable equals attemptLimit during a transmission. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2). The contents of this attribute are undefined for MAC entities operating in full duplex mode.;

**30.3.1.1.12 aFramesLostDueToIntMACXmitError****ATTRIBUTE****APPROPRIATE SYNTAX:**

Generalized nonresetable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of frames that would otherwise be transmitted by the station, but could not be sent due to an internal MAC sublayer transmit error. If this counter is incremented, then none of the other counters in this section are incremented. The exact meaning and mechanism for incrementing this counter is implementation dependent.;

**30.3.1.1.13 aCarrierSenseErrors****ATTRIBUTE****APPROPRIATE SYNTAX:**

Generalized nonresetable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of times that the carrierSense variable was not asserted or was deasserted during the transmission of a frame without collision. This counter is incremented when the carrierSenseFailure flag is true at the end of transmission. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2). The contents of this attribute are undefined for MAC entities operating in full duplex mode.;

**30.3.1.1.14 aOctetsReceivedOK****ATTRIBUTE****APPROPRIATE SYNTAX:**

Generalized nonresetable counter. This counter has a maximum increment rate of 1 230 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of data and padding octets in frames that are successfully received. This does not include octets in frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error. This counter is incremented when the result of a reception is

reported as a receiveOK status. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

#### **30.3.1.1.15 aFramesLostDueToIntMACRcvError**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that would otherwise be received by the station, but could not be accepted due to an internal MAC sublayer receive error. If this counter is incremented, then none of the other counters in this section are incremented. The exact meaning and mechanism for incrementing this counter is implementation dependent.;

#### **30.3.1.1.16 aPromiscuousStatus**

ATTRIBUTE

APPROPRIATE SYNTAX:

BOOLEAN

BEHAVIOUR DEFINED AS:

A GET operation returns the value “true” for promiscuous mode enabled, and “false” otherwise.

Frames without errors received solely because this attribute has the value “true” are counted as frames received correctly; frames received in this mode that do contain errors update the appropriate error counters.

A SET operation to the value “true” provides a means to cause the LayerMgmtRecognizeAddress function to accept frames regardless of their destination address.

A SET operation to the value “false” causes the MAC sublayer to return to the normal operation of carrying out address recognition procedures for station, broadcast, and multicast group addresses (LayerMgmtRecognizeAddress function).;

#### **30.3.1.1.17 aReadMulticastAddressList**

ATTRIBUTE

APPROPRIATE SYNTAX:

SEQUENCE OF MAC addresses

BEHAVIOUR DEFINED AS:

The current multicast address list.;

#### **30.3.1.1.18 aMulticastFramesXmittedOK**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that are successfully transmitted, as indicated by the status value transmitOK, to a group destination address other than broadcast. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

**30.3.1.1.19 aBroadcastFramesXmittedOK**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of the frames that were successfully transmitted as indicated by the TransmitStatus transmitOK, to the broadcast address. Frames transmitted to multicast addresses are not broadcast frames and are excluded. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

**30.3.1.1.20 aFramesWithExcessiveDeferral**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 412 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that deferred for an excessive period of time. This counter may only be incremented once per LLC transmission. This counter is incremented when the excessDefer flag is set. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2). The contents of this attribute are undefined for MAC entities operating in full duplex mode.;

**30.3.1.1.21 aMulticastFramesReceivedOK**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that are successfully received and are directed to an active nonbroadcast group address. This does not include frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error. This counter is incremented as indicated by the receiveOK status, and the value in the destinationField. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

**30.3.1.1.22 aBroadcastFramesReceivedOK**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that are successfully received and are directed to the broadcast group address. This does not include frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error. This counter is incremented as indicated by the receiveOK status, and the value in the destinationField. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

**30.3.1.1.23 aInRangeLengthErrors**

ATTRIBUTE

**APPROPRIATE SYNTAX:**

Generalized nonresetable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of frames with a length/type field (see 3.2.6<sup>12</sup>) value between the minimum unpadded MAC client data size and the maximum allowed MAC client data size, inclusive, that does not match the number of MAC client data octets received. The counter also increments for frames whose length/type field value is less than the minimum allowed unpadded MAC client data size and the number of MAC client data octets received is greater than the minimum unpadded MAC client data size. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

**30.3.1.1.24 aOutOfRangeLengthField****ATTRIBUTE****APPROPRIATE SYNTAX:**

Generalized nonresetable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of frames with a length field value greater than the maximum allowed LLC data size. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

NOTE—In the past, this counter was incremented by frames containing “Type” fields. Due to the modification to legitimize “Type” fields, such frames will now increment aFramesReceivedOK and this counter will not increment.;

**30.3.1.1.25 aFrameTooLongErrors****ATTRIBUTE****APPROPRIATE SYNTAX:**

Generalized nonresetable counter. This counter has a maximum increment rate of 815 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of frames received that exceed the maximum permitted frame size. This counter is incremented when the status of a frame reception is frameTooLong. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

NOTE—Implementations may use either maxUntaggedFrameSize or (maxUntaggedFrameSize + qTagPrefixSize) (see 4.2.7.1 and 4.4.2) for the maximum permitted frame size, either as a constant or as a function of whether the frame received is a basic or tagged frame (see 3.2 and 3.5). In implementations that treat this as a constant it is recommended that the larger value be used. The use of any value other than the larger value in this case may result in the counter being incremented by valid tagged frames.;

**30.3.1.1.26 aMACEnableStatus****ATTRIBUTE****APPROPRIATE SYNTAX:**

BOOLEAN

**BEHAVIOUR DEFINED AS:**

True if MAC sublayer is enabled and false if disabled. This is accomplished by setting or checking the values of the receiveEnabled and transmitEnabled variables. Setting to true provides a means to cause the MAC sublayer to enter the normal operational state at idle. The PLS is reset by this operation (see 7.2.2.2.1). This is accomplished by setting receiveEnabled and transmitEnabled to true.

Setting to false causes the MAC sublayer to end all transmit and receive operations, leaving it in a

<sup>12</sup>The “Type” interpretation of this field was added to the standard by IEEE Std 802.3x-1997™.

disabled state. This is accomplished by setting `receiveEnabled` and `transmitEnabled` to `false`;

#### **30.3.1.1.27 aTransmitEnableStatus**

ATTRIBUTE

APPROPRIATE SYNTAX:  
BOOLEAN

BEHAVIOUR DEFINED AS:

True if transmission is enabled and false otherwise. This is accomplished by setting or checking the value of the `transmitEnabled` variable.

Setting this to true provides a means to enable MAC sublayer frame transmission (`TransmitFrame` function). This is accomplished by setting `transmitEnabled` to true.

Setting this to false will inhibit the transmission of further frames by the MAC sublayer (`TransmitFrame` function). This is accomplished by setting `transmitEnabled` to false;

#### **30.3.1.1.28 aMulticastReceiveStatus**

ATTRIBUTE

APPROPRIATE SYNTAX:  
BOOLEAN

BEHAVIOUR DEFINED AS:

True if multicast receive is enabled, and false otherwise. Setting this to true provides a means to cause the MAC sublayer to return to the normal operation of multicast frame reception. Setting this to false will inhibit the reception of further multicast frames by the MAC sublayer;

#### **30.3.1.1.29 aReadWriteMACAddress**

ATTRIBUTE

APPROPRIATE SYNTAX:  
MACAddress

BEHAVIOUR DEFINED AS:

Read the MAC station address or change the MAC station address to the one supplied (`RecognizeAddress` function). Note that the supplied station address shall not have the group bit set and shall not be the null address;

#### **30.3.1.1.30 aCollisionFrames**

ATTRIBUTE

APPROPRIATE SYNTAX:

A SEQUENCE of 32 generalized nonresettable counters. Each counter has a maximum increment rate of 13 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A histogram of collision activity. The indices of this array (1 to `attemptLimit - 1`) denote the number of collisions experienced in transmitting a frame. Each element of this array contains a counter that denotes the number of frames that have experienced a specific number of collisions. When the `TransmitStatus` is reported as `transmitOK` and the value of the `attempts` variable equals `n`, then `collisionFrames[n-1]` counter is incremented. The elements of this array are incremented in the `LayerMgmtTransmitCounters` procedure (5.2.4.2). The contents of this attribute are undefined for MAC entities operating in full duplex mode;

#### **30.3.1.1.31 aMACCapabilities**

ATTRIBUTE

**APPROPRIATE SYNTAX:**

A SEQUENCE that meets the requirements of the description below:

half duplex      Capable of operating in half duplex mode

full duplex      Capable of operating in full duplex mode

**BEHAVIOUR DEFINED AS:**

This indicates the capabilities of the MAC.;

**30.3.1.1.32 aDuplexStatus****ATTRIBUTE****APPROPRIATE SYNTAX:**

An ENUMERATED VALUE that has one of the following entries:

half duplex      Half duplex mode

full duplex      Full duplex mode

unknown          Duplex status unknown

**BEHAVIOUR DEFINED AS:**

A GET operation returns the current mode of operation of the MAC entity, either half duplex, full duplex, or unknown.

A SET operation changes the mode of operation of the MAC entity to the indicated value. A SET operation shall have no effect on a device whose mode cannot be changed through management or that can only operate in a single mode.;

**30.3.1.2 MAC entity actions****30.3.1.2.1 acInitializeMAC****ACTION****APPROPRIATE SYNTAX:**

None required

**BEHAVIOUR DEFINED AS:**

This action provides a means to call the Initialize procedure (4.2.7.5). This action also results in the initialization of the PLS.;

**30.3.1.2.2 acAddGroupAddress****ACTION****APPROPRIATE SYNTAX:**

MACAddress

**BEHAVIOUR DEFINED AS:**

Add the supplied multicast group address to the address recognition filter (RecognizeAddress function).;

**30.3.1.2.3 acDeleteGroupAddress****ACTION****APPROPRIATE SYNTAX:**

MACAddress

**BEHAVIOUR DEFINED AS:**

Delete the supplied multicast group address from the address recognition filter (RecognizeAddress function).;



**30.3.1.2.4 acExecuteSelfTest**

ACTION

APPROPRIATE SYNTAX:

None required

BEHAVIOUR DEFINED AS:

Execute a self-test and report the results (success or failure). The actual mechanism employed to carry out the self-test is not defined in this standard. If a Clause 22 MII or Clause 35 GMII is present then this action shall also invoke a data integrity test using MII or GMII loopback, returning to normal operation on completion of the test.;

**30.3.2 PHY devicePHY device managed object class**

This subclause formally defines the behaviours for the oPHYEntity managed object class attributes, actions and notifications. Management of that portion of the physical sublayer whose physical containment within the DTE is optional is outside the scope of this clause.

**30.3.2.1 PHY device attributes****30.3.2.1.1 aPHYID**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The value of aPHYID is assigned so as to uniquely identify a PHY, i.e., Physical Layer among the subordinate managed objects of system (systemID and system are defined in ISO/IEC 10165-2: 1992 [SMI], Definition of management information).;

**30.3.2.1.2 aPhyType**

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

other	Undefined
unknown	Initializing, true state or type not yet known
none	MII present and nothing connected
10 Mb/s	Clause 7 10 Mb/s Manchester
100BASE-T4	Clause 23 100 Mb/s 8B/6T
100BASE-X	Clause 24 100 Mb/s 4B/5B
100BASE-T2	Clause 32 100 Mb/s PAM5X5
1000BASE-X	Clause 36 1000 Mb/s 8B/10B
1000BASE-T	Clause 40 1000 Mb/s 4D-PAM5

BEHAVIOUR DEFINED AS:

A read-only value that identifies the PHY type. The enumeration of the type is such that the value matches the clause number of this International Standard that specifies the particular PHY. The value of this attribute maps to the value of aMAUType. The enumeration “none” can only occur in a standard implementation where an MII exists and there is nothing connected. However, the attribute aMIIDetect should be used to determine whether an MII exists or not.;

**30.3.2.1.3 aPhyTypeList**

ATTRIBUTE

APPROPRIATE SYNTAX:

A SEQUENCE that meets the requirements of the description below:

other	Undefined
unknown	Initializing, true state or type not yet known
none	MII present and nothing connected
10 Mb/s	Clause 7 10 Mb/s Manchester
100BASE-T4	Clause 23 100 Mb/s 8B/6T
100BASE-X	Clause 24 100 Mb/s 4B/5B
100BASE-T2	Clause 32 100 Mb/s PAM5X5
1000BASE-X	Clause 36 1000 Mb/s 8B/10B
1000BASE-T	Clause 40 1000 Mb/s 4D-PAM5

#### BEHAVIOUR DEFINED AS:

A read-only list of the possible types that the PHY could be, identifying the ability of the PHY. If Clause 28 or Clause 37, Auto-Negotiation, is present, then this attribute will map to the local technology ability or advertised ability of the local device.;

#### 30.3.2.1.4 aSQETestErrors

##### ATTRIBUTE

##### APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

##### BEHAVIOUR DEFINED AS:

A count of times that the SQE\_TEST\_ERROR was received. The SQE\_TEST\_ERROR is set in accordance with the rules for verification of the SQE detection mechanism in the PLS Carrier Sense function (see 7.2.4.6). The SQE test function is not a part of 100 or 1000 Mb/s PHY operation, and so SQETestErrors will not occur in 100 or 1000 Mb/s PHYs. The contents of this attribute are undefined for full duplex operation.;

#### 30.3.2.1.5 aSymbolErrorDuringCarrier

##### ATTRIBUTE

##### APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 160 000 counts per second for 100 Mb/s implementations

##### BEHAVIOUR DEFINED AS:

For 100 Mb/s operation it is a count of the number of times when valid carrier was present and there was at least one occurrence of an invalid data symbol (see 23.2.1.4, 24.2.2.1.6, and 32.3.4.1).

For half duplex operation at 1000 Mb/s, it is a count of the number of times the receiving media is non-idle (a carrier event) for a period of time equal to or greater than slotTime (see 4.2.4), and during which there was at least one occurrence of an event that causes the PHY to indicate “Data reception error” or “Carrier Extend Error” on the GMII (see Table 35–2).

For full duplex operation at 1000 Mb/s, it is a count of the number of times the receiving media is non-idle (a carrier event) for a period of time equal to or greater than minFrameSize, and during which there was at least one occurrence of an event that causes the PHY to indicate “Data reception error” on the GMII (see Table 35–2). At all speeds this counter shall be incremented only once per valid CarrierEvent and if a collision is present this counter shall not increment.;

#### 30.3.2.1.6 aMIIDetect

##### ATTRIBUTE

##### APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

- unknown
- present, nothing connected

present, connected  
absent

**BEHAVIOUR DEFINED AS:**

An attribute of the PhyEntity managed object class indicating whether an MII connector is physically present, and if so whether it is detectably connected as specified in 22.2.2.12.;

**30.3.2.1.7 aPhyAdminState**

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has the following entries:

disabled  
enabled

**BEHAVIOUR DEFINED AS:**

A disabled PHY neither transmits nor receives. The PHY shall be explicitly enabled to restore operation. The acPhyAdminControl action provides this ability. The port enable/disable function as reported by this attribute is preserved across DTE reset including loss of power. Only one PHY per MAC can be enabled at any one time. Setting a PHY to the enabled state using the action acPhyAdminControl will result in all other instances of PHY (indicated by PhyID) instantiated within the same MAC to be disabled. If a Clause 22 MII or Clause 35 GMII is present then setting this attribute to “disabled” will result in electrical isolation as defined in 22.2.4.1.6, Isolate; and setting this attribute to “enabled” will result in normal operation as defined in 22.2.4.1.5, Power down; and 22.2.4.1.6, Isolate.;

**30.3.2.2 PHY device actions**

**30.3.2.2.1 acPhyAdminControl**

ACTION

APPROPRIATE SYNTAX:

Same as aPortAdminState

**BEHAVIOUR DEFINED AS:**

This action provides a means to alter aPhyAdminState. Setting a PHY to the enabled state will result in all other instances of PHY being disabled.;

**30.3.3 MAC control entity object class**

This subclause formally defines the behaviours for the oMACControlEntity managed object class attributes.

**30.3.3.1 aMACControlID**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

**BEHAVIOUR DEFINED AS:**

The value of aMACControlID is assigned so as to uniquely identify a MAC Control entity among the subordinate managed objects of the containing object.;

**30.3.3.2 aMACControlFunctionsSupported**

ATTRIBUTE

APPROPRIATE SYNTAX:

A SEQUENCE that meets the requirements of the description below:

PAUSE PAUSE command implemented

**BEHAVIOUR DEFINED AS:**

A read-write list of the possible MAC Control functions implemented within the device. Each function implemented will have an associated MAC Control Function Entity object class. Currently, the only function defined is the PAUSE command.;

**30.3.3.3 aMACControlFramesTransmitted**

ATTRIBUTE

**APPROPRIATE SYNTAX:**

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of MAC Control frames passed to the MAC sublayer for transmission. This counter is incremented when a MA\_CONTROL.request primitive is generated within the MAC Control sublayer.;

**30.3.3.4 aMACControlFramesReceived**

ATTRIBUTE

**APPROPRIATE SYNTAX:**

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of MAC Control frames passed by the MAC sublayer to the MAC Control sublayer. This counter is incremented when a ReceiveFrame function call returns a valid frame with a lengthOrType field value equal to the reserved Type for 802.3\_MAC\_Control as specified in 31.4.1.3.;

**30.3.3.5 aUnsupportedOpcodesReceived**

ATTRIBUTE

**APPROPRIATE SYNTAX:**

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

**BEHAVIOUR DEFINED AS:**

A count of MAC Control frames received that contain an opcode from Table 31A–1 that is not supported by the device. This counter is incremented when a ReceiveFrame function call returns a valid frame with a lengthOrType field value equal to the reserved Type for 802.3\_MAC\_Control as specified in 31.4.1.3, and with an opcode for a function that is not supported by the device.;

**30.3.4 PAUSE entity managed object class**

This subclause formally defines the behaviours for the oPAUSEEntity managed object class attributes.

**30.3.4.1 aPAUSELinkDelayAllowance**

ATTRIBUTE

**APPROPRIATE SYNTAX:**

INTEGER

**BEHAVIOUR DEFINED AS:**

A GET operation returns the value, in bits, of the allowance made by the PAUSE MAC Control entity for round-trip propagation delay of the full duplex link.

A SET operation changes the value of the allowance made by the PAUSE MAC Control entity for round-trip propagation delay of the full duplex link to the indicated value, in bits.;

#### 30.3.4.2 aPAUSEMACCtrlFramesTransmitted

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of PAUSE frames passed to the MAC sublayer for transmission. This counter is incremented when a MA\_CONTROL.request primitive is generated within the MAC Control sublayer with an opcode indicating the PAUSE operation.;

#### 30.3.4.3 aPAUSEMACCtrlFramesReceived

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of MAC Control frames passed by the MAC sublayer to the MAC Control sublayer. This counter is incremented when a ReceiveFrame function call returns a valid frame with: (1) a lengthOrType field value equal to the reserved Type for 802.3\_MAC\_Control as specified in 31.4.1.3, and (2) an opcode indicating the PAUSE operation.;

### 30.4 Layer management for 10, 100, and 1000 Mb/s baseband repeaters

#### 30.4.1 Repeater managed object class

This subclause formally defines the behaviours for the oRepeater managed object class, attributes, actions, and notifications.

##### 30.4.1.1 Repeater attributes

###### 30.4.1.1.1 aRepeaterID

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The value of aRepeaterID is assigned so as to uniquely identify a repeater among the subordinate managed objects of system (systemID and system are defined in ISO/IEC 10165-2: 1992 [SMI], Definition of management information).;

###### 30.4.1.1.2 aRepeaterType

ATTRIBUTE

APPROPRIATE SYNTAX:

An INTEGER that meets the requirements of the following description:

9	10 Mb/s Baseband
271	100 Mb/s Baseband, Class I
272	100 Mb/s Baseband, Class II
41	1000 Mb/s Baseband
other	See 30.2.5
unknown	Initializing, true state or type not yet known

**BEHAVIOUR DEFINED AS:**

Returns a value that identifies the CSMA/CD repeater type. The enumeration of the type is such that the value matches the clause number of the standard that specifies the particular repeater, with further numerical identification for the repeater classes within the same clause.;

**30.4.1.1.3 aRepeaterGroupCapacity**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

INTEGER

## BEHAVIOUR DEFINED AS:

The `aRepeaterGroupCapacity` is the number of groups that can be contained within the repeater. Within each managed repeater, the groups are uniquely numbered in the range from 1 to `aRepeaterGroupCapacity`.

Some groups may not be present in a given repeater instance, in which case the actual number of groups present is less than `aRepeaterGroupCapacity`. The number of groups present is never greater than `aRepeaterGroupCapacity`.;

**30.4.1.1.4 aGroupMap**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

BITSTRING

## BEHAVIOUR DEFINED AS:

A string of bits which reflects the current configuration of units that are viewed by group managed objects. The length of the bitstring is “`aRepeaterGroupCapacity`” bits. The first bit relates to group 1. A “1” in the bitstring indicates presence of the group, “0” represents absence of the group.;

**30.4.1.1.5 aRepeaterHealthState**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

An ENUMERATED VALUE LIST that has the following entries:

other	undefined or unknown
ok	no known failures
repeaterFailure	known to have a repeater related failure
groupFailure	known to have a group related failure
portFailure	known to have a port related failure
generalFailure	has a failure condition, unspecified type

## BEHAVIOUR DEFINED AS:

The `aRepeaterHealthState` attribute indicates the operational state of the repeater. The `aRepeaterHealthData` and `aRepeaterHealthText` attributes may be consulted for more specific information about the state of the repeater’s health. In case of multiple kinds of failures (e.g., repeater failure and port failure), the value of this attribute shall reflect the highest priority in the following order:  
 repeater failure  
 group failure  
 port failure  
 general failure;

**30.4.1.1.6 aRepeaterHealthText**

ATTRIBUTE

APPROPRIATE SYNTAX:

A PrintableString, 255 characters max

BEHAVIOUR DEFINED AS:

The aRepeaterHealthText attribute is a text string that provides information relevant to the operational state of the repeater. Repeater vendors may use this mechanism to provide detailed failure information or instructions for problem resolution.

The contents are vendor specific.;

**30.4.1.1.7 aRepeaterHealthData**

ATTRIBUTE

APPROPRIATE SYNTAX:

OCTET STRING, 0–255

BEHAVIOUR DEFINED AS:

The aRepeaterHealthData attribute is a block of data octets that provides information relevant to the operational state of the repeater. The encoding of this data block is vendor dependent. Repeater vendors may use this mechanism to provide detailed failure information or instructions for problem resolution.;

**30.4.1.1.8 aTransmitCollisions**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

For a Clause 9 repeater, the counter increments every time the repeater state diagram enters the TRANSMIT COLLISION state from any state other than ONE PORT LEFT (Figure 9–2). For a Clause 27 repeater, the counter increments every time the Repeater Core state diagram enters the JAM state as a result of Activity(ALL) > 1 (Figure 27–2). For a Clause 41 repeater, the counter increments every time the Repeater Unit state diagram enters the JAM state (Figure 41–2).

NOTE—Some non-collision events such as false carriers will cause the repeater unit to enter the JAM state and increment this counter.;

**30.4.1.2 Repeater actions****30.4.1.2.1 acResetRepeater**

ACTION

APPROPRIATE SYNTAX:

None required

BEHAVIOUR DEFINED AS:

This causes a transition to the START state of Figure 9–2 for a Clause 9 repeater, to the START state of Figure 27–2 for a Clause 27 repeater, or to the START state of Figure 41–2 for a Clause 41 repeater. The repeater performs a disruptive self-test that has the following characteristics:

1. The components are not specified
2. The test resets the repeater but without affecting management information about the repeater
3. The test does not inject packets onto any segment
4. Packets received during the test may or may not be transferred

5. The test does not interfere with management functions  
This causes a nRepeaterReset notification to be sent.;

#### 30.4.1.2.2 acExecuteNonDisruptiveSelfTest

ACTION

APPROPRIATE SYNTAX:

None required

BEHAVIOUR DEFINED AS:

The repeater performs a vendor-specific, non-disruptive self-test that has the following characteristics:

1. The components are not specified
2. The test does not change the state of the repeater or management information about the repeater
3. The test does not inject packets onto any segment
4. The test does not prevent the transfer of any packets
5. Completion of the test causes a nRepeaterHealth to be sent.;

#### 30.4.1.3 Repeater notifications

##### 30.4.1.3.1 nRepeaterHealth

NOTIFICATION

APPROPRIATE SYNTAX:

A SEQUENCE of three data types. The first is mandatory, the following two are optional. The first is the value of the attribute aRepeaterHealthState. The second is the value of the attribute aRepeaterHealthText. The third is the value of the attribute aRepeaterHealthData

BEHAVIOUR DEFINED AS:

This notification conveys information related to the operational state of the repeater. See the aRepeaterHealthState, aRepeaterHealthText, and aRepeaterHealthData attributes for descriptions of the information that is sent.

The nRepeaterHealth notification is sent only when the health state of the repeater changes. The nRepeaterHealth notification shall contain repeaterHealthState. repeaterHealthData and repeaterHealthText may or may not be included. The nRepeaterHealth notification is not sent as a result of powering up a repeater.;

##### 30.4.1.3.2 nRepeaterReset

NOTIFICATION

APPROPRIATE SYNTAX:

A SEQUENCE of three data types. The first is mandatory, the following two are optional. The first is the value of the attribute aRepeaterHealthState. The second is the value of the attribute aRepeaterHealthText. The third is the value of the attribute aRepeaterHealthData

BEHAVIOUR DEFINED AS:

This notification conveys information related to the operational state of the repeater. The nRepeaterReset notification is sent when the repeater is reset as the result of a power-on condition or upon completion of the acResetRepeater action. The nRepeaterReset notification shall contain repeaterHealthState. repeaterHealthData and repeaterHealthText may or may not be included.;

##### 30.4.1.3.3 nGroupMapChange

NOTIFICATION

APPROPRIATE SYNTAX:

BITSTRING



**BEHAVIOUR DEFINED AS:**

This notification is sent when a change occurs in the group structure of a repeater. This occurs only when a group is logically removed from or added to a repeater. The nGroupMapChange notification is not sent when powering up a repeater. The value of the notification is the updated value of the aGroupMap attribute.;

**30.4.2 Group managed object class**

This subclause formally defines the behaviours for the oGroup managed object class, attributes, actions, and notifications.

**30.4.2.1 Group attributes****30.4.2.1.1 aGroupID**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

INTEGER

**BEHAVIOUR DEFINED AS:**

A value unique within the repeater. The value of aGroupID is assigned so as to uniquely identify a group among the subordinate managed objects of the containing object (oRepeater). This value is never greater than aRepeaterGroupCapacity.;

**30.4.2.1.2 aGroupPortCapacity**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

INTEGER

**BEHAVIOUR DEFINED AS:**

The aGroupPortCapacity is the number of ports contained within the group. Valid range is 1–1024. Within each group, the ports are uniquely numbered in the range from 1 to aGroupPortCapacity. Some ports may not be present in a given group instance, in which case the actual number of ports present is less than aGroupPortCapacity. The number of ports present is never greater than aGroupPortCapacity.;

**30.4.2.1.3 aPortMap**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

BitString

**BEHAVIOUR DEFINED AS:**

A string of bits that reflects the current configuration of port managed objects within this group. The length of the bitstring is “aGroupPortCapacity” bits. The first bit relates to group 1. A “1” in the bitstring indicates presence of the port, “0” represents absence of the port.;

**30.4.2.2 Group notifications****30.4.2.2.1 nPortMapChange**

## NOTIFICATION

## APPROPRIATE SYNTAX:

BitString

**BEHAVIOUR DEFINED AS:**

This notification is sent when a change occurs in the port structure of a group. This occurs only when a port is logically removed from or added to a group. The nPortMapChange notification is not sent when powering up a repeater. The value of the notification is the updated value of the aPortMap attribute.;

### 30.4.3 Repeater port managed object class

This subclause formally defines the behaviours for the oRepeaterPort managed object class, attributes, actions, and notifications.

#### 30.4.3.1 Port attributes

##### 30.4.3.1.1 aPortID

ATTRIBUTE

APPROPRIATE SYNTAX:  
INTEGER

BEHAVIOUR DEFINED AS:

A value unique in the group. It is assumed that ports are partitioned into groups that also have IDs. The value of aPortID is assigned so as to uniquely identify a repeater port among the subordinate managed objects of the containing object (oGroup). This value can never be greater than aGroupPortCapacity.;

##### 30.4.3.1.2 aPortAdminState

ATTRIBUTE

APPROPRIATE SYNTAX:  
An ENUMERATED VALUE LIST that has the following entries:  
disabled  
enabled

BEHAVIOUR DEFINED AS:

A disabled port neither transmits nor receives. The port shall be explicitly enabled to restore operation. The acPortAdminControl action provides this ability. The port enable/disable function as reported by this attribute is preserved across repeater reset including loss of power. aPortAdminState takes precedence over auto-partition and functionally operates between the auto-partition mechanism and the AUI/PMA, PCS/PMA, or GMII/PCS as applicable. For a Clause 9 and 27 repeater, the port auto-partition is reinitialized upon acPortAdminControl taking the value “enabled.” For a Clause 41 repeater, the port auto-partition, receive jabber and carrier integrity functions are reinitialized upon acPortAdminControl taking the value “enabled.”;

##### 30.4.3.1.3 aAutoPartitionState

ATTRIBUTE

APPROPRIATE SYNTAX:  
An ENUMERATED VALUE LIST that has the following entries:  
autoPartitioned  
notAutoPartitioned

BEHAVIOUR DEFINED AS:

The aAutoPartitionState flag indicates whether the port is currently partitioned by the repeater’s auto-partition protection. The conditions that cause port partitioning are specified in partition state diagram in Clauses 9, 27, and 41. They are not differentiated here. A Clause 27 and 41 repeater port partitions on entry to the PARTITION WAIT state of the partition state diagram (Figure 27–8 and Figure 41–4).;

#### 30.4.3.1.4 aReadableFrames

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 15 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A representation of the total frames of valid frame length. Increment counter by one for each frame whose OctetCount is greater than or equal to minFrameSize and for which the FCSError and CollisionEvent signals are not asserted, and for which the attribute aFramesTooLong has not been incremented. Additionally, for 1000 Mb/s repeaters, this count shall only be incremented for frames which are received within a CarrierEvent which has a ActivityDuration of greater than or equal to (slotTime + JamSize) BT (see 4.4.2).

NOTE—This statistic provides one of the parameters necessary for obtaining the packet error rate.;

#### 30.4.3.1.5 aReadableOctets

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 1 240 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

Increment counter by OctetCount for each frame which has been determined to be a readable frame.

NOTE—This statistic provides an indicator of the total data transferred.;

#### 30.4.3.1.6 aFrameCheckSequenceErrors

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 15 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

Increment counter by one for each frame with the FCSError signal asserted and the FramingError and CollisionEvent signals deasserted and whose OctetCount is greater than or equal to minFrameSize and for which the attribute aFramesTooLong has not been incremented.;

#### 30.4.3.1.7 aAlignmentErrors

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 15 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

Increment counter by one for each frame with the FCSError and FramingError signals asserted and CollisionEvent signal deasserted and whose OctetCount is greater than or equal to minFrameSize and for which the attribute aFramesTooLong has not been incremented. If aAlignmentErrors is incremented then the aFrameCheckSequenceErrors attribute shall not be incremented for the same frame. This counter will not increment for 8 bit wide group encoding schemes.;

**30.4.3.1.8 aFramesTooLong**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 815 counts per second at 10 Mb/s

## BEHAVIOUR DEFINED AS:

Increment counter by one for each frame whose OctetCount is greater than maxUntaggedFrameSize or (maxUntaggedFrameSize + qTagPrefixSize) (see 4.2.7.1 and 4.4.2). A repeater may use either value in a constant manner, in which case the larger value is recommended. Alternatively, a repeater may use the appropriate value as a function of whether the frame in question is a basic or tagged MAC frame (see 3.2 and 3.5). If aFrameTooLong is counted then neither the aAlignmentErrors nor the aFrameCheckSequenceErrors attribute shall be incremented for the frame.

**30.4.3.1.9 aShortEvents**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

## BEHAVIOUR DEFINED AS:

Increment counter by one for each CarrierEvent with ActivityDuration less than ShortEventMaxTime. In the 10 Mb/s case ShortEventMaxTime is greater than 74 BT and less than 82 BT. ShortEventMaxTime has tolerances included to provide for circuit losses between a conformance test point at the AUI and the measurement point within the state diagram. In the 100 Mb/s case ShortEventMaxTime is 84 bits (21 nibbles), and for the 1000 Mb/s case ShortEventMaxTime is 72 bits (9 octets).

NOTE 1—shortEvents may indicate externally generated noise hits which will cause the repeater to transmit Runts to its other ports, or propagate a collision (which may be late) back to the transmitting DTE and damaged frames to the rest of the network.

NOTE 2—implementors may wish to consider selecting the ShortEventMaxTime towards the lower end of the allowed tolerance range to accommodate bit losses suffered through physical channel devices not budgeted for within this standard.

NOTE 3—Note also that the significance of this attribute is different in 10, 100, and 1000 Mb/s collision domains. Clause 9 repeaters perform fragment extension of short events which would be counted as runts on the interconnect ports of other repeaters. Clause 27 repeaters do not perform fragment extension. Clause 41 repeaters support one repeater per collision domain and do not perform fragment extension.;

**30.4.3.1.10 aRunts**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

## BEHAVIOUR DEFINED AS:

Increment counter by one for each CarrierEvent that meets one of the following two conditions. Only one test need be made. a) The ActivityDuration is greater than ShortEventMaxTime and less than ValidPacketMinTime and the CollisionEvent signal is deasserted. b) The OctetCount is less than 64, the ActivityDuration is greater than ShortEventMaxTime, and the CollisionEvent signal is deasserted. For 10 and 100 Mb/s repeaters, ValidPacketMinTime is greater than or equal to 552 BT and less than 565 BT. A CarrierEvent greater than or equal to 552 BT but less than 565 BT may or may not be counted as a runt. At 10 Mb/s an event whose length is greater than 74 BT but

less than 82 BT shall increment either the aShortEvents attribute or the aRunts attribute, but not both. ValidPacketMinTime has tolerances included to provide for circuit losses between a conformance test point at the AUI and the measurement point within the state diagram. For 1000 Mb/s repeaters, ValidPacketMinTime is 4136 BT.

NOTE—Runts usually indicate collision fragments, a normal network event. In certain situations associated with large diameter networks a percentage of runs may exceed ValidPacketMinTime.;

#### 30.4.3.1.11 aCollisions

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

This counter increments for any CarrierEvent signal on any port in which the CollisionEvent signal on this port is asserted.;

#### 30.4.3.1.12 aLateEvents

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

For a Clause 9 repeater port this counter increments for each CarrierEvent in which the CollIn(X) variable transitions to the value SQE (see 9.6.6.2) while the ActivityDuration is greater than the LateEventThreshold. For a Clause 27 and Clause 41 repeater port this counter increments for each CarrierEvent in which the CollisionEvent signal assertion occurs while the ActivityDuration is greater than the LateEventThreshold. In both cases such a CarrierEvent is counted twice, as both an aCollision and as an aLateEvent.

For 10 and 100 Mb/s repeaters, the LateEventThreshold is greater than 480 BT and less than 565 BT. LateEventThreshold has tolerances included to permit an implementation to build a single threshold to serve as both the LateEventThreshold and ValidPacketMinTime threshold. For 1000 Mb/s repeaters, the LateEventThreshold is equal to ValidPacketMinTime.;

#### 30.4.3.1.13 aVeryLongEvents

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 250 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

For a Clause 9 repeater port this counter increments for each CarrierEvent whose ActivityDuration is greater than the MAU Jabber Lockup Protection timer TW3 (see 9.6.1, 9.6.5). For a Clause 27 and Clause 41 repeater port this counter increments on entry to the RX JABBER state of the receive timer state diagram (Figure 27–7 and Figure 41–3). Other counters may be incremented as appropriate.;

#### 30.4.3.1.14 aDataRateMismatches

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter

**BEHAVIOUR DEFINED AS:**

Increment counter by one for each frame received by this port that meets all of the conditions required by only one of the following three measurement methods:

Measurement method A, which is valid for 10 and 100 Mb/s operation only: a) The CollisionEvent signal is not asserted. b) The ActivityDuration is greater than ValidPacketMinTime. c) The received frequency (data rate) is detectably mismatched from the local transmit frequency.

Measurement method B, which is valid for 10 and 100 Mb/s operation only: a) The CollisionEvent signal is not asserted. b) The OctetCount is greater than 63. c) The received frequency (data rate) is detectably mismatched from the local transmit frequency.

Measurement method C, which is valid for 1000 Mb/s operation only: The received frequency (data rate) is detectably mismatched from the local transmit frequency.

The exact degree of mismatch is vendor specific and is to be defined by the vendor for conformance testing. When this event occurs, other counters whose increment conditions were satisfied may or may not also be incremented, at the implementor's discretion.

NOTE—Whether or not the repeater was able to maintain data integrity is beyond the scope of this standard.;

**30.4.3.1.15 aAutoPartitions**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter

## BEHAVIOUR DEFINED AS:

Increment counter by one for each time that the repeater has automatically partitioned this port. The conditions that cause a Clause 9 repeater port to partition are specified in the partition state diagram in Clause 9. They are not differentiated here. A Clause 27 and Clause 41 repeater port partitions on entry to the PARTITION WAIT state of the partition state diagram (Figure 27–8 and Figure 41–4).;

**30.4.3.1.16 alsolates**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 400 counts per second at 100 Mb/s

## BEHAVIOUR DEFINED AS:

Increment counter by one each time that the repeater port automatically isolates as a consequence of false carrier events. The conditions that cause a port to automatically isolate are as defined by the transition from the FALSE CARRIER state to the LINK UNSTABLE state of the carrier integrity state diagram (Figure 27–9 and Figure 41–5).

NOTE—Isolates do not affect the value of aPortAdminState.;

**30.4.3.1.17 aSymbolErrorDuringPacket**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 160 000 counts per second for 100 Mb/s implementations

## BEHAVIOUR DEFINED AS:

A count of the number of times when valid length packet was received at the port and there was at least one occurrence of an invalid data symbol. This can increment only once per valid CarrierEvent. A collision presence at any port of the repeater containing port N, will not cause this attribute to increment.;

#### **30.4.3.1.18 aLastSourceAddress**

ATTRIBUTE

APPROPRIATE SYNTAX:

MACAddress

BEHAVIOUR DEFINED AS:

The Source Address of the last readable Frame received by this port.;

#### **30.4.3.1.19 aSourceAddressChanges**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 15 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

Increment counter by one each time that the aLastSourceAddress attribute has changed.

NOTE—This may indicate whether a link is connected to a single DTE or another multi-user segment.;

#### **30.4.3.1.20 aBursts**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 235 000 counts per second at 1000 Mb/s

BEHAVIOUR DEFINED AS:

This counter is valid for 1000 Mb/s operation only. This counter increments by one for each CarrierEvent with ActivityDuration greater than or equal to slotTime during which the CollisionEvent signal has not been asserted. Note that this counter will not increment for a 10 or 100 Mb/s port as packet bursting is not supported at these speeds.;

### **30.4.3.2 Port actions**

#### **30.4.3.2.1 acPortAdminControl**

ACTION

APPROPRIATE SYNTAX:

Same as aPortAdminStateBEHAVIOUR DEFINED AS:

This action provides a means to alter aPortAdminState. For a Clause 9 and 27 repeater it should exert a BEGIN on the Partitioning state diagram (Figures 9–6 and 27–8) upon taking the value “enabled.” For a Clause 41 repeater it shall exert a BEGIN on the receive timer, partition, and carrier integrity state diagrams (Figure 41–3, Figure 41–4, and Figure 41–5) upon taking the value “enabled.”;

## 30.5 Layer management for 10, 100, and 1000 Mb/s medium attachment units (MAUs)

### 30.5.1 MAU managed object class

This subclause formally defines the behaviours for the oMAU managed object class, attributes, actions, and notifications.

#### 30.5.1.1 MAU attributes

##### 30.5.1.1.1 aMAUID

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The value of aMAUID is assigned so as to uniquely identify a MAU among the subordinate managed objects of the containing object.;

##### 30.5.1.1.2 aMAUType

ATTRIBUTE

APPROPRIATE SYNTAX:

A GET-SET ENUMERATION that meets the requirements of the description below:

global	undefined
other	See 30.2.5
unknown	Initializing, true state or type not yet known
AUI	no internal MAU, view from AUI
10BASE5	Thick coax MAU as specified in Clause 8
FOIRL	FOIRL MAU as specified in 9.9
10BASE2	Thin coax MAU as specified in Clause 10
10BROAD36	Broadband DTE MAU as specified in Clause 11
10BASE-T	UTP MAU as specified in Clause 14, duplex mode unknown
10BASE-THD	UTP MAU as specified in Clause 14, half duplex mode
10BASE-TFD	UTP MAU as specified in Clause 14, full duplex mode
10BASE-FP	Passive fiber MAU as specified in Clause 16
10BASE-FB	Synchronous fiber MAU as specified in Clause 17
10BASE-FL	Asynchronous fiber MAU as specified in Clause 18, duplex mode unknown
10BASE-FLHD	Asynchronous fiber MAU as specified in Clause 18, half duplex mode
10BASE-FLFD	Asynchronous fiber MAU as specified in Clause 18, full duplex mode
100BASE-T4	Four-pair Category 3 UTP as specified in Clause 23
100BASE-TX	Two-pair Category 5 UTP as specified in Clause 25, duplex mode unknown
100BASE-TXHD	Two-pair Category 5 UTP as specified in Clause 25, half duplex mode
100BASE-TXFD	Two-pair Category 5 UTP as specified in Clause 25, full duplex mode
100BASE-FX	X fiber over PMD as specified in Clause 26, duplex mode unknown
100BASE-FXHD	X fiber over PMD as specified in Clause 26, half duplex mode
100BASE-FXFD	X fiber over PMD as specified in Clause 26, full duplex mode
100BASE-T2	Two-pair Category 3 UTP as specified in Clause 32, duplex mode unknown
100BASE-T2HD	Two-pair Category 3 UTP as specified in Clause 32, half duplex mode
100BASE-T2FD	Two-pair Category 3 UTP as specified in Clause 32, full duplex mode
1000BASE-X	X PCS/PMA as specified in Clause 36 over undefined PMD, duplex mode unknown
1000BASE-XHD	X PCS/PMA as specified in Clause 36 over undefined PMD, half duplex mode



1000BASE-XFD X PCS/PMA as specified in Clause 36 over undefined PMD, full duplex mode  
 1000BASE-LX X fiber over long-wavelength laser PMD as specified in Clause 38, duplex mode  
                   unknown  
 1000BASE-LXHD X fiber over long-wavelength laser PMD as specified in Clause 38, half duplex  
                   mode  
 1000BASE-LXFDX fiber over long-wavelength laser PMD as specified in Clause 38, full duplex  
                   mode  
 1000BASE-SX X fiber over short-wavelength laser PMD as specified in Clause 38, duplex mode  
                   unknown  
 1000BASE-SXHD X fiber over short-wavelength laser PMD as specified in Clause 38, half duplex  
                   mode  
 1000BASE-SXFD X fiber over short-wavelength laser PMD as specified in Clause 38, full duplex  
                   mode  
 1000BASE-CX X copper over 150-Ohm balanced cable PMD as specified in Clause 39, duplex  
                   mode unknown  
 1000BASE-CXHDX copper over 150-Ohm balanced cable PMD as specified in Clause 39, half  
                   duplex mode  
 1000BASE-CXFD X copper over 150-Ohm balanced cable PMD as specified in Clause 39, full  
                   duplex mode  
 1000BASE-T Four-pair Category 5 UTP PHY to be specified in Clause 40, duplex mode  
                   unknown  
 1000BASE-THD Four-pair Category 5 UTP PHY to be specified in Clause 40, half duplex mode  
 1000BASE-TFD Four-pair Category 5 UTP PHY to be specified in Clause 40, full duplex mode  
 802.9a Integrated services MAU as specified in IEEE Std 802.9® ISLAN-16T

#### BEHAVIOUR DEFINED AS:

Returns a value that identifies the 10 Mb/s, 100 Mb/s, or 1000 Mb/s internal MAU type. If an AUI is to be identified to access an external MAU, the type “AUI” is returned. A SET operation to one of the possible enumerations indicated by aMAUTypeList will force the MAU into the new operating mode. If a Clause 22 MII or Clause 35 GMII is present, then this will map to the mode force bits specified in 22.2.4.1. If Clause 28 or Clause 37 Auto-Negotiation is operational, then this will change the advertised ability to the single enumeration specified in the SET operation, and cause an immediate link renegotiation. A change in the MAU type will also be reflected in aPHYType.

The enumerations 1000BASE-X, 1000BASE-XHD and 1000BASE-XFD shall only be returned if the underlying PMD type is unknown.;

#### 30.5.1.1.3 aMAUTypeList

ATTRIBUTE

APPROPRIATE SYNTAX:

A SEQUENCE of ENUMERATIONS that match the syntax of aMAUType

BEHAVIOUR DEFINED AS:

A GET attribute that returns the possible types that the MAU could be, identifying the ability of the MAU. This attribute maps to aPHYTypeList.;

#### 30.5.1.1.4 aMediaAvailable

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED value list that has the following entries:

other                    undefined

unknown	initializing, true state not yet known
available	link or light normal, loopback normal
not available	link loss or low light, no loopback
remote fault	remote fault with no detail
invalid signal	invalid signal, applies only to 10BASE-FB
remote jabber	remote fault, reason known to be jabber
remote link loss	remote fault, reason known to be far-end link loss
remote test	remote fault, reason known to be test
offline	offline, applies only to Clause 37 Auto-Negotiation
auto neg error	Auto-Negotiation Error, applies only to Clause 37 Auto-Negotiation

**BEHAVIOUR DEFINED AS:**

If the MAU is a link or fiber type (FOIRL, 10BASE-T, 10BASE-F), then this is equivalent to the link test fail state/low light function. For an AUI, 10BASE2, 10BASE5, or 10BROAD36 MAU, this indicates whether or not loopback is detected on the DI circuit. The value of this attribute persists between packets for MAU types AUI, 10BASE5, 10BASE2, 10BROAD36, and 10BASE-FP.

At power-up or following a reset, the value of this attribute will be “unknown” for AUI, 10BASE5, 10BASE2, 10BROAD36, and 10BASE-FP MAUs. For these MAUs loopback will be tested on each transmission during which no collision is detected. If DI is receiving *input* when DO returns to IDL after a transmission and there has been no collision during the transmission, then loopback will be detected. The value of this attribute will only change during noncollided transmissions for AUI, 10BASE2, 10BASE5, 10BROAD36, and 10BASE-FP MAUs.

For 100BASE-T2, 100BASE-T4, 100BASE-TX, and 100BASE-FX the enumerations match the states within the respective link integrity state diagrams, Figures 32-16, 23-12 and 24-15. Any MAU that implements management of Clause 28 Auto-Negotiation will map remote fault indication to MediaAvailable “remote fault.” Any MAU that implements management of Clause 37 Auto-Negotiation will map the received RF1 and RF2 bits as specified in Table 37-2, as follows. Offline maps to the enumeration “offline,” Link\_Failure maps to the enumeration “remote fault” and Auto-Negotiation Error maps to the enumeration “auto neg error.”

The enumeration “remote fault” applies to 10BASE-FB remote fault indication, the 100BASE-X far-end fault indication and nonspecified remote faults from a system running Clause 28 Auto-Negotiation. The enumerations “remote jabber,” “remote link loss,” or “remote test” should be used instead of “remote fault” where the reason for remote fault is identified in the remote signaling protocol.

Where a Clause 22 MII or Clause 35 GMII is present, a logic one in the remote fault bit (22.2.4.2.11) maps to the enumeration “remote fault,” a logic zero in the link status bit (22.2.4.2.13) maps to the enumeration “not available.” The enumeration “not available” takes precedence over “remote fault.”;

**30.5.1.1.5 aLoseMediaCounter****ATTRIBUTE****APPROPRIATE SYNTAX:**

Generalized nonresettable counter. This counter has a maximum increment rate of 10 counts per second

**BEHAVIOUR DEFINED AS:**

Counts the number of times that the MediaAvailable attribute changes from the enumeration “available” to any other enumeration. Mandatory for MAU type “AUI,” optional for all others.;

**30.5.1.1.6 aJabber****ATTRIBUTE****APPROPRIATE SYNTAX:**

A SEQUENCE of two indications. The first, JabberFlag, consists of an ENUMERATED value list that has the following entries:

other	undefined
unknown	initializing, true state not yet known

normal state is true or normal  
 fault state is false, fault, or abnormal

The second, jabberCounter, is a generalized nonresettable counter. This counter has a maximum increment rate of 40 counts per second

#### BEHAVIOUR DEFINED AS:

If the MAU is in the JABBER state, the jabberFlag portion of the attribute is set to the “fault” value. The jabberCounter portion of the attribute is incremented each time the flag is set to the “fault” value. This attribute returns the value “other” for type AUI. Note that this counter will not increment for a 100 or 1000 Mb/s PHY, as there is no defined JABBER state.;

#### 30.5.1.1.7 aMAUAdminState

##### ATTRIBUTE

##### APPROPRIATE SYNTAX:

An ENUMERATED value list that has the following entries:

other	undefined
unknown	initializing, true state not yet known
operational	powered and connected
standby	inactive but on
shutdown	similar to power down

##### BEHAVIOUR DEFINED AS:

A MAU in management state “standby” forces DI and CI to idle and the media transmitter to idle or fault, if supported. The management state “standby” only applies to link type MAUs. The state of MediaAvailable is unaffected. A MAU or AUI in the management state “shutdown” assumes the same condition on DI, CI and the media transmitter as if it were powered down or not connected. For an AUI, this management state will remove power from the AUI. The MAU may return the value “undefined” for Jabber and MediaAvailable attributes when it is in this management state. A MAU in the management state “operational” is fully functional, and operates and passes signals to its attached DTE or repeater port in accordance with its specification.;

#### 30.5.1.1.8 aBbMAUXmitRcvSplitType

##### ATTRIBUTE

##### APPROPRIATE SYNTAX:

An ENUMERATED value list that has the following entries:

other	undefined
single	single-cable system
dual	dual-cable system, offset normally zero

##### BEHAVIOUR DEFINED AS:

Returns a value that indicates the type of frequency multiplexing/cabling system used to separate the transmit and receive paths for the 10BROAD36 MAU. All other types return “undefined.”;

#### 30.5.1.1.9 aBroadbandFrequencies

##### ATTRIBUTE

##### APPROPRIATE SYNTAX:

A SEQUENCE of two instances of the type INTEGER.

The first INTEGER represents the Transmitter Carrier Frequency. The value of its INTEGER represents the frequency of the carrier divided by 250 kHz.

The second INTEGER represents the Translation Offset Frequency. The value of its INTEGER represents the frequency of the offset divided by 250 kHz

##### BEHAVIOUR DEFINED AS:

Returns a value that indicates the transmit carrier frequency and translation offset frequency in MHz/

4 for the 10BROAD36 MAU. This allows the frequencies to be defined to a resolution of 250 kHz.;

#### 30.5.1.1.10 aFalseCarriers

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 160 000 counts per second under maximum network load, and 10 counts per second under zero network load, for 100 Mb/s implementations. This counter has a maximum increment rate of 1 600 000 counts per second under maximum network load, and 100 000 counts per second under zero network load, for 1000 Mb/s implementations

BEHAVIOUR DEFINED AS:

A count of the number of false carrier events during IDLE in 100BASE-X and 1000BASE-X links. This counter does not increment at the symbol rate. For 100BASE-X, it can increment after a valid carrier completion at a maximum rate of once per 100 ms until the nextCarrierEvent. For 1000BASE-X, it can increment after a valid carrier completion at a maximum rate of once per 10  $\mu$ s until the next CarrierEvent.

NOTE—The increased increment rate for this attribute at 1000 Mb/s relative to its increment rate at 100 Mb/s has been provided to improve its use as an indication of line quality.;

#### 30.5.1.1.11 aIdleErrorCount

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

This attribute takes the eight-bit value from the 100BASE-T2 Status register (MII management register 10) bits 7:0 “Idle Error Count” as described in 100BASE-T2, 32.5.3.2.6 and 40.5.;

### 30.5.1.2 MAU actions

#### 30.5.1.2.1 acResetMAU

ACTION

APPROPRIATE SYNTAX:

None required

BEHAVIOUR DEFINED AS:

Resets the MAU in the same manner as would a power-off, power-on cycle of at least 0.5 s duration. During the 0.5 s DO, DI, and CI should be idle.;

#### 30.5.1.2.2 acMAUAdminControl

ACTION

APPROPRIATE SYNTAX:

The same as used for aMAUAdminState

BEHAVIOUR DEFINED AS:

Executing an acMAUAdminControl action causes the MAU to assume the aMAUAdminState attribute value of one of the defined valid management states for control input. The valid inputs are “standby,” “operational,” and “shutdown” state (see the behaviour definition bMAUAdminState for the description of each of these states) except that a “standby” action to a mixing type MAU or an AUI will cause the MAU to enter the “shutdown” management state.;

### 30.5.1.3 MAU notifications

#### 30.5.1.3.1 nJabber

NOTIFICATION

APPROPRIATE SYNTAX:

The same as used for aJabber

BEHAVIOUR DEFINED AS:

The notification is sent whenever a managed 10 Mb/s MAU enters the JABBER state.;

## 30.6 Management for link Auto-Negotiation

### 30.6.1 Auto-Negotiation managed object class

This subclause formally defines the behaviours for the oAuto-Negotiation managed object class, attributes, actions, and notifications.

#### 30.6.1.1 Auto-Negotiation attributes

##### 30.6.1.1.1 aAutoNegID

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The value of aAutoNegID is assigned so as to uniquely identify an Auto-Negotiation managed object among the subordinate managed objects of the containing object.;

##### 30.6.1.1.2 aAutoNegAdminState

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

enabled

disabled

BEHAVIOUR DEFINED AS:

An interface which has Auto-Negotiation signaling ability will be enabled to do so when this attribute is in the enabled state. If disabled then the interface will act as it would if it had no Auto-Negotiation signaling.;

**30.6.1.1.3 aAutoNegRemoteSignaling**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

detected  
notdetected

## BEHAVIOUR DEFINED AS:

The value indicates whether the remote end of the link is operating Auto-Negotiation signaling or not. It shall take the value detected if, during the previous link negotiation, FLP Bursts or /C/ ordered\_sets (see 36.2.4.10) were received from the remote end.;

**30.6.1.1.4 aAutoNegAutoConfig**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

other  
configuring  
complete  
disabled  
parallel detect fail

## BEHAVIOUR DEFINED AS:

Indicates whether Auto-Negotiation signaling is in progress or has completed. The enumeration “parallel detect fail” maps to a failure in parallel detection as defined in 28.2.3.1.;

**30.6.1.1.5 aAutoNegLocalTechnologyAbility**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

A SEQUENCE that meets the requirements of the description below:

global	Reserved for future use
other	See 30.2.5
unknown	Initializing, true state or type not yet known
10BASE-T	10BASE-T half duplex as defined in Clause 14
10BASE-TFD	Full duplex 10BASE-T as defined in Clauses 14 and 31
100BASE-T4	100BASE-T4 half duplex as defined in Clause 23
100BASE-TX	100BASE-TX half duplex as defined in Clause 25
100BASE-TXFD	Full duplex 100BASE-TX as defined in Clauses 25 and 31
FDX PAUSE	PAUSE operation for full duplex links as defined in Annex 31B
FDX APAUSE	Asymmetric PAUSE operation for full duplex links as defined in Clause 37 and Annex 31B
FDX SPAUSE	Symmetric PAUSE operation for full duplex links as defined in Clause 37 and Annex 31B
FDX BPAUSE	Asymmetric and Symmetric PAUSE operation for full duplex links as defined in Clause 37 and Annex 31B
100BASE-T2	100BASE-T2 half duplex as defined in Clause 32
100BASE-T2FD	Full duplex 100BASE-T2 as defined in Clauses 31 and 32
1000BASE-X	1000BASE-X half duplex as specified in Clause 36
1000BASE-XFD	Full duplex 1000BASE-X as specified in Clauses 31 and 36
1000BASE-T	1000BASE-T half duplex UTP PHY as specified in Clause 40
1000BASE-TFD	Full duplex 1000BASE-T UTP PHY as specified in Clause 31 and as specified in Clause 40

Rem Fault1	Remote fault bit 1 (RF1) as specified in Clause 37
Rem Fault2	Remote fault bit 2 (RF2) as specified in Clause 37
isoethernet	IEEE Std 802.9® ISLAN-16T

**BEHAVIOUR DEFINED AS:**

This indicates the technology ability of the local device, as defined in Clause 28 and Clause 37.

**30.6.1.1.6 aAutoNegAdvertisedTechnologyAbility****ATTRIBUTE****APPROPRIATE SYNTAX:**

Same as aAutoNegLocalTechnologyAbility

**BEHAVIOUR DEFINED AS:**

For Clause 28 Auto-Negotiation this GET-SET attribute maps to the Technology Ability Field of the Auto-Negotiation Link Code Word. For Clause 37 Auto-Negotiation, this GET-SET attribute maps to bits D0-D13 of Config\_Reg base page (see 37.2.1).

A SET operation to a value not available in aAutoNegLocalTechnologyAbility will be rejected. A successful SET operation will result in immediate link renegotiation if aAutoNegAdminState is enabled.

NOTE—This will in every case cause temporary link loss during link renegotiation. If set to a value incompatible with aAutoNegReceivedTechnologyAbility, link negotiation will not be successful and will cause permanent link loss.;

**30.6.1.1.7 aAutoNegReceivedTechnologyAbility****ATTRIBUTE****APPROPRIATE SYNTAX:**

Same as aAutoNegLocalTechnologyAbility

**BEHAVIOUR DEFINED AS:**

Indicates the advertised technology ability of the remote hardware. For Clause 28 Auto-Negotiation, this attribute maps to the Technology Ability Field of the last received Auto-Negotiation Link Code Word(s). For Clause 37 Auto-Negotiation, this attribute maps to bits D0-D13 of the received Config\_Reg base page (see 37.2.1).;

**30.6.1.1.8 aAutoNegLocalSelectorAbility****ATTRIBUTE****APPROPRIATE SYNTAX:**

A SEQUENCE that meets the requirements of the description below:

other	Undefined
ethernet	IEEE Std 802.3®
isoethernet	IEEE Std 802.9a-1995™
tokenring	IEEE Std 802.5®

**BEHAVIOUR DEFINED AS:**

This indicates the value of the selector field of the local hardware. Selector field is defined in 28.2.1.2.1. The enumeration of the Selector Field indicates the standard that defines the remaining encodings for Auto-Negotiation using that value of enumeration. For Clause 37 Auto-Negotiation devices, a SET of this attribute will have no effect, and a GET will return the value ethernet.;

**30.6.1.1.9 aAutoNegAdvertisedSelectorAbility****ATTRIBUTE****APPROPRIATE SYNTAX:**

Same as aAutoNegLocalSelectorAbility

**BEHAVIOUR DEFINED AS:**

In the case of Clause 28 Auto-Negotiation, this GET-SET attribute maps to the Message Selector Field of the Auto-Negotiation Link Code Word. A SET operation to a value not available in aAutoNegLocalSelectorAbility will be rejected. A successful SET operation will result in immediate link renegotiation if aAutoNegAdminState is enabled. For Clause 37 Auto-Negotiation devices, a SET of this attribute will have no effect, and a GET will return the value ethernet.

NOTE—This will in every case cause temporary link loss during link renegotiation. If set to a value incompatible with aAutoNegReceivedSelectorAbility, link negotiation will not be successful and will cause permanent link loss.;

**30.6.1.1.10 aAutoNegReceivedSelectorAbility**

**ATTRIBUTE**

**APPROPRIATE SYNTAX:**

Same as aAutoNegLocalSelectorAbility

**BEHAVIOUR DEFINED AS:**

In the case of Clause 28 Auto-Negotiation, this attribute indicates the advertised message transmission ability of the remote hardware. Maps to the Message Selector Field of the last received Auto-Negotiation Link Code Word. For Clause 37 Auto-Negotiation devices, a SET of this attribute will have no effect, and a GET will return the value ethernet.;

**30.6.1.2 Auto-Negotiation actions**

**30.6.1.2.1 acAutoNegRestartAutoConfig**

**ATTRIBUTE**

**APPROPRIATE SYNTAX:**

None required

**BEHAVIOUR DEFINED AS:**

Forces Auto-Negotiation to begin link renegotiation. Has no effect if Auto-Negotiation signaling is disabled.;

**30.6.1.2.2 acAutoNegAdminControl**

**ATTRIBUTE**

**APPROPRIATE SYNTAX:**

Same as aAutoNegAdminState

**BEHAVIOUR DEFINED AS:**

This action provides a means to turn Auto-Negotiation signaling on or off.;

**30.7 Management for Link Aggregation**

**30.7.1 Aggregator managed object class**

This subclause formally defines the behaviors for the oAggregator managed object class, attributes, and notifications.

Some of the attributes that are part of the definition of the oAggregator managed object class are derived by summing counter values from attributes of other objects; e.g., to generate a count of received frames for the Aggregator, the individual value for each Aggregation Port contributes to the sum. Where calculations of this form are used, the values that contribute to the Aggregator's attributes are *increments* in the values of the



component attributes, not their absolute values. As any individual Aggregation Port is potentially only temporarily attached to its current Aggregator, the count values it contributes to the Aggregator's counters are the increments in its values that it has experienced during the period of time that it has been attached to that Aggregator.

The counter values defined for the Aggregator have been formulated as far as possible to make the Aggregator behave like an individual IEEE 802.3<sup>®</sup> MAC. The counts of frames received and transmitted are formulated to reflect the counts that would be expected by the MAC Client; they do not include frames transmitted and received as part of the operation of LACP or the Marker protocol, only frames that pass through the interface between the MAC Client and the Aggregator. However, as LACP and the Marker protocol are, as far as the individual MACs are concerned, part of their MAC Client, the RX/TX counters for the individual MACs will reflect both control and data traffic. As counts of errors at the port level cannot always be cleanly delineated between those that occurred as a result of aggregation activity and those that did not, no attempt has been made to separate these aspects of the port error counts. Therefore, there is not necessarily a direct correspondence between the individual MAC counters and the corresponding derived counters at the Aggregator level.

It should also be noted that the counters defined for the Aggregator include values that can only apply to half duplex links. This is consistent with the approach taken in Link Aggregation that a link that can only operate as an individual link is nonetheless considered as being attached to an Aggregator. This simplifies the modelling of managed objects for links that can operate in either half or full duplex, and ensures a consistent presentation of the attributes regardless of the type of links attached to the Aggregator.

NOTE—The operation of autonegotiation may mean that a given link can operate in full duplex or half duplex, depending upon the capabilities of the device(s) connected to it. Keeping the management view the same regardless of a link's current mode of operation allows a consistent management approach to be taken across all types of links.

### 30.7.1.1 Aggregator attributes

#### 30.7.1.1.1 aAggID

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The unique identifier allocated to this Aggregator by the local System. This attribute identifies an Aggregator instance among the subordinate managed objects of the containing object. This value is read-only.

NOTE—The aAggID is represented in the SNMP MIB as an ifIndex—see 30C.4.2.;

#### 30.7.1.1.2 aAggDescription

ATTRIBUTE

APPROPRIATE SYNTAX:

A PrintableString, 255 characters max.

BEHAVIOUR DEFINED AS:

A human-readable text string containing information about the Aggregator. This string could include information about the distribution algorithm in use on this Aggregator; for example, “Aggregator 1, Dist Alg=Dest MAC address.” This string is read-only. The contents are vendor specific.;

#### **30.7.1.1.3 aAggName**

ATTRIBUTE

APPROPRIATE SYNTAX:

A PrintableString, 255 characters max.

BEHAVIOUR DEFINED AS:

A human-readable text string containing a locally significant name for the Aggregator. This string is read-write.;

#### **30.7.1.1.4 aAggActorSystemID**

ATTRIBUTE

APPROPRIATE SYNTAX:

MACAddress

BEHAVIOUR DEFINED AS:

A 6-octet read-write MAC address value used as a unique identifier for the System that contains this Aggregator.

NOTE—From the perspective of the Link Aggregation mechanisms described in Clause 43, only a single combination of Actor’s System ID and System Priority are considered, and no distinction is made between the values of these parameters for an Aggregator and the port(s) that are associated with it (i.e., the protocol is described in terms of the operation of aggregation within a single System). However, the managed objects provided for the Aggregator and the port both allow management of these parameters. The result of this is to permit a single piece of equipment to be configured by management to contain more than one System from the point of view of the operation of Link Aggregation. This may be of particular use in the configuration of equipment that has limited aggregation capability (see 43.6).;

#### **30.7.1.1.5 aAggActorSystemPriority**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

A 2-octet read-write value indicating the priority value associated with the Actor’s System ID.;

#### **30.7.1.1.6 aAggAggregateOrIndividual**

ATTRIBUTE

APPROPRIATE SYNTAX:

BOOLEAN

BEHAVIOUR DEFINED AS:

A read-only Boolean value indicating whether the Aggregator represents an Aggregate (“TRUE”) or an Individual link (“FALSE”).;

#### **30.7.1.1.7 aAggActorAdminKey**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The current administrative value of the Key for the Aggregator. The administrative Key value may differ from the operational Key value for the reasons discussed in 43.6.2. This is a 16-bit read-write value. The meaning of particular Key values is of local significance.;

#### **30.7.1.1.8 aAggActorOperKey**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The current operational value of the Key for the Aggregator. The administrative Key value may differ from the operational Key value for the reasons discussed in 43.6.2. This is a 16-bit read-only value. The meaning of particular Key values is of local significance.;

#### **30.7.1.1.9 aAggMACAddress**

ATTRIBUTE

APPROPRIATE SYNTAX:

MACAddress

BEHAVIOUR DEFINED AS:

A 6-octet read-only value carrying the individual MAC address assigned to the Aggregator.;

#### **30.7.1.1.10 aAggPartnerSystemID**

ATTRIBUTE

APPROPRIATE SYNTAX:

MACAddress

BEHAVIOUR DEFINED AS:

A 6-octet read-only MAC address value consisting of the unique identifier for the current protocol Partner of this Aggregator. A value of zero indicates that there is no known Partner. If the aggregation is manually configured, this System ID value will be a value assigned by the local System.;

#### 30.7.1.1.11 aAggPartnerSystemPriority

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

A 2-octet read-only value that indicates the priority value associated with the Partner's System ID. If the aggregation is manually configured, this System Priority value will be a value assigned by the local System.;

#### 30.7.1.1.12 aAggPartnerOperKey

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The current operational value of the Key for the Aggregator's current protocol Partner. This is a 16-bit read-only value. If the aggregation is manually configured, this Key value will be a value assigned by the local System.;

#### 30.7.1.1.13 aAggAdminState

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

up  
down

BEHAVIOUR DEFINED AS:

This read-write value defines the administrative state of the Aggregator. A value of "up" indicates that the operational state of the Aggregator (aAggOperState) is permitted to be either up or down. A value of "down" forces the operational state of the Aggregator to be down. Changes to the administrative state affect the operational state of the Aggregator only, not the operational state of the Aggregation Ports that are attached to the Aggregator. A GET opera-

tion returns the current administrative state. A SET operation changes the administrative state to a new value.;

#### 30.7.1.1.14 aAggOperState

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

up  
down

BEHAVIOUR DEFINED AS:

This read-only value defines the operational state of the Aggregator. The operational state is “up” if one or more of the Aggregation Ports that are attached to the Aggregator are collecting, or both collecting and distributing, and if the value of aAggAdminState for the Aggregator is also “up.” If none of the Aggregation Ports that are attached to the Aggregator are collecting and/or distributing, or if there are no Aggregation Ports attached to this Aggregator, then the operational state is “down.” An operational state of “up” indicates that the Aggregator is available for use by the MAC Client; a value of “down” indicates that the Aggregator is not available for use by the MAC Client.;

#### 30.7.1.1.15 aAggTimeOfLastOperChange

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The value of aTimeSinceSystemReset (Annex F.2.1) at the time the interface entered its current operational state. If the current state was entered prior to the last re-initialization of the local network management subsystem, then this object contains a value of zero. This value is read-only.;

#### 30.7.1.1.16 aAggDataRate

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The current data rate, in bits per second, of the aggregate link. The value is calculated as N times the data rate of a single link in the aggregation, where N is the number of active links. This attribute is read-only.;

**30.7.1.1.17 aAggOctetsTxOK**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 1 230 000 counts per second for a single 10 Mb/s aggregation.

BEHAVIOUR DEFINED AS:

A count of the data and padding octets transmitted by this Aggregator on all Aggregation Ports that are (or have been) members of the aggregation. The count does not include octets transmitted by the Aggregator in frames that carry LACPDUs or Marker PDUs (30.7.3.1.7, 30.7.3.1.8, 30.7.3.1.9). However, it includes frames discarded by the Distribution function of the Aggregator (30.7.1.1.25). This value is read-only.;

**30.7.1.1.18 aAggOctetsRxOK**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 1 230 000 counts per second for a single 10 Mb/s aggregation.

BEHAVIOUR DEFINED AS:

A count of the data and padding octets received by this Aggregator, from the Aggregation Ports that are (or have been) members of the aggregation. The count does not include octets received in frames that carry LACP or Marker PDUs (30.7.3.1.2, 30.7.3.1.3, 30.7.3.1.4), or frames discarded by the Collection function of the Aggregator (30.7.1.1.26). This value is read-only.;

**30.7.1.1.19 aAggFramesTxOK**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second for a single 10 Mb/s aggregation.

BEHAVIOUR DEFINED AS:

A count of the data frames transmitted by this Aggregator on all Aggregation Ports that are (or have been) members of the aggregation. The count does not include frames transmitted by the Aggregator that carry LACP or Marker PDUs (30.7.3.1.7, 30.7.3.1.8, 30.7.3.1.9). However, it includes frames discarded by the Distribution function of the Aggregator (30.7.1.1.25). This value is read-only.;

**30.7.1.1.20 aAggFramesRxOK**

ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second for a single 10 Mb/s aggregation.

## BEHAVIOUR DEFINED AS:

A count of the data frames received by this Aggregator, from the Aggregation Ports that are (or have been) members of the aggregation. The count does not include frames that carry LACP or Marker PDUs (30.7.3.1.2, 30.7.3.1.3, 30.7.3.1.4), or frames discarded by the Collection function of the Aggregator (30.7.1.1.26). This value is read-only.;

**30.7.1.1.21 aAggMulticastFramesTxOK**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second for a single 10 Mb/s aggregation.

## BEHAVIOUR DEFINED AS:

A count of the data frames transmitted by this Aggregator on all Aggregation Ports that are (or have been) members of the aggregation, to a group destination address other than the broadcast address. The count does not include frames transmitted by the Aggregator that carry LACP or Marker PDUs (30.7.3.1.7, 30.7.3.1.8, 30.7.3.1.9). However, it includes frames discarded by the Distribution function of the Aggregator (30.7.1.1.25). This value is read-only.;

**30.7.1.1.22 aAggMulticastFramesRxOK**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second for a single 10 Mb/s aggregation.

## BEHAVIOUR DEFINED AS:

A count of the data frames received by this Aggregator, from the Aggregation Ports that are (or have been) members of the aggregation, that were addressed to an active group address other than the broadcast address. The count does not include frames that carry LACP or Marker PDUs (30.7.3.1.2, 30.7.3.1.3, 30.7.3.1.4), or frames discarded by the Collection function of the Aggregator (30.7.1.1.26). This value is read-only.;

**30.7.1.1.23 aAggBroadcastFramesTxOK**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second for a single 10 Mb/s aggregation.

## BEHAVIOUR DEFINED AS:

A count of the broadcast data frames transmitted by this Aggregator on all Aggregation Ports that are (or have been) members of the aggregation. The count does not include frames transmitted by the Aggregator that carry LACP or Marker PDUs (30.7.3.1.7, 30.7.3.1.8, 30.7.3.1.9). However, it includes frames discarded by the Distribution function of the Aggregator (30.7.1.1.25). This value is read-only.;

**30.7.1.1.24 aAggBroadcastFramesRxOK**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second for a single 10 Mb/s aggregation.

## BEHAVIOUR DEFINED AS:

A count of the broadcast data frames received by this Aggregator, from the Aggregation Ports that are (or have been) members of the aggregation. The count does not include frames that carry LACP or Marker PDUs (30.7.3.1.2, 30.7.3.1.3, 30.7.3.1.4), illegal or unknown protocol frames (30.7.3.1.5, 30.7.3.1.6), or frames discarded by the Collection function of the Aggregator (30.7.1.1.26). This value is read-only.;

**30.7.1.1.25 aAggFramesDiscardedOnTx**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second for a single 10 Mb/s aggregation.

## BEHAVIOUR DEFINED AS:

A count of data frames requested to be transmitted by this Aggregator that were discarded by the Distribution function of the Aggregator when conversations are re-allocated to different ports, due to the requirement to ensure that the conversations are flushed on the old ports in order to maintain proper frame ordering (43A.3), or discarded as a result of excessive collisions by ports that are (or have been) members of the aggregation. This value is read-only.;



**30.7.1.1.26 aAggFramesDiscardedOnRx**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second for a single 10 Mb/s aggregation.

BEHAVIOUR DEFINED AS:

A count of data frames, received on all ports that are (or have been) members of the aggregation, that were discarded by the Collection function of the Aggregator as they were received on ports whose Collection function was disabled. This value is read-only.;

**30.7.1.1.27 aAggFramesWithTxErrors**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second for a single 10 Mb/s aggregation.

BEHAVIOUR DEFINED AS:

A count of data frames requested to be transmitted by this Aggregator that experienced transmission errors on ports that are (or have been) members of the aggregation. This count does not include frames discarded due to excess collisions. This value is read-only.;

**30.7.1.1.28 aAggFramesWithRxErrors**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second for a single 10 Mb/s aggregation.

BEHAVIOUR DEFINED AS:

A count of data frames discarded on reception by all ports that are (or have been) members of the aggregation, or that were discarded by the Collection function of the Aggregator, or that were discarded by the Aggregator due to the detection of an illegal Slow Protocols PDU (30.7.3.1.6). This value is read-only.;

**30.7.1.1.29 aAggUnknownProtocolFrames**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second for a single 10 Mb/s aggregation.

## BEHAVIOUR DEFINED AS:

A count of data frames discarded on reception by all ports that are (or have been) members of the aggregation, due to the detection of an unknown Slow Protocols PDU (30.7.3.1.5). This value is read-only.;

**30.7.1.1.30 aAggPortList**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

A SEQUENCE OF INTEGERS that match the syntax of aAggPortID.

## BEHAVIOUR DEFINED AS:

The value of this read-only attribute contains the list of Aggregation Ports that are currently attached to the Aggregator. An empty list indicates that there are no Aggregation Ports attached. Each integer value in the list carries an aAggPortID attribute value (30.7.2.1.1).;

**30.7.1.1.31 aAggLinkUpDownNotificationEnable**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

enabled  
disabled

## BEHAVIOUR DEFINED AS:

When set to “enabled,” Link Up and Link Down notifications are enabled for this Aggregator. When set to “disabled,” Link Up and Link Down notifications are disabled for this Aggregator. This value is read-write.;

**30.7.1.1.32 aAggCollectorMaxDelay**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

INTEGER

## BEHAVIOUR DEFINED AS:

The value of this 16-bit read-write attribute defines the maximum delay, in tens of microseconds, that may be imposed by the Frame Collector between receiving a frame from an Aggregator Parser, and either delivering the frame to its MAC Client or discarding the frame (see 43.2.3.1.1).;

### 30.7.1.2 Aggregator Notifications

#### 30.7.1.2.1 nAggLinkUpNotification

NOTIFICATION

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

When aAggLinkUpDownNotificationEnable is set to “enabled,” a Link Up notification is generated when the Operational State of the aggregator changes from “down” to “up.” When aAggLinkUpDownNotificationEnable is set to “disabled,” no Link Up notifications are generated. The notification carries the identifier of the Aggregator whose state has changed.;

#### 30.7.1.2.2 nAggLinkDownNotification

NOTIFICATION

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

When aAggLinkUpDownNotificationEnable is set to “enabled,” a Link Down notification is generated when the Operational State of the aggregator changes from “up” to “down.” When aAggLinkUpDownNotificationEnable is set to “disabled,” no Link Down notifications are generated. The notification carries the identifier of the Aggregator whose state has changed.;

### 30.7.2 Aggregation Port managed object class

This subclause formally defines the behaviors for the oAggregationPort managed object class attributes.

#### 30.7.2.1 Aggregation Port Attributes

##### 30.7.2.1.1 aAggPortID

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The unique identifier allocated to this Aggregation Port by the local System. This attribute identifies an Aggregation Port instance among the subordinate managed objects of the containing object. This value is read-only.

NOTE—The aAggPortID is represented in the SNMP MIB as an ifIndex—see 30C.4.4.;

**30.7.2.1.2 aAggPortActorSystemPriority**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

A 2-octet read-write value used to define the priority value associated with the Actor's System ID.;

**30.7.2.1.3 aAggPortActorSystemID**

ATTRIBUTE

APPROPRIATE SYNTAX:

MACAddress

BEHAVIOUR DEFINED AS:

A 6-octet read-only MAC address value that defines the value of the System ID for the System that contains this Aggregation Port.;

**30.7.2.1.4 aAggPortActorAdminKey**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The current administrative value of the Key for the Aggregation Port. This is a 16-bit read-write value. The meaning of particular Key values is of local significance.;

**30.7.2.1.5 aAggPortActorOperKey**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The current operational value of the Key for the Aggregation Port. This is a 16-bit read-only value. The meaning of particular Key values is of local significance.;

**30.7.2.1.6 aAggPortPartnerAdminSystemPriority**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

A 2-octet read-write value used to define the administrative value of priority associated with the Partner's System ID. The assigned value is used, along with the value of aAggPortPartnerAdminSystemID, aAggPortPartnerAdminKey, aAggPortPartnerAdminPort, and aAggPortPartnerAdminPortPriority, in order to achieve manually configured aggregation.;

**30.7.2.1.7 aAggPortPartnerOperSystemPriority**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

A 2-octet read-only value indicating the operational value of priority associated with the Partner's System ID. The value of this attribute may contain the manually configured value carried in aAggPortPartnerAdminSystemPriority if there is no protocol Partner.;

**30.7.2.1.8 aAggPortPartnerAdminSystemID**

ATTRIBUTE

APPROPRIATE SYNTAX:

MACAddress

BEHAVIOUR DEFINED AS:

A 6-octet read-write MACAddress value representing the administrative value of the Aggregation Port's protocol Partner's System ID. The assigned value is used, along with the value of aAggPortPartnerAdminSystemPriority, aAggPortPartnerAdminKey, aAggPortPartnerAdminPort, and aAggPortPartnerAdminPortPriority, in order to achieve manually configured aggregation.;

**30.7.2.1.9 aAggPortPartnerOperSystemID**

ATTRIBUTE

APPROPRIATE SYNTAX:

MACAddress

BEHAVIOUR DEFINED AS:

A 6-octet read-only MACAddress value representing the current value of the Aggregation Port's protocol Partner's System ID. A value of zero indicates that there is no known protocol Partner. The value of this attribute may contain the manually configured value carried in aAggPortPartnerAdminSystemID if there is no protocol Partner.;

#### **30.7.2.1.10 aAggPortPartnerAdminKey**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The current administrative value of the Key for the protocol Partner. This is a 16-bit read-write value. The assigned value is used, along with the value of aAggPortPartnerAdminSystemPriority, aAggPortPartnerAdminSystemID, aAggPortPartnerAdminPort, and aAggPortPartnerAdminPortPriority, in order to achieve manually configured aggregation.;

#### **30.7.2.1.11 aAggPortPartnerOperKey**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The current operational value of the Key for the protocol Partner. The value of this attribute may contain the manually configured value carried in aAggPortPartnerAdminKey if there is no protocol Partner. This is a 16-bit read-only value.;

#### **30.7.2.1.12 aAggPortSelectedAggID**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The identifier value of the Aggregator that this Aggregation Port has currently selected. Zero indicates that the Aggregation Port has not selected an Aggregator, either because it is in the process of detaching from an Aggregator or because there is no suitable Aggregator available for it to select. This value is read-only.;

#### **30.7.2.1.13 aAggPortAttachedAggID**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The identifier value of the Aggregator that this Aggregation Port is currently attached to. Zero indicates that the Aggregation Port is not currently attached to an Aggregator. This value is read-only.;

#### **30.7.2.1.14 aAggPortActorPort**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The port number locally assigned to the Aggregation Port. The port number is communicated in LACPDUs as the Actor\_Port. This value is read-only.;

#### **30.7.2.1.15 aAggPortActorPortPriority**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The priority value assigned to this Aggregation Port. This 16-bit value is read-write.;

#### **30.7.2.1.16 aAggPortPartnerAdminPort**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The current administrative value of the port number for the protocol Partner. This is a 16-bit read-write value. The assigned value is used, along with the value of aAggPortPartnerAdminSystemPriority, aAggPortPartnerAdminSystemID, aAggPortPartnerAdminKey, and aAggPortPartnerAdminPortPriority, in order to achieve manually configured aggregation.;

#### **30.7.2.1.17 aAggPortPartnerOperPort**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The operational port number assigned to this Aggregation Port by the Aggregation Port's protocol Partner. The value of this attribute may contain the manually configured value carried in aAggPortPartnerAdminPort if there is no protocol Partner. This 16-bit value is read-only.;

#### **30.7.2.1.18 aAggPortPartnerAdminPortPriority**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The current administrative value of the port priority for the protocol Partner. This is a 16-bit read-write value. The assigned value is used, along with the value of aAggPortPartnerAdminSystemPriority, aAggPortPartnerAdminSystemID, aAggPortPartnerAdminKey, and aAggPortPartnerAdminPort, in order to achieve manually configured aggregation.;

#### **30.7.2.1.19 aAggPortPartnerOperPortPriority**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The priority value assigned to this Aggregation Port by the Partner. The value of this attribute may contain the manually configured value carried in aAggPortPartnerAdminPortPriority if there is no protocol Partner. This 16-bit value is read-only.;

#### **30.7.2.1.20 aAggPortActorAdminState**

ATTRIBUTE

APPROPRIATE SYNTAX:

BIT STRING [SIZE (1..8)]

BEHAVIOUR DEFINED AS:

A string of 8 bits, corresponding to the administrative values of Actor\_State (43.4.2) as transmitted by the Actor in LACPDUs. The first bit corresponds to bit 0 of Actor\_State (LACP\_Activity), the second bit corresponds to bit 1 (LACP\_Timeout), the third bit corresponds to bit 2 (Aggregation), the fourth bit corresponds to bit 3 (Synchronization), the fifth bit corresponds to bit 4 (Collecting), the sixth bit corresponds to bit 5 (Distributing), the seventh bit corresponds to bit 6 (Defaulted), and the eighth bit corresponds to bit 7 (Expired).



These values allow administrative control over the values of LACP\_Activity, LACP\_Timeout, and Aggregation. This attribute value is read-write.;

#### **30.7.2.1.21 aAggPortActorOperState**

ATTRIBUTE

APPROPRIATE SYNTAX:

BIT STRING [SIZE (1..8)]

BEHAVIOUR DEFINED AS:

A string of 8 bits, corresponding to the current operational values of Actor\_State (40.4.2) as transmitted by the Actor in LACPDU. The bit allocations are as defined in 30.7.2.1.20. This attribute value is read-only.;

#### **30.7.2.1.22 aAggPortPartnerAdminState**

ATTRIBUTE

APPROPRIATE SYNTAX:

BIT STRING [SIZE (1..8)]

BEHAVIOUR DEFINED AS:

A string of 8 bits, corresponding to the current administrative value of Actor\_State for the protocol Partner. The bit allocations are as defined in 30.7.2.1.20. This attribute value is read-write. The assigned value is used in order to achieve manually configured aggregation.;

#### **30.7.2.1.23 aAggPortPartnerOperState**

ATTRIBUTE

APPROPRIATE SYNTAX:

BIT STRING [SIZE (1..8)]

BEHAVIOUR DEFINED AS:

A string of 8 bits, corresponding to the current values of Actor\_State in the most recently received LACPDU transmitted by the protocol Partner. The bit allocations are as defined in 30.7.2.1.20. In the absence of an active protocol Partner, this value may reflect the manually configured value aAggPortPartnerAdminState. This attribute value is read-only.;

#### **30.7.2.1.24 aAggPortAggregateOrIndividual**

ATTRIBUTE

APPROPRIATE SYNTAX:

BOOLEAN

BEHAVIOUR DEFINED AS:

A read-only Boolean value indicating whether the Aggregation Port is able to Aggregate (“TRUE”) or is only able to operate as an Individual link (“FALSE”).;

### **30.7.3 Aggregation Port Statistics managed object class**

This subclause formally defines the behaviors for the oAggPortStats managed object class attributes.

#### **30.7.3.1 Aggregation Port Statistics attributes**

##### **30.7.3.1.1 aAggPortStatsID**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

This read-only attribute identifies an Aggregation Port Statistics object instance among the subordinate managed objects of the containing object. The value allocated to this attribute shall be the same as the containing oAggregationPort managed object.;

##### **30.7.3.1.2 aAggPortStatsLACPDUsRx**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 5 counts per second.

BEHAVIOUR DEFINED AS:

The number of valid LACPDUs received on this Aggregation Port. This value is read-only.;

##### **30.7.3.1.3 aAggPortStatsMarkerPDUsRx**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 5 counts per second.

BEHAVIOUR DEFINED AS:

The number of valid Marker PDUs received on this Aggregation Port. This value is read-only.;

#### 30.7.3.1.4 aAggPortStatsMarkerResponsePDUsRx

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 5 counts per second.

BEHAVIOUR DEFINED AS:

The number of valid Marker Response PDUs received on this Aggregation Port. This value is read-only.;

#### 30.7.3.1.5 aAggPortStatsUnknownRx

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 50 counts per second.

BEHAVIOUR DEFINED AS:

The number of frames received that either

- Carry the Slow Protocols Ethernet Type value (43B.4), but contain an unknown PDU, or
- Are addressed to the Slow Protocols group MAC Address (43B.3), but do not carry the Slow Protocols Ethernet Type. This value is read-only.;

#### 30.7.3.1.6 aAggPortStatsIllegalRx

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 50 counts per second.

BEHAVIOUR DEFINED AS:

The number of frames received that carry the Slow Protocols Ethernet Type value (43B.4), but contain a badly formed PDU or an illegal value of Protocol Subtype (43B.4). This value is read-only.;

#### 30.7.3.1.7 aAggPortStatsLACPDUsTx

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresetable counter. This counter has a maximum increment rate of 5 counts per second.

BEHAVIOUR DEFINED AS:

The number of LACPDU transmitted on this Aggregation Port. This value is read-only.;

#### **30.7.3.1.8 aAggPortStatsMarkerPDUsTx**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 5 counts per second.

BEHAVIOUR DEFINED AS:

The number of Marker PDUs transmitted on this Aggregation Port. This value is read-only.;

#### **30.7.3.1.9 aAggPortStatsMarkerResponsePDUsTx**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 5 counts per second.

BEHAVIOUR DEFINED AS:

The number of Marker Response PDUs transmitted on this Aggregation Port. This value is read-only.;

### **30.7.4 Aggregation Port Debug Information managed object class**

This subclause formally defines the behaviors for the oAggPortDebugInformation managed object class attributes.

#### **30.7.4.1 Aggregation Port Debug Information attributes**

##### **30.7.4.1.1 aAggPortDebugInformationID**

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

This read-only attribute identifies an LACP Debug Information object instance among the subordinate managed objects of the containing object. The value allocated to this attribute shall be the same as the containing oAggregationPort managed object.;

### 30.7.4.1.2 aAggPortDebugRxState

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

- current
- expired
- defaulted
- initialize
- lACPDisabled
- portDisabled

BEHAVIOUR DEFINED AS:

This attribute holds the value “current” if the Receive state machine for the Aggregation Port is in the CURRENT state, “expired” if the Receive state machine is in the EXPIRED state, “defaulted” if the Receive state machine is in the DEFAULTED state, “initialize” if the Receive state machine is in the INITIALIZE state, “lACPDisabled” if the Receive state machine is in the LACP\_DISABLED state, or “portDisabled” if the Receive state machine is in the PORT\_DISABLED state. This value is read-only.;

### 30.7.4.1.3 aAggPortDebugLastRxTime

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

The value of aTimeSinceSystemReset (F.2.1) when the last LACPDU was received by this Aggregation Port. This value is read-only.;

### 30.7.4.1.4 aAggPortDebugMuxState

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

- detached
- waiting
- attached
- collecting
- distributing
- collecting\_distributing

## BEHAVIOUR DEFINED AS:

This attribute holds the value “detached” if the Mux state machine (43.4.15) for the Aggregation Port is in the DETACHED state, “waiting” if the Mux state machine for the Aggregation Port is in the WAITING state, “attached” if the Mux state machine for the Aggregation Port is in the ATTACHED state, “collecting” if the Mux state machine for the Aggregation Port is in the COLLECTING state, “distributing” if the Mux state machine for the Aggregation Port is in the DISTRIBUTING state, and “collecting\_distributing” if the Mux state machine for the Aggregation Port is in the COLLECTING\_DISTRIBUTING state. This value is read-only.;

**30.7.4.1.5 aAggPortDebugMuxReason**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

A PrintableString, 255 characters max.

## BEHAVIOUR DEFINED AS:

A human-readable text string indicating the reason for the most recent change of Mux machine state. This value is read-only.;

**30.7.4.1.6 aAggPortDebugActorChurnState**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

noChurn  
churn

## BEHAVIOUR DEFINED AS:

The state of the Actor Churn Detection machine (43.4.17) for the Aggregation Port. A value of “noChurn” indicates that the state machine is in either the NO\_ACTOR\_CHURN or the ACTOR\_CHURN\_MONITOR state, and “churn” indicates that the state machine is in the ACTOR\_CHURN state. This value is read-only.;

**30.7.4.1.7 aAggPortDebugPartnerChurnState**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:

noChurn  
churn

## BEHAVIOUR DEFINED AS:

The state of the Partner Churn Detection machine (43.4.17) for the Aggregation Port. A value of “noChurn” indicates that the state machine is in either the NO\_PARTNER\_CHURN or the PARTNER\_CHURN\_MONITOR state, and “churn” indicates that the state machine is in the PARTNER\_CHURN state. This value is read-only.;

**30.7.4.1.8 aAggPortDebugActorChurnCount**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 5 counts per second.

## BEHAVIOUR DEFINED AS:

Count of the number of times the Actor Churn state machine has entered the ACTOR\_CHURN state. This value is read-only.;

**30.7.4.1.9 aAggPortDebugPartnerChurnCount**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 5 counts per second.

## BEHAVIOUR DEFINED AS:

Count of the number of times the Partner Churn state machine has entered the PARTNER\_CHURN state. This value is read-only.;

**30.7.4.1.10 aAggPortDebugActorSyncTransitionCount**

## ATTRIBUTE

## APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 5 counts per second.

## BEHAVIOUR DEFINED AS:

Count of the number of times the Actor’s Mux state machine (43.4.15) has entered the IN\_SYNC state. This value is read-only.;

**30.7.4.1.11 aAggPortDebugPartnerSyncTransitionCount**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 5 counts per second.

BEHAVIOUR DEFINED AS:

Count of the number of times the Partner's Mux state machine (43.4.15) has entered the IN\_SYNC state. This value is read-only.;

**30.7.4.1.12 aAggPortDebugActorChangeCount**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 5 counts per second.

BEHAVIOUR DEFINED AS:

Count of the number of times the Actor's perception of the LAG ID for this Aggregation Port has changed. This value is read-only.;

**30.7.4.1.13 aAggPortDebugPartnerChangeCount**

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 5 counts per second.

BEHAVIOUR DEFINED AS:

Count of the number of times the Partner's perception of the LAG ID (43.3.6.1) for this Aggregation Port has changed. This value is read-only.;



## 31. MAC Control

### 31.1 Overview

This clause specifies an optional MAC Control Sublayer (MAC Control) for use with the CSMA/CD MAC. MAC Control provides for real-time control and manipulation of MAC sublayer operation. This clause specifies a generalized architecture and protocol for MAC Control. Specific implementations of control functions using this protocol are specified in the normative annexes to this clause. The MAC Control protocol is specified such that it can support new functions to be implemented and added to this standard in the future.

Non-realtime, or quasistatic control (e.g., configuration of MAC operational parameters) is provided by Layer Management.

Operation of the MAC Control sublayer is transparent to the CSMA/CD MAC.

The body of this clause and its associated annexes contain state diagrams, including definitions of variables, constants, and functions. Should there be a discrepancy between a state diagram and descriptive text, the state diagram shall prevail. The notation used in the state diagrams follows the conventions of 21.5.

### 31.2 Layer architecture

The MAC Control sublayer is a client of the CSMA/CD MAC. Figure 31–1 depicts the architectural positioning of the MAC Control sublayer with respect to the CSMA/CD MAC and the MAC Control client. MAC Control clients may include the Bridge Relay Entity, LLC, or other applications.

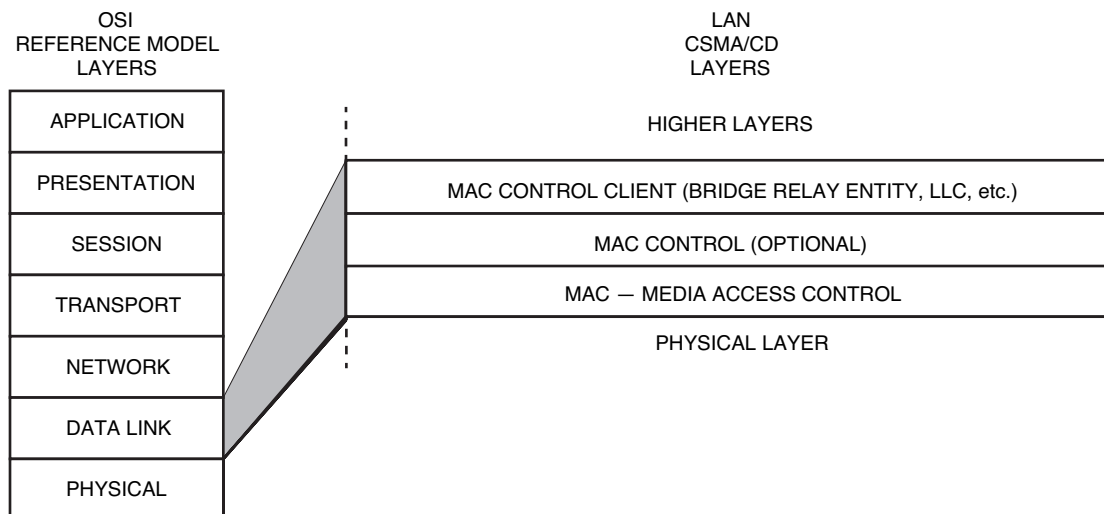


Figure 31–1—Architectural positioning of MAC Control sublayer

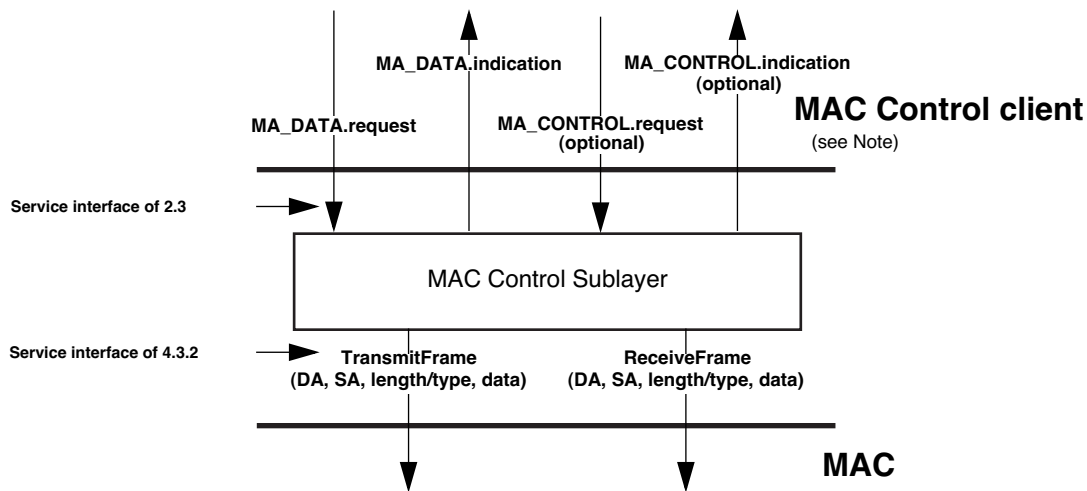
### 31.3 Support by interlayer interfaces

This subclause describes how the MAC Control sublayer uses the service interfaces specified in clauses 2 and 4. In the absence of the optional MAC Control sublayer, the MAC sublayer provides services to its client directly, through the interface specified in 4.3.2.

The optional MAC Control sublayer is inserted between the MAC sublayer and its normal client (i.e., its client in the absence of the MAC Control sublayer). The MAC Control sublayer provides a client service interface to the MAC client as specified in Clause 2, and maps these service primitives to those specified in 4.3.2.

NOTE—In the absence of the MAC Control sublayer, Clause 31 makes no attempt to reconcile the long-standing inconsistencies between the interface definitions in subclauses 4.3.2 and 2.3. These existing inconsistencies have not historically hampered the construction of interoperable networking equipments, and are not sufficiently important to merit further attention.

Figure 31–2 depicts the usage of interlayer interfaces by the MAC Control sublayer. Devices that implement the MAC Control sublayer shall support the optional MAC service primitives, MA\_CONTROL.request and MA\_CONTROL.indication, as specified in 2.3.3 and 2.3.4.



**Figure 31–2—MAC Control sublayer support of interlayer service interfaces**

Clients of the MAC Control sublayer may generate either MA\_CONTROL.request or MA\_DATA.request primitives. MA\_CONTROL.request primitives generated by MAC Control clients are interpreted by the MAC Control sublayer, and may result in the generation of TransmitFrame function calls to the MAC sublayer, or other actions as necessary to support the requested MAC Control sublayer function. Based upon the state of the MAC Control sublayer, MA\_DATA.request primitives may cause the immediate generation of a TransmitFrame function call to the MAC sublayer, or be delayed, discarded, or modified in order to perform the requested MAC Control function.

All frames validly received by the CSMA/CD MAC are passed to the MAC Control sublayer for interpretation. If the frame is destined for the MAC Control client, the MAC Control sublayer generates an MA\_DATA.indication primitive, providing complete transparency for normal data exchange between MAC Control clients. If the frame is destined for the MAC Control sublayer entity, it is interpreted and acted on internal to the MAC Control sublayer. This may result in state changes within the MAC Control sublayer, the generation of MA\_CONTROL.indication primitives, or other actions as necessary to support the MAC Control sublayer function. MAC Control sublayer functions shall always sink MAC Control frames.

Frames destined for the MAC Control sublayer (MAC Control frames) are distinguished from frames destined for MAC Control clients by a unique Length/Type field identifier.

The MAC Control sublayer generates MA\_CONTROL.indication primitives to its client, signaling the current value of internal state variables. Each MAC Control function implemented may have its own function-specific state indications.

### 31.4 MAC Control frames

MAC Control frames comprise MAC client data for the CSMA/CD MAC, as specified in Clause 3. They are encapsulated by the CSMA/CD MAC; that is, they are prepended by a Preamble and Start-of-Frame delimiter and appended by an FCS.

MAC Control frames are distinguished from other MAC frames only by their Length/Type field identifier.

#### 31.4.1 MAC Control frame format

For any particular implementation of this standard, MAC Control frames are fixed length, containing minFrameSize–32 bits. The underlying MAC prepends the Preamble and Start-of-Frame delimiter fields, and appends the FCS. Figure 31–3 depicts the MAC Control frame format.

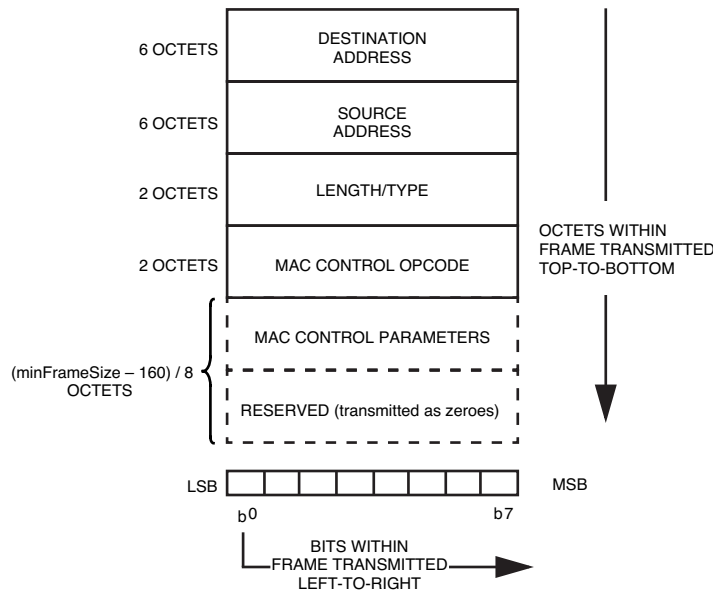


Figure 31–3—MAC Control frame format

##### 31.4.1.1 Destination Address field

The Destination Address field of a MAC Control frame contains the 48-bit address of the station(s) for which the frame is intended. It may be an individual or multicast (including broadcast) address. Permitted values for the Destination Address field may be specified separately for each MAC Control opcode in the annexes to Clause 31.

##### 31.4.1.2 Source Address field

The Source Address field of a MAC Control frame contains the 48-bit individual address of the station sending the frame.

##### 31.4.1.3 Length/Type field

The Length/Type field of a MAC Control frame is a 2-octet field that shall contain the hexadecimal value: 88-08. This value carries the Type interpretation (see 3.2.6), and has been universally assigned for MAC Control of CSMA/CD LANs.

#### 31.4.1.4 MAC Control Opcode field

The MAC Control Opcode field shall contain a 2-octet operation code indicating the MAC Control function. It defines the semantics of the MAC Control Parameters field specified in 31.4.1.5. Annex 31A contains the list of defined MAC Control opcodes and interpretations.

A MAC Control frame shall contain exactly one MAC Control opcode.

#### 31.4.1.5 MAC Control Parameters field

The MAC Control Parameters field shall contain MAC Control opcode-specific parameters. This field may contain none, one, or more parameters as defined by the MAC Control Opcode. The opcode-specific semantics of the MAC Control Parameters field are defined in the normative annex specifying each MAC Control function.

The MAC Control Parameters field shall contain an integral number of octets. The length of the MAC Control Parameters field varies from a minimum of zero, to a maximum of  $\text{minFrameSize} - 160$  bits. See 4.2.3.3 for a discussion of  $\text{minFrameSize}$ .

#### 31.4.1.6 Reserved field

The Reserved field is used when the MAC Control parameters do not fill the fixed length MAC Control frame. The size of the Reserved field, if any, is determined by the size of the MAC Control Parameters field supplied by MAC Control and the minimum frame size parameter of the particular implementation.

The length of Reserved field required for a MAC Control Parameters field that is  $n$  octets long is  $[\text{minFrameSize} - (8 \times n + 160)]$  bits. See 4.2.3.3 for a discussion of  $\text{minFrameSize}$ . The Reserved field is transmitted as all zeroes.

### 31.5 Opcode-independent MAC Control sublayer operation

The MAC Control sublayer generates `ReceiveFrame` function calls continuously to the underlying MAC. The MAC passes to the MAC Control sublayer all valid frames. Invalid frames, as specified in 3.4, are not passed to the MAC Control sublayer in response to a `ReceiveFrame` function call.

#### 31.5.1 Frame parsing and data frame reception

Upon receipt, the MAC Control sublayer parses the incoming frame to determine whether it is destined for the MAC client (Data frame) or for a specific function within the MAC Control sublayer entity itself (MAC Control frame). MAC Control frames with a length of  $\text{minFrameSize}$  and a supported opcode field are interpreted and acted upon by the MAC Control sublayer.

A frame that does not contain the unique Length/Type field specified in 31.4.1.3 is a Data frame. The receipt of a Data frame results in the generation of a `MA_DATA.indication` primitive by the MAC Control sublayer, with the following parameters:

- a) The `destination_address` parameter is set equal to the `destinationParam` from the `ReceiveFrame` function.
- b) The `source_address` parameter is set equal to the `sourceParam` from the `ReceiveFrame` function.
- c) The `m_sdu` parameter is set equal to the concatenation of the `lengthOrTypeParam` and the `dataParam` from the `ReceiveFrame` function.
- d) The `reception_status` parameter is set equal to the `ReceiveStatus` from the `ReceiveFrame` function.

NOTE—For Length/Type field values in the range between `maxValidLength` and `minTypeValue`, the behavior of the `RemovePad` function in the underlying MAC sublayer is unspecified. Frames with Length/Type field values in this range may or may not be passed up by the MAC sublayer.

MAC Control frames with a length greater than minFrameSize and a supported opcode field may be discarded, or truncated to minFrameSize, interpreted, and acted upon. Unsupported MAC Control frames are discarded. Discarded frames are neither passed to the MAC client nor interpreted nor acted upon by the MAC Control sublayer.

### 31.5.2 Control frame reception

Validly received MAC Control frames are further parsed to determine the opcode. The opcode is contained in the first two octets of the dataParam from the ReceiveFrame function. If the MAC Control sublayer entity supports the function requested by the specified opcode, it interprets and acts upon the MAC Control frame in an opcode- and request\_operand-specific manner. (See annexes.) This action may change the state of the MAC Control sublayer, affecting its behavior with respect to data transmission requests by the MAC client, future control frame receptions, or control indications to the MAC client.

If the MAC Control sublayer entity does not support the function requested by the specified opcode, it discards the MAC Control frame. The discard of a frame in this manner may be reported to network management.

### 31.5.3 Opcode-independent MAC Control receive state diagram

The MAC Control sublayer shall implement the receive state machine specified in this subclause.

#### 31.5.3.1 Constants

802.3\_MAC\_Control

The 16-bit Length/Type field value reserved for CSMA/CD MAC Control usage, specified in 31.4.1.3.

#### 31.5.3.2 Variables

receiveEnabled

A boolean set by Network Management to indicate that the station is permitted to receive from the network.

Values: true; Receiver is enabled by management  
false; Receiver is disabled by management

#### 31.5.3.3 Functions

ReceiveFrame

The MAC Sublayer primitive called to receive a frame with the specified parameters.

#### 31.5.3.4 Messages

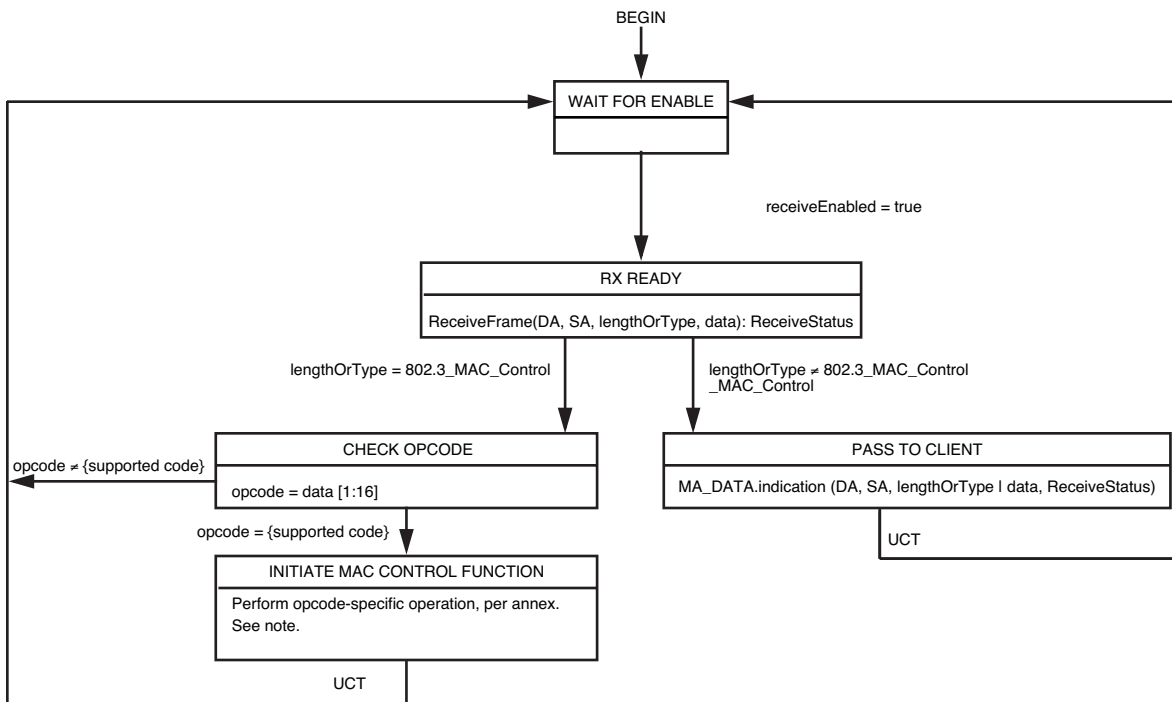
MA\_CONTROL.indication

A signal sent by the MAC Control sublayer signifying a change in the internal sublayer state.

MA\_DATA.indication

The service primitive used by the MAC Control sublayer to pass a received data frame to the MAC Control client with the specified parameters.

### 31.5.3.5 Opcode-independent MAC Control Receive state diagram



NOTE—The opcode-specific operation is launched as a parallel process by the MAC Control sublayer, and not as a synchronous function. Progress of the generic MAC Control Receive state machine (as shown in this figure) is not implicitly impeded by the launching of the opcode-specific function.

**Figure 31–4 – Generic MAC Control Receive state diagram**

The functions performed in the INITIATE MAC CONTROL FUNCTION state are opcode-specific, and are provided in the annexes to Clause 31.

## 31.6 Compatibility requirements

An instantiation of the MAC Control sublayer is not required to implement all valid control functions specified in Annex 31A.

## 31.7 MAC Control client behavior

The MAC Control sublayer uses the services of the underlying connectionless-mode MAC sublayer to exchange both Data and Control frames. The MAC Control sublayer does not provide any mechanism for recovery from lost, discarded, damaged, or delayed frames. It is the responsibility of the MAC Control client to implement mechanisms for dealing with lost, discarded, damaged, and delayed frames, if necessary.

Since implementation of the MAC Control sublayer is optional, a MAC Control client cannot assume the existence of a MAC Control sublayer entity in a peer DTE.

### 31.8 Protocol Implementation Conformance Statement (PICS) proforma for Clause 31, MAC Control<sup>13</sup>

#### 31.8.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Clause 31, the optional MAC Control sublayer, shall complete the following PICS proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

#### 31.8.2 Identification

##### 31.8.2.1 Implementation identification

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification— e.g., name(s) and version(s) for machines and/or operating systems; System Names(s)	
<p>NOTE 1—Only the first three items are required for all implementations, other information may be completed as appropriate in meeting the requirements for the identification.</p> <p>NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier’s terminology (e.g., Type, Series, Model).</p>	

##### 31.8.2.2 Protocol summary

Identification of protocol specification	IEEE Std 802.3-2002 <sup>®</sup> , Clause 31, MAC Control
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
<p>Have any Exception items been required? No [ ] Yes [ ] (See Clause 21; The answer Yes means that the implementation does not conform to IEEE Std 802.3-2002<sup>®</sup>.)</p>	

Date of Statement	
-------------------	--

<sup>13</sup> *Copyright release for PICS proformas:* Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

**31.8.3 PICS proforma for MAC Control frames****31.8.3.1 Support by interlayer interfaces**

Item	Feature	Subclause	Value/comment	Status	Support
SI1	Support for optional MAC service primitives, MA_CONTROL.request and MA_CONTROL.indication	31.3	Required	M	Yes [ ]

**31.8.3.2 MAC Control frame format**

Item	Feature	Subclause	Value/comment	Status	Support
FF1	Length/Type field	31.4.1.3	2-octet field containing 88-08	M	Yes [ ]
FF2	MAC Control opcode	31.4.1.4	2-octet operation code	M	Yes [ ]
FF3	Number of opcodes	31.4.1.4	1	M	Yes [ ]
FF4	MAC Control parameters	31.4.1.5	MAC Control Parameter field as described	M	Yes [ ]

**31.8.3.3 Opcode-independent MAC Control sublayer operation**

Item	Feature	Subclause	Value/comment	Status	Support
SD1	Generic MAC Control receive state diagram	31.5.3	Meets requirements of Figure 31-4	M	Yes [ ]
SD2	MAC Control frame action	31.3	Sink MAC Control frames	M	Yes [ ]

**31.8.3.4 Control opcode assignments**

Item	Feature	Subclause	Value/comment	Status	Support
COA1	Opcode values and interpretations	31A	Reserved opcodes not used	M	Yes [ ]



## 32. Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) sublayer and baseband medium, type 100BASE-T2

NOTE—This PHY is not recommended for new installations.

### 32.1 Overview

The 100BASE-T2 PHY is one of the 100BASE-T family of high-speed CSMA/CD network specifications. The 100BASE-T2 Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA), and baseband medium specifications are aimed at users who want 100 Mb/s performance over basic data grade Category 3 twisted-pair cabling systems. 100BASE-T2 signaling requires two pairs of Category 3 cabling, or cabling with better transfer characteristics than Category 3, installed according to ISO/IEC 11801, as specified in 32.7. This type of cabling, and the connectors used with it, are simple to install and reconfigure.

This clause defines the type 100BASE-T2 PCS, type 100BASE-T2 PMA sublayer, and type 100BASE-T2 Medium Dependent Interface (MDI). Together, the PCS and the PMA sublayer comprise a 100BASE-T2 Physical Layer (PHY). Control actions needed for correct PHY operations are specified by the 100BASE-T2 PHY Control function. Provided in this document are full functional, electrical, and mechanical specifications for the type 100BASE-T2 PHY Control function, PCS, PMA, and MDI. This clause also specifies the baseband medium used with 100BASE-T2.

The objectives of 100BASE-T2 are as follows:

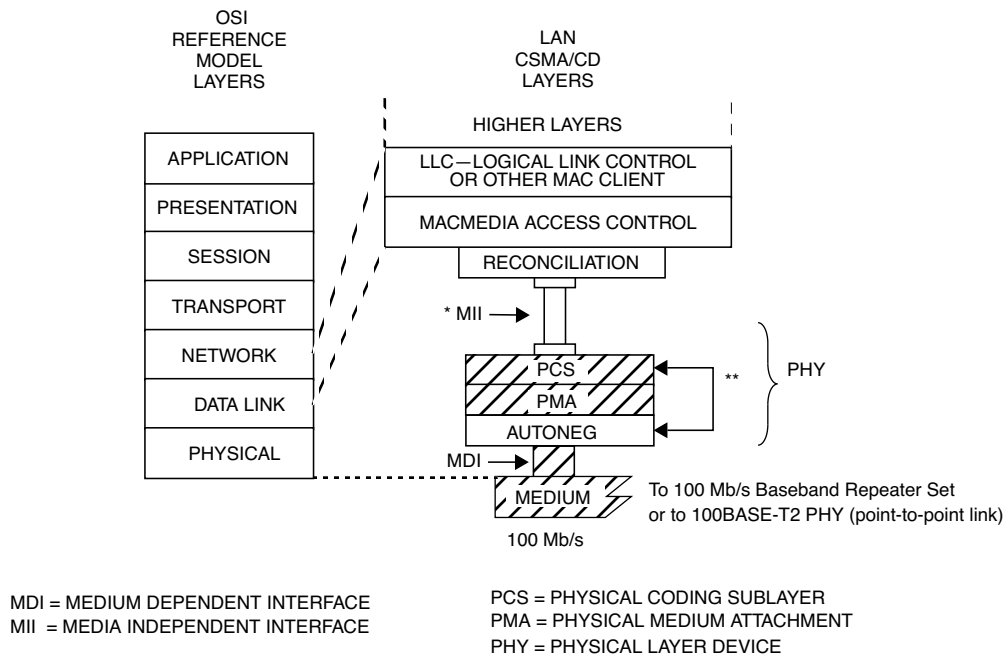
- a) To support the CSMA/CD MAC;
- b) To support the 100BASE-T Media Independent Interface (MII), repeater, and Auto-Negotiation;
- c) To support full duplex operations (Clause 31);
- d) To provide 100 Mb/s data rate at the MII;
- e) To provide for operating over two pairs of Category 3, 4, or 5 balanced twisted-pair cabling systems installed in accordance with ISO/IEC 11801, as specified in 32.7, at distances up to 100 m;
- f) To support operation of other applications on adjacent pairs;
- g) To allow for a nominal network extent of 200 m including
  - 1) Balanced cabling links of 100 m to support both half duplex and full duplex operation and
  - 2) Two-repeater networks of approximately 200 m span;
- h) To provide a communication channel with a symbol error rate of less than one part in  $10^{10}$  at the PMA service interface.

#### 32.1.1 Relation of 100BASE-T2 to other standards

Relations between the 100BASE-T2 PHY and the ISO/IEC Open Systems Interconnection (OSI) Reference Model and the IEEE 802.3® CSMA/CD LAN Model are shown in Figure 32–1. The PHY layers shown in Figure 32–1 connect one Clause 4 Media Access Control (MAC) layer to a Clause 27 repeater. This clause also discusses other variations of the basic configuration shown in Figure 32–1. This whole clause builds on Clauses 1, 2, 3, 4, 21, 22, 27, 28, 29, and 30 of this standard.

#### 32.1.2 Operation of 100BASE-T2

The 100BASE-T2 PHY employs dual-duplex baseband transmission over two wire pairs BI\_DA and BI\_DB, whereby the aggregate data rate of 100 Mb/s is achieved by transmission at a modulation rate of 25



NOTE 1—\* MII is optional.

NOTE 2—\*\* AUTONEG communicates with the PMA sublayer through the PMA service interface messages PMA\_LINK.request and PMA\_LINK.indicate.

**Figure 32-1—Type 100BASE-T2 PHY relationship to the ISO/IEC Open Systems Interconnection (OSI) reference model and the IEEE 802.3<sup>®</sup> CSMA/CD LAN model**

MBd over each wire pair in each direction simultaneously (full duplex transmission), as shown in Figure 32-2. Transmitted symbols are selected from the two-dimensional  $5 \times 5$  symbol constellation illustrated in Figure 32-3. Redundancy in the  $5 \times 5$  constellation allows specific encoding rules to be employed to represent MII data streams, an idle mode or control signals as sequences of two-dimensional symbols. Each two-dimensional symbol can be viewed as a pair  $(A_n, B_n)$  of one-dimensional quinary symbols taken from the set  $\{-2, -1, 0, +1, +2\}$ . Five-level Pulse Amplitude Modulation is employed for transmission over each wire pair (PAM  $5 \times 5$ ). The modulation rate of 25 MBd matches the MII clock rate of 25 MHz. The corresponding symbol period is 40 ns. This specification permits the use of Category 3, 4, or 5 balanced cabling, installed according to ISO/IEC 11801, as defined in 32.7.

A 100BASE-T2 PHY can be configured either as a *master* PHY or as a *slave* PHY. The master-slave relationship between two stations sharing a link segment is established during Auto-Negotiation (see Clause 28, 32.5, Annex 28C, and 32.5.2). The master PHY uses an external clock to determine the timing of transmitter and receiver operations. The slave PHY recovers the clock from the received signal and uses it to determine the timing of transmitter operations, i.e., it performs loop timing, as illustrated in Figure 32-2. In a DTE to repeater connection, the repeater is typically set to be master and the DTE is typically set to be slave.

The following subclauses summarize the PCS, PMA, and PHY Control sections of this document. Figure 32-4 shows the division of responsibilities between the PCS, the PMA sublayer, and the PHY Control function.

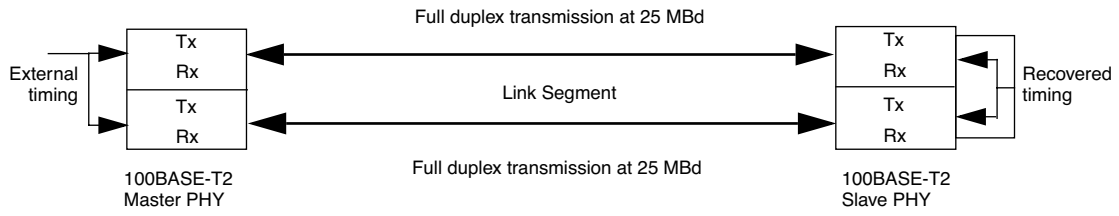


Figure 32-2—100BASE-T2 topology

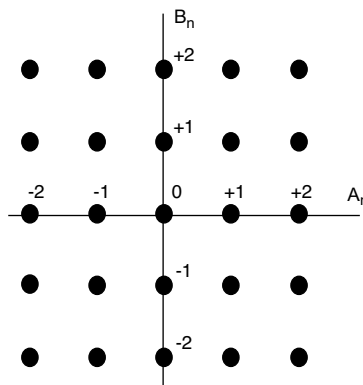
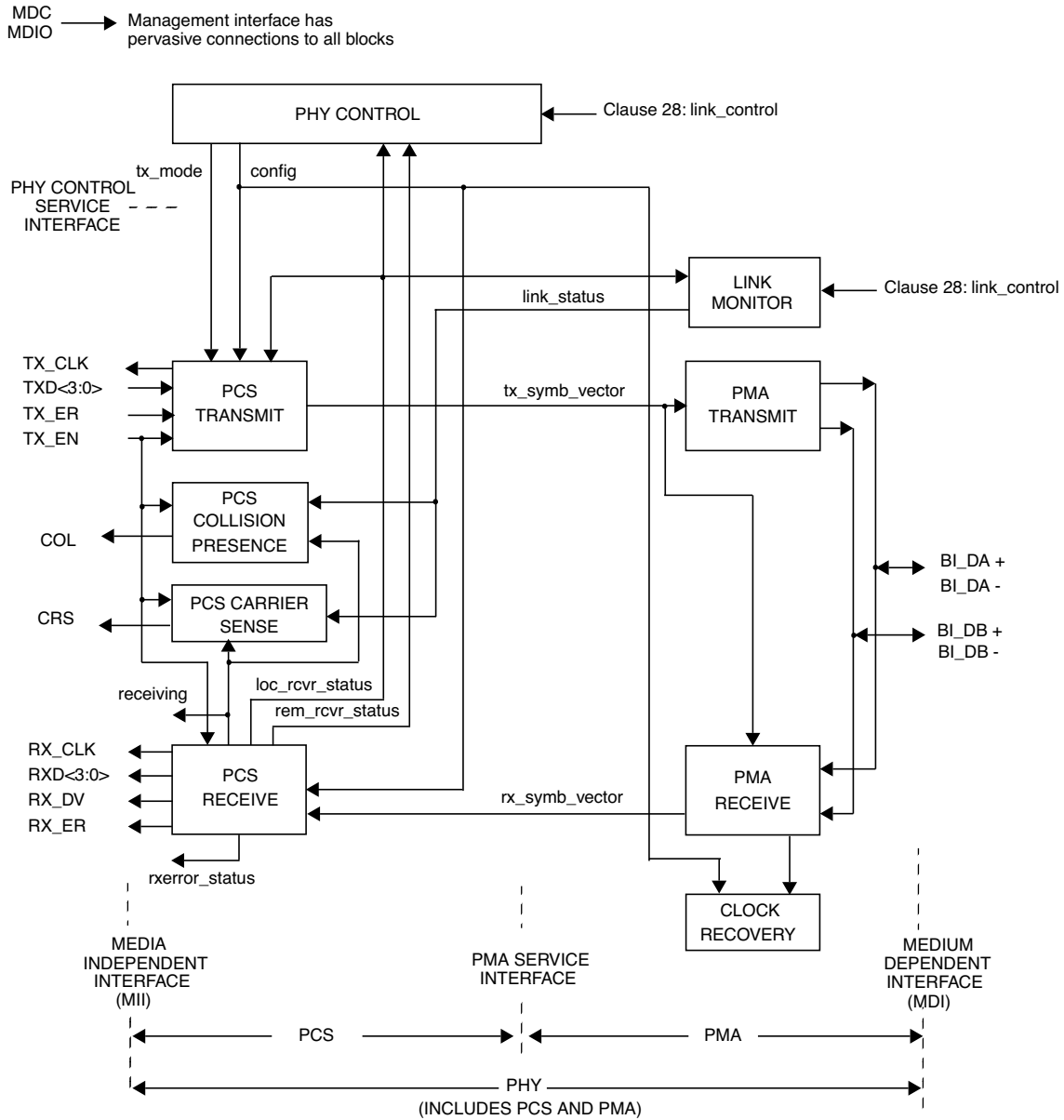


Figure 32-3—PAM5x5 symbol constellation



**Figure 32-4—Division of responsibilities between 100BASE-T2 PCS, PMA, and PHY Control**

### 32.1.2.1 Physical coding sublayer (PCS)

The 100BASE-T2 PCS couples an MII, as described in Clause 22, to a PMA sublayer.

The functions performed by the PCS comprise the generation of continuous quinary symbol sequences to be transmitted over each wire pair. During *data mode*, i.e., when a data stream from the MII is transmitted, the four bits representing the TXD<3:0> data nibble are scrambled by a side-stream scrambler and encoded into a pair of quinary symbols. During *idle mode*, i.e., between transmission of consecutive data streams, the sequences of quinary symbols are generated with an encoding rule that differs from the encoding rule used in data mode. Through this technique, sequences of arbitrary quinary symbols that represent data can easily be distinguished from sequences that represent the idle mode. Furthermore, idle mode encoding takes into account the information of whether the local PHY is operating reliably or not and allows conveying this information to the remote station. A transition from the idle to the data mode is signaled by inserting a Start-of-Stream delimiter that consists of a pattern of two consecutive pairs of quinary symbols. Similarly, the end of a data stream transmission is signaled by inserting an End-of-Stream delimiter that also consists of a pattern of two consecutive pairs of quinary symbols. Further patterns are reserved for signaling a transmit error during transmission of a data stream.

PCS Receive processes pairs of quinary symbols provided by the PMA. It detects the beginning and the end of streams of data and, during the reception of a data stream, descrambles and decodes the received quinary symbol pairs into nibbles RXD<3:0> that are passed to the MII. PCS Receive also detects errors in the received sequences and signals them to the MII. Furthermore, the PCS contains a PCS Carrier Sense function, a PCS Collision Presence function, and a management interface.

The PCS functions and state diagrams are specified in 32.3. The signals provided by the PCS at the MII conform to the interface requirements of Clause 22. The PCS interfaces to PHY Control and to the PMA are abstract message-passing interfaces specified in 32.2 and 32.4, respectively.

### 32.1.2.2 PMA sublayer

The PMA couples messages from the PMA service interface onto the balanced cabling physical medium. The PMA provides dual-duplex communications at 25 MBd over two pairs of balanced cabling up to 100 m in length.

The PMA Transmit function comprises two independent transmitters to generate five-level pulse-amplitude modulated signals on each of the two pairs BI\_DA and BI\_DB, as described in 32.4.1.1.2.

The PMA Receive function comprises two independent receivers for five-level pulse-amplitude modulated signals on each of the two pairs BI\_DA and BI\_DB, as described in 32.4.1.1.3. The receivers are responsible for acquiring clock, and providing quinary symbol pairs to the PCS as defined by the PMA\_UNITDATA.indicate message. The PMA also contains functions for Link Monitor.

PMA functions and state diagrams are specified in 32.4. PMA electrical specifications are given in 32.6.

### 32.1.2.3 PHY Control function

PCS and PMA sublayer operations are controlled via signals generated by the PHY Control function. PHY Control does not itself represent a sublayer but rather a logical grouping of the control functions necessary for proper operations of a 100BASE-T2 transceiver.

PHY Control determines whether the PHY should operate in a normal state, where packets can be exchanged over the link segment, or whether the PHY should be forced to send quinary symbol sequences that represent the idle mode. The latter occurs when either one of the PHYs, or both PHYs, that share a link segment are not operating reliably.

The PHY Control function and state diagram are specified in 32.2, prior to introducing the PCS and PMA functional specifications. The PHY Control interface to the PCS and PMA sublayer is an abstract message-passing interface also specified in 32.2.

### **32.1.3 Application of 100BASE-T2**

#### **32.1.3.1 Compatibility considerations**

All implementations of the balanced cabling link shall be compatible at the MDI. The PCS, PMA, and the medium are defined to provide compatibility among devices designed by different manufacturers. Designers are free to implement circuitry within the PCS and PMA in an application-dependent manner provided the MDI and MII specifications are met.

#### **32.1.3.2 Incorporating the 100BASE-T2 PHY into a DTE**

When the PHY is incorporated within the physical bounds of a DTE, conformance to the MII is optional, provided that the observable behavior of the resulting system is identical to that of a system with a full MII implementation. For example, an integrated PHY may incorporate an interface between PCS and MAC that is logically equivalent to the MII, but does not have the full output current drive capability called for in the MII specification.

#### **32.1.3.3 Use of 100BASE-T2 PHY for point-to-point communication**

The 100BASE-T2 PHY, in conjunction with the MAC specified in Clauses 1 through 4 (including parameterized values in 4.4.2.3 to support 100 Mb/s operation) may be used at both ends of a link for point-to-point applications between two DTEs. Such a configuration does not require a repeater. In this case each PHY may connect through an MII to its respective DTE. Optionally, either PHY (or both PHYs) may be incorporated into the DTEs without an exposed MII.

#### **32.1.3.4 Auto-Negotiation requirement**

Full Auto-Negotiation, as specified in Clause 28, shall be included in all compliant implementations.

### **32.1.4 State diagram conventions**

The body of this clause and its associated annexes contain state diagrams, including definitions of variables, constants and functions. Should there be a discrepancy between a state diagram and descriptive text, the state diagram shall prevail. The notation used in the state diagrams follows the conventions of 21.5.

## **32.2 PHY Control functional specifications and service interface**

### **32.2.1 PHY Control function**

PHY Control generates the control actions that are needed to bring the PHY in a mode of operation during which packets can be exchanged with the link partner. PHY Control shall comply with the state diagram description given in Figure 32–5.

During Auto-Negotiation, PHY Control ensures that the transmitter is disabled. When the Auto-Negotiation process asserts `link_control=ENABLE`, PHY Control enters the TRAINING state. During training, PHY Control enforces transmission in the idle mode by setting `tx_mode=SEND_I` and the PHY transmits sequences of quinary symbols encoded with the parameter `loc_rcvr_status=NOT_OK`. When the PHY achieves successful training and establishes proper receiver operations, PCS Receive asserts the parameter `loc_rcvr_status=OK`, and PCS Transmit conveys this information to the link partner via idle transmission.

The criterion for assertion of the parameter `loc_rcvr_status` is left to the implementor. It can be based, for example, on observing the mean-square error at the decision point of the receiver or detecting errors during reception of symbol streams that represent the idle mode. Upon observation that the remote PHY also operates reliably (`rem_rcvr_status=OK`), the normal mode of operation is entered where transmission of packets over the link segment can take place.

The normal mode of operation corresponds to the SEND IDLE OR DATA state, where PHY Control asserts `tx_mode=SEND_N`. In this state, when no packets have to be sent, idle mode transmission takes place. Encoding of quinary symbols is realized with the parameter `loc_rcvr_status = OK`.

If during the normal mode of operation unsatisfactory receiver operations is detected (`loc_rcvr_status=NOT_OK`), transmission of the current packet, if any, is completed and PHY Control enters the TRAINING state.

Whenever a PHY that operates reliably detects unsatisfactory operation of the remote PHY (`rem_rcvr_status=NOT_OK`), it enters the SEND IDLE state where `tx_mode=SEND_I` is asserted and idle transmission takes place. In this state, encoding is performed with the parameter `loc_rcvr_status=OK`. As soon as the remote PHY signals satisfactory receiver operation (`rem_rcvr_status=OK`), the SEND IDLE OR DATA state is entered. Note that if in the SEND IDLE state `loc_rcvr_status` takes the value NOT\_OK transition to the TRAINING state occurs.

PHY Control may force the transmit scrambler state to be initialized to a random value by requesting the execution of the PCS Reset function defined in 32.3.1.1.

### 32.2.2 PHY Control Service interface

The following specifies the services provided by PHY Control. These services are described in an abstract manner and do not imply any particular implementation.

The following primitives are defined:

```
PHYC_CONFIG.indicate
PHYC_TXMODE.indicate
PHYC_RXSTATUS.request
PHYC_REMRXSTATUS.request
```

The parameter `link_control` is identical to the `link_control` parameter defined for the PMA Service interface in 32.4.2.4.

#### 32.2.2.1 PHYC\_CONFIG.indicate

Each PHY in a 100BASE-T2 link is configured as a master PHY or as a slave PHY. Master/slave configuration is determined during Auto-Negotiation (see 32.5). The result of this negotiation is provided to PHY Control.

##### 32.2.2.1.1 Semantics of the primitive

```
PHYC_CONFIG.indicate (config)
```

`PHYC_CONFIG.indicate` specifies to PCS and PMA Clock Recovery via the parameter `config` whether the PHY must operate as a master PHY or as a slave PHY. The parameter `config` can take on one of two values of the form:

```
MASTER      This value is continuously asserted when the PHY must operate as a master PHY.
```

SLAVE            This value is continuously asserted when the PHY must operate as a slave PHY.

### 32.2.2.1.2 When generated

PHY Control shall generate PHYC\_CONFIG.indicate messages synchronously with every MII TX\_CLK cycle.

### 32.2.2.1.3 Effect of receipt

Upon reception of this primitive, PCS and PMA Clock Recovery shall perform their functions in master or slave configuration according to the value assumed by the parameter config.

### 32.2.2.2 PHYC\_TXMODE.indicate

The transmitter in a 100BASE-T2 link normally sends over the two pairs quinary symbols that can represent an MII data stream or the idle mode.

#### 32.2.2.2.1 Semantics of the primitive

PHYC\_TXMODE.indicate (tx\_mode)

PHYC\_TXMODE.indicate specifies to PCS Transmit via the parameter tx\_mode what sequence of quinary symbols the PCS should be transmitting. The parameter tx\_mode can take on one of two values of the form:

SEND_N	This value is continuously asserted when transmission of sequences of quinary symbols representing an MII data stream or the idle mode is to take place.
SEND_I	This value is continuously asserted in case transmission of sequences of quinary symbols representing the idle mode is to take place.

#### 32.2.2.2.2 When generated

PHY Control shall generate PHYC\_TXMODE.indicate messages synchronously with every MII TX\_CLK cycle.

#### 32.2.2.2.3 Effect of receipt

Upon receipt of this primitive, the PCS shall perform its Transmit function as described in 32.3.1.2.

### 32.2.2.3 PHYC\_RXSTATUS.request

This primitive is generated by PCS Receive to communicate the status of the receive link for the local PHY. The parameter loc\_rcvr\_status conveys to PHY Control and Link Monitor the information on whether the status of the overall receive link is satisfactory or not. Note that loc\_rcvr\_status is used by PCS Transmit encoding functions.

#### 32.2.2.3.1 Semantics of the primitive

PHYC\_RXSTATUS.request (loc\_rcvr\_status)

The loc\_rcvr\_status parameter can take on one of two values of the form:

OK	This value is asserted and remains true during reliable operation of the receive link for the local PHY.
NOT_OK	This value is asserted whenever operation of the receive link for the local PHY is unreliable.



### 32.2.2.3.2 When generated

PCS Receive shall generate PHYC\_RXSTATUS.request messages synchronously with signals received at the MDI. It shall prevent that the value of the parameter `loc_rcvr_status` is modified while `TX_EN=1` in order to avoid that a transition from data to idle mode or from idle to data mode occurs while a packet is being presented to the PCS at the MII.

### 32.2.2.3.3 Effect of receipt

The effect of receipt of this primitive is specified in 32.2.5 and 32.4.1.3.3.

### 32.2.2.4 PHYC\_REMRXSTATUS.request

This primitive is generated by PCS Receive to indicate the status of the receive link as communicated by the remote PHY. The parameter `rem_rcvr_status` conveys to PHY Control the information on whether reliable operation of the remote PHY is detected or not.

#### 32.2.2.4.1 Semantics of the primitive

PHYC\_REMRXSTATUS.request (`rem_rcvr_status`)

The `rem_rcvr_status` parameter can take on one of two values of the form:

OK	The receive link for the remote PHY is operating reliably.
NOT_OK	Reliable operation of the receive link for the remote PHY is not detected.

#### 32.2.2.4.2 When generated

The PCS shall generate PHYC\_REMRXSTATUS.request messages synchronously with signals received at the MDI.

#### 32.2.2.4.3 Effect of receipt

The effect of receipt of this primitive is specified in 32.2.5.

### 32.2.3 State diagram variables

`link_control`

See 32.4.1.3.1.

`link_status`

See 32.4.1.3.1.

`loc_rcvr_status`

Variable set by the PCS Receive function to indicate correct or incorrect operation of the receive link for the local PHY.

Values: OK: the receive link for the local PHY is operating reliably,  
NOT\_OK: operation of the receive link for the local PHY is not reliable.

`pma_reset`

Allows reset of all PMA functions.

Values: ON and OFF

Set by: PMA Reset

rem\_rcvr\_status

Variable set by the PCS Receive function to indicate whether correct operation of the receive link for the remote PHY is detected or not.

Values: OK: the receive link for the remote PHY is operating reliably,  
 NOT\_OK: reliable operation of the receive link for the remote PHY is not detected.

tx\_mode

PCS Transmit shall send quinary symbols according to the value assumed by this variable.

Values: SEND\_N: transmission of sequences of quinary symbols representing an MII data stream, the idle mode, or control signals shall take place,  
 SEND\_I: transmission of sequences of quinary symbols representing the idle mode shall take place.

**32.2.4 State diagram timers**

All timers operate in the manner described in 14.2.3.2 with the following addition. A timer is reset and stops counting upon entering a state where “stop x\_timer” is asserted.

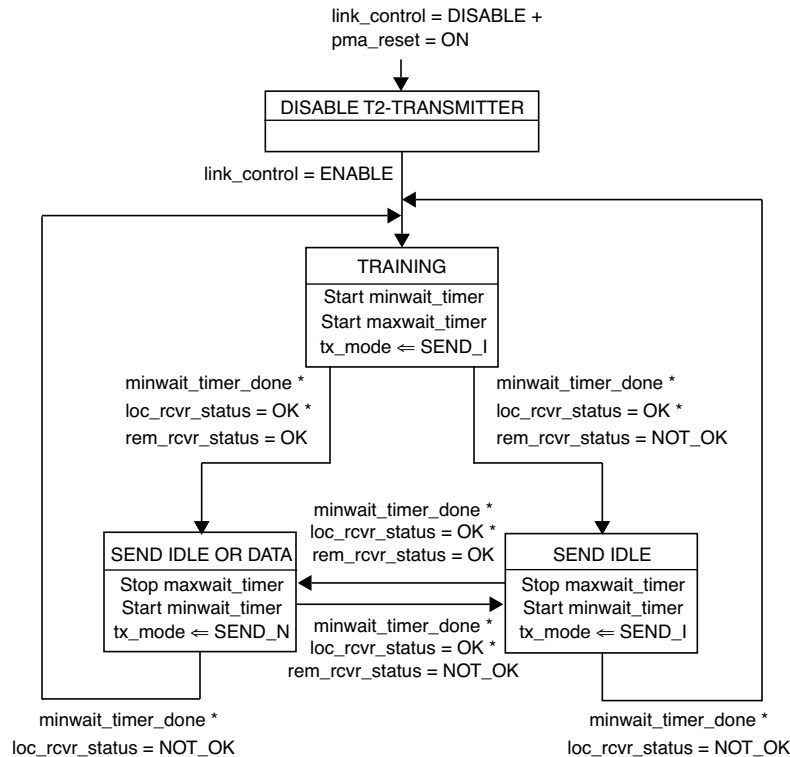
maxwait\_timer

A timer used to measure the amount of time during which a receiver dwells in the TRAINING state. The timer shall expire 2500–3000 ms after entering the TRAINING state.

minwait\_timer

A timer used to measure the amount of time during which a receiver waits in the SEND IDLE OR DATA, the TRAINING, or the SEND IDLE state before being allowed to leave the current state. The timer shall expire 128T = 5.12 μs after being started.

**32.2.5 PHY Control state diagram**



**Figure 32–5—PHY Control state diagram**

### 32.3 PCS functional specifications

The PCS comprises one PCS Reset function and four simultaneous and asynchronous operating functions. The PCS operating functions are: PCS Transmit, PCS Receive, PCS Carrier Sense, and PCS Collision Presence. All operating functions start immediately after the successful completion of the PCS Reset function.

The PCS reference diagram, Figure 32–5, shows how the four operating functions relate to the messages of the PCS-PMA and the PCS-PHY Control interfaces. Connections from the management interface (signals MDC and MDIO) to other layers are pervasive, and are not shown in Figure 32–5. The management functions are specified in Clause 30. See also Figure 32–7, which presents a block diagram helpful for understanding the definitions of PCS Transmit function variables, and Figure 32–11, which defines the structure of frames passed from PCS to PMA.

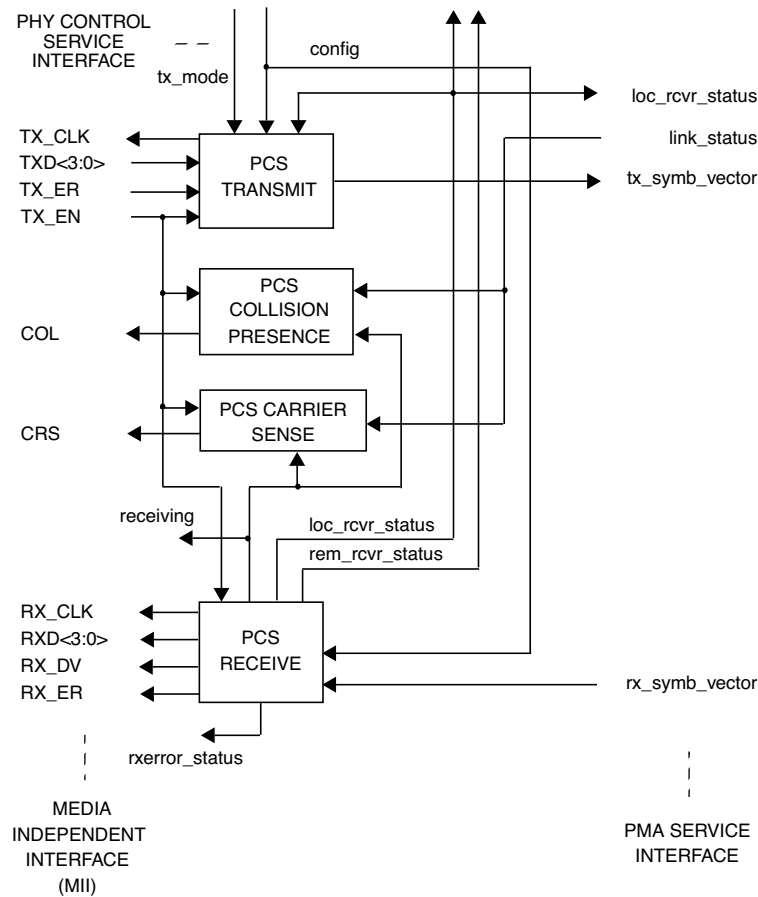


Figure 32-6—PCS reference diagram

### 32.3.1 PCS functions

#### 32.3.1.1 PCS Reset function

The PCS Reset function shall be executed any time one of three conditions occurs. These three conditions are “power on,” the receipt of a request for reset from the management entity, and the receipt of a request for reset from PHY Control. PCS Reset initializes all PCS functions. PCS Reset sets pcs\_reset = ON for the duration of its Reset function. All state diagrams take the open-ended pcs\_reset branch upon execution of PCS Reset. The reference diagrams do not explicitly show the PCS Reset function.

#### 32.3.1.2 PCS Transmit function

The PCS Transmit function shall conform to the PCS Transmit state diagram in Figure 32–12.

In normal mode of operation, the tx\_mode parameter, which is transferred from PHY Control to the PCS via the PHYC\_TXMODE.indicate message, assumes the value tx\_mode=SEND\_N, and the PCS Transmit function generates at each symbol period pairs of quinary symbols that represent data or the idle mode. A symbol period T is equal to 40 ns. A time index n, where n is an integer, is introduced to establish a temporal relationship between different symbol periods. The tx\_symb\_vector parameter at time n is a two-element vector of quinary symbols ( $A_n$ ,  $B_n$ ) that is transferred to the PMA via PMA\_UNITDATA.request. The PMA shall transmit symbols  $A_n$  and  $B_n$  over wire pairs BI\_DA and BI\_DB, respectively. During transmission of data, the four bits representing the TXD<3:0> data nibble are scrambled by the OCS using a side-stream scrambler then encoded into a pair of quinary symbols and transferred to the PMA. The idle mode is signaled by a sequence of pairs of quinary symbols that are also generated using the side-stream transmit scrambler. However, the encoding rules by which the quinary symbols are obtained are different for the data and the idle modes. This allows, at the receiver, sequences of quinary symbol pairs that represent data to be distinguished from sequences of quinary symbol pairs that represent the idle mode. A transition from the idle mode to the data mode is signalled by inserting a Start-of-Stream delimiter that consists of a pattern of two consecutive pairs of quinary symbols. Similarly, the end of transmission of data is signalled by an End-of-Stream delimiter that also consists of a pattern of two consecutive pairs of quinary symbols. Further patterns are reserved for signaling the assertion of TX\_ER within a stream of data.

If tx\_mode = SEND\_I is asserted, PCS Transmit generates sequences of symbol pairs ( $A_n$ ,  $B_n$ ) according to the encoding rule in idle mode. Idle mode encoding takes into account the value of the parameter loc\_rcvr\_status. By this mechanism, a PHY indicates during idle transmission the status of its own receiver to the link partner.

The PCS Transmit reference diagram is shown in Figure 32–7. PCS encoding involves the generation of the three-bit words  $Sa_n[2:0]$ ,  $Sb_n[2:0]$ ,  $Ta_n[2:0]$ , and  $Tb_n[2:0]$ , from which the pairs of quinary symbols ( $A_n$ ,  $B_n$ ) are obtained. The three-bit words  $Sa_n[2:0]$  and  $Sb_n[2:0]$  are determined first, as explained in 32.3.1.2.2, from sequences of random binary symbols derived from the transmit side-stream scrambler. The generation of  $Ta_n[2:0]$  and  $Tb_n[2:0]$  and the quinary symbols  $A_n$  and  $B_n$  is given in 32.3.1.2.3. The physical structure represented in Figure 32–7 is not required. Implementors are free to construct any logical devices having functionality identical to that described by the following specifications and the PCS Transmit state diagram.

##### 32.3.1.2.1 Side-stream scrambler polynomials

The PCS Transmit function employs side-stream scrambling. If the parameter config provided to the PCS by the PHY Control function via the PHYC\_CONFIG.indicate message assumes the value MASTER, PCS Transmit shall employ

$$g_M(x) = 1 + x^{13} + x^{33}$$

as transmitter side-stream scrambler generator polynomial. If config = SLAVE, PCS Transmit shall employ

$$g_S(x) = 1 + x^{20} + x^{33}$$

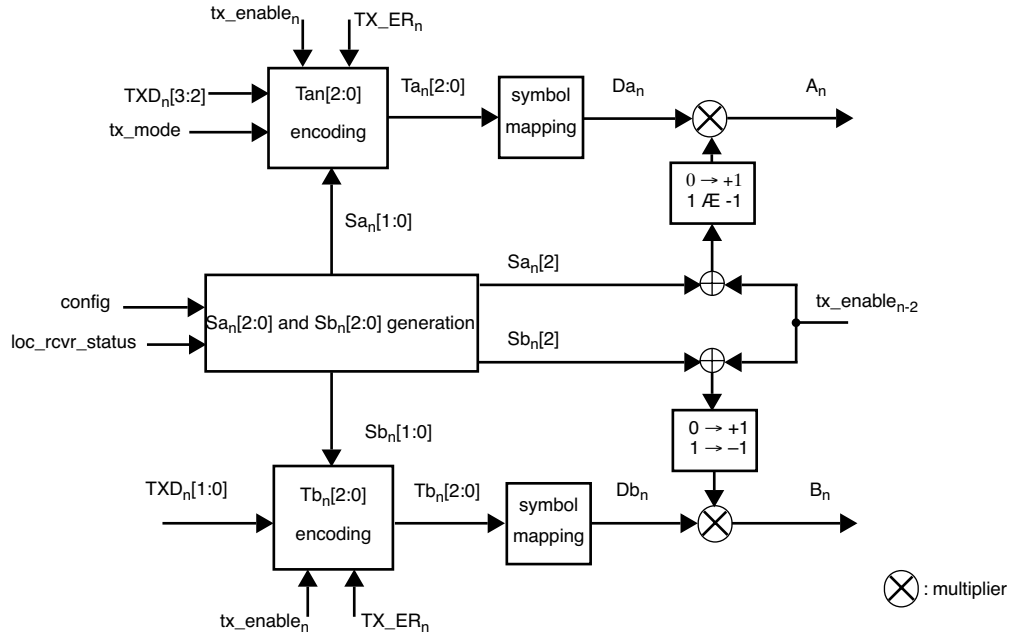
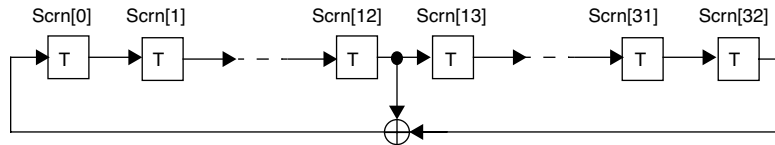
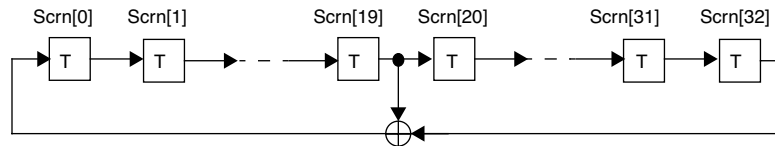


Figure 32-7—PCS Transmit reference diagram

as transmitter side-stream scrambler generator polynomial. The implementation of master and slave PHY side-stream scramblers by linear-feedback shift registers is shown in Figure 32-8. The bits stored in the shift register delay line at time  $n$  are denoted by  $Scr_n[32:0]$ . At each symbol period, the shift register is advanced by one bit and one new bit represented by  $Scr_n[0]$  is generated. The transmitter side-stream scrambler is reset upon execution of the PCS Reset function. If PCS Reset is executed, all bits of the 33-bit vector representing the side-stream scrambler state are randomly set. The generation of the random bits is left to the implementor.



a) Side-stream scrambler employed by the master PHY



b) Side-stream scrambler employed by the slave PHY

Figure 32-8—Realization of side-stream scramblers by linear feedback shift registers

### 32.3.1.2.2 Generation of bits $Sa_n[2:0]$ and $Sb_n[2:0]$

PCS Transmit encoding rules are based on the generation, at time  $n$ , of the four bits  $Sx_n$ ,  $Sy_n$ ,  $Sa_n[2]$ , and  $Sb_n[2]$ . These four bits are mutually uncorrelated and also uncorrelated with the bit  $Scr_n[0]$  in data and idle

modes. For both master and slave PHYs, they are obtained by the same linear combinations of bits stored in the transmit scrambler shift register delay line. The four bits are elements of the same maximum-length shift register sequence of length  $2^{33} - 1$  as  $Scr_n[0]$ , but shifted in time. The associated delays are all large and different, such that there is no apparent correlation among the five bits  $Scr_n[0]$ ,  $Sx_n$ ,  $Sy_n$ ,  $San[2]$ , and  $Sbn[2]$ . The bits  $Sx_n$  and  $Sy_n$  are given by

$$Sx_n = Scr_n[3] \oplus Scr_n[8]$$

$$Sy_n = Scr_n[4] \oplus Scr_n[6]$$

where  $\oplus$  denotes the XOR logic operator. Four bits  $X_n[1:0]$  and  $Y_n[1:0]$  are obtained by

$$X_n[0] = \begin{cases} Sx_n & \text{if } \{n - n_0 = 0 \pmod{2} \text{ or } \text{loc\_rcvr\_status} = \text{OK} \} \\ Sx_n \oplus 1 & \text{else} \end{cases}$$

$$X_n[1] = X_n[0] \oplus 1$$

$$Y_n[0] = \begin{cases} Sy_n & \text{if } n - n_0 = 0 \pmod{2} \\ Sy_{n-1} \oplus 1 & \text{else} \end{cases}$$

$$Y_n[1] = Y_n[0]$$

where  $n_0$  denotes the time index of the last transmitter side-stream scrambler reset.

The bits  $San[2:0]$  and  $Sbn[2:0]$  are given by

$$San[2] = Scr_n[1] \oplus Scr_n[5]$$

$$San[1:0] = \begin{cases} X_n[1:0] & \text{if } Scr_n[0] = 1 \\ Y_n[1:0] & \text{else} \end{cases}$$

$$Sbn[2] = Scr_n[2] \oplus Scr_n[12]$$

$$Sbn[1:0] = \begin{cases} Y_n[1:0] & \text{if } Scr_n[0] = 1 \\ X_n[1:0] & \text{else} \end{cases}$$

### 32.3.1.2.3 Generation of sequences $A_n$ and $B_n$

If  $tx\_mode = SEND\_N$ , the PCS Transmit function generates sequences  $A_n$  and  $B_n$  that represent either a stream of data or an idle mode. If  $tx\_mode = SEND\_I$ , idle transmission is enforced. The encoding rule is determined by the value of the signal  $tx\_enable_n$ , given by

$$\text{tx\_enable}_n = \begin{cases} \text{TX\_EN}_n & \text{if } \text{tx\_mode} = \text{SEND\_N} \\ 0 & \text{else} \end{cases}$$

where  $\text{TX\_EN}_n$  represents the MII signal  $\text{TX\_EN}$  at time  $n$ . If  $\text{tx\_enable}_n = 1$ , transmission of a stream of data takes place. As illustrated in Figure 32–11, the definition of a Start-of-Stream Delimiter (“SSD”) is related to the condition  $\text{SSD}_n = (\text{tx\_enable}_n) * (!\text{tx\_enable}_{n-2}) = 1$ , where “\*” and “!” denote the logic AND and NOT operators, respectively. For the generation of “SSD”, PCS Transmit replaces the first two nibbles of the preamble in a data stream with the symbols defined below. Similarly, the definition of an End-of-Stream Delimiter (ESD) is related to the condition  $\text{ESD}_n = (!\text{tx\_enable}_n) * (\text{tx\_enable}_{n-2}) = 1$ . This occurs during the first two symbol periods after transmission of the last nibble of a data stream.

The symbols  $A_n$  and  $B_n$  are obtained from the three-bit words  $Ta_n[2:0]$  and  $Tb_n[2:0]$  whose definitions in the data and the idle modes are given below.

#### **Data mode ( $\text{tx\_enable}_n=1$ ):**

*Definition of “SSD”:*

$$Ta_n[2:0] = [1, 0, 0]$$

$$Tb_n[2:0] = [!\text{tx\_enable}_{n-1}, \text{tx\_enable}_{n-2}, \text{tx\_enable}_{n-2}]$$

A most significant bit  $Ta_n[2]=1$  or  $Tb_n[2]=1$  results in the transmission of a symbol that can be interpreted as an ESC symbol.

*Encoding of data nibbles:*

$$Ta_n[2:0] = [0, (Sa_n[1] \oplus \text{TXD}_n[3]), (Sa_n[0] \oplus \text{TXD}_n[2] \oplus 1)]$$

$$Tb_n[2:0] = [0, (Sb_n[1] \oplus \text{TXD}_n[1]), (Sb_n[0] \oplus \text{TXD}_n[0] \oplus 1)]$$

where  $\text{TXD}_n[3:0]$  denotes the data nibble  $\text{TXD}[3:0]$  at time  $n$ .

*Encoding of error indication:*

If  $\text{TX\_ER}_n=1$  is asserted, where  $\text{TX\_ER}_n$  denotes the value of the signal  $\text{TX\_ER}$  at time  $n$ , error indication is signaled by means of the ESC and 0 symbols. The encoding rule is given by

$$Ta_n[2:0] = [(Sa_n[1] \oplus Sa_n[0]), 0, 0]$$

$$Tb_n[2:0] = [(Sb_n[1] \oplus Sb_n[0]), 0, 0]$$

#### **Idle mode ( $\text{tx\_enable}_n=0$ ):**

*Definition of “ESD”:*

$$Ta_n[2:0] = [1, 0, 0]$$

$$Tb_n[2:0] = [\text{tx\_enable}_{n-1}, !\text{tx\_enable}_{n-2}, !\text{tx\_enable}_{n-2}]$$

*Encoding in the idle mode:*

$$Ta_n[2:0] = [0, Sa_n[1], Sa_n[0]]$$

$$Tb_n[2:0] = [0, Sb_n[1], Sb_n[0]]$$

$$tx\_enable_n = \begin{cases} 0 & \text{“”} \\ \text{else} & \end{cases}$$

where TX\_EN<sub>n</sub> represents the MII signal TX\_EN at time n. If tx\_enable<sub>n</sub> = 1, transmission of a stream of data takes place. As illustrated in Figure 32–11, the definition of a Start-of-Stream Delimiter (“SSD”) is related to the condition SSD<sub>n</sub> = (tx\_enable<sub>n</sub>) \* (! tx\_enable<sub>n-2</sub>) = 1, where “\*” and “!” denote the logic AND and NOT operators, respectively. For the generation of “SSD”, PCS Transmit replaces the first two nibbles of the preamble in a data stream with the symbols defined below. Similarly, the definition of an End-of-Stream Delimiter (ESD) is related to the condition ESD<sub>n</sub> = (! tx\_enable<sub>n</sub>) \* (tx\_enable<sub>n-2</sub>) = 1. This occurs during the first two symbol periods after transmission of the last nibble of a data stream.

The symbols A<sub>n</sub> and B<sub>n</sub> are obtained from the three-bit words Ta<sub>n</sub>[2:0] and Tb<sub>n</sub>[2:0] whose definitions in the data and the idle modes are given below.

#### Data mode (tx\_enable<sub>n</sub>=1):

*Definition of “SSD”:*

$$Ta_n[2:0] = [1, 0, 0]$$

$$Tb_n[2:0] = [!tx\_enable_{n-1}, tx\_enable_{n-2}, tx\_enable_{n-2}]$$

A most significant bit Ta<sub>n</sub>[2]=1 or Tb<sub>n</sub>[2]=1 results in the transmission of a symbol that can be interpreted as an ESC symbol.

*Encoding of data nibbles:*

$$Ta_n[2:0] = [0, (Sa_n[1] \oplus TXD_n[3]), (Sa_n[0] \oplus TXD_n[2] \oplus 1)]$$

$$Tb_n[2:0] = [0, (Sb_n[1] \oplus TXD_n[1]), (Sb_n[0] \oplus TXD_n[0] \oplus 1)]$$

where TXD<sub>n</sub>[3:0] denotes the data nibble TXD[3:0] at time n.

*Encoding of error indication:*

If TX\_ER<sub>n</sub>=1 is asserted, where TX\_ER<sub>n</sub> denotes the value of the signal TX\_ER at time n, error indication is signaled by means of the ESC and 0 symbols. The encoding rule is given by

$$Ta_n[2:0] = [(Sa_n[1] \oplus Sa_n[0]), 0, 0]$$

$$Tb_n[2:0] = [(Sb_n[1] \oplus Sb_n[0]), 0, 0]$$

#### Idle mode (tx\_enable<sub>n</sub>=0):

*Definition of “ESD”:*

$$Ta_n[2:0] = [1, 0, 0]$$

$$Tb_n[2:0] = [tx\_enable_{n-1}, !tx\_enable_{n-2}, !tx\_enable_{n-2}]$$

*Encoding in the idle mode:*

$$Ta_n[2:0] = [0, Sa_n[1], Sa_n[0]]$$

$$Tb_n[2:0] = [0, Sb_n[1], Sb_n[0]]$$



The mapping of  $Ta_n[2:0]$  and  $Tb_n[2:0]$  into quinary symbols  $Da_n$  and  $Db_n$  is given in Figure 32–9. This mapping ensures that the symbols representing data are Gray coded. The quinary symbols  $A_n$  and  $B_n$  are obtained from  $Da_n$  and  $Db_n$  by

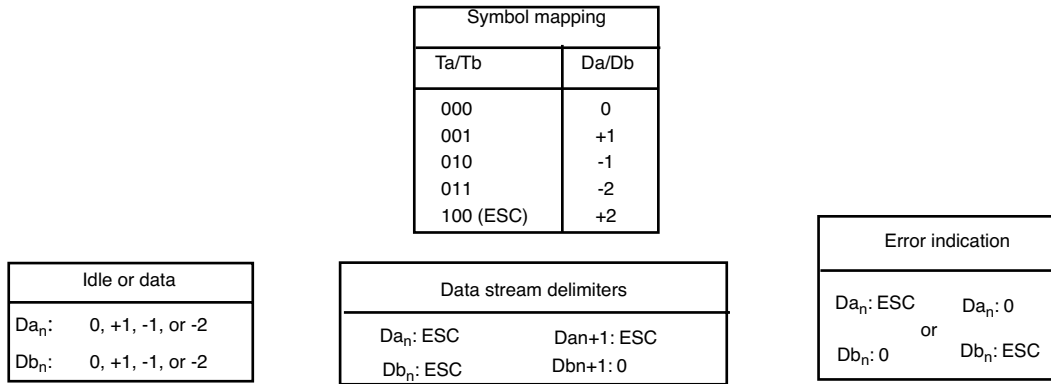


Figure 32–9—Symbol mapping and encoding rule summary

$$A_n = \begin{cases} Da_n & \text{if } Sa_n[2] \oplus tx\_enable_{n-2} = 0 \\ -Da_n & \text{else} \end{cases}$$

$$B_n = \begin{cases} Db_n & \text{if } Sb_n[2] \oplus tx\_enable_{n-2} = 0 \\ -Db_n & \text{else} \end{cases}$$

With the rules defined in this subclause, if in idle mode a transmitted symbol on one wire pair belongs to the set  $\{-2,0,+2\}$ , the symbol on the other wire pair belongs to the set  $\{-1,+1\}$ . Moreover, one of the quinary symbols that are transmitted at time  $2(n-n_0)$  and  $2(n-n_0)+1$  is guaranteed to be either +2 or -2. Both in data and idle modes, the symbol sequences on each wire pair can be modeled as sequences of independent and identically distributed quinary symbols. The symbol constellations and symbol probabilities for these two modes are shown in Figure 32–10. The average symbol energy is the same in data and idle modes.

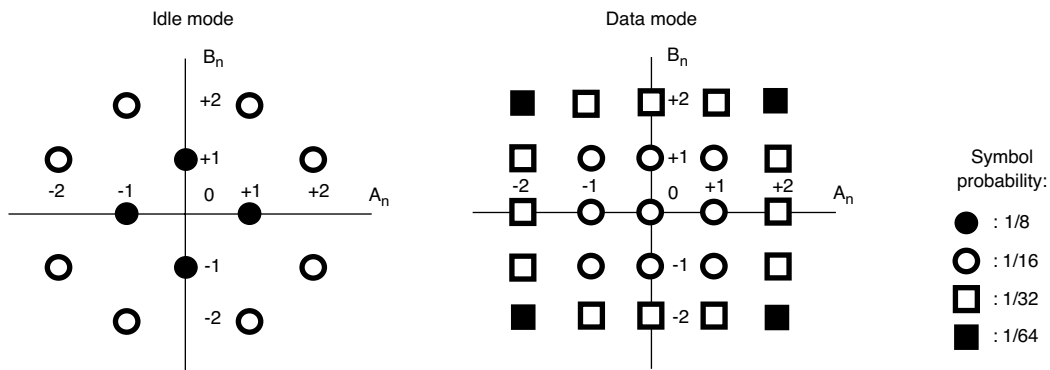


Figure 32–10—Symbol constellations in idle and data modes

### 32.3.1.3 PCS Receive function

The PCS Receive function shall conform to the PCS Receive state diagram in Figure 32–13.

The PCS Receive function accepts pairs of detected quinary symbols provided by the PMA Receive function via the parameter `rx_symb_vector`. To achieve correct operation, PCS Receive uses the knowledge of the encoding rules that are employed in the idle mode. For example, the property that in the idle mode if  $A_n$  belongs to the set  $\{-2,0,+2\}$  then  $B_n$  belongs to the set  $\{-1,+1\}$ , and vice versa, can be used to acquire the correct state for the receiver side-stream descrambler, and to determine which detected quinary symbol was transmitted on wire pair BI\_DA and which on wire pair BI\_DB. Also, correct polarity of the detected quinary symbols can reliably be obtained by ensuring in the idle mode that the encoding rule holds whenever a  $-2$  symbol is received. PCS Receive generates the sequence of vectors of two quinary symbols ( $RA_n, RB_n$ ) and indicates the reliability of receiver operations by setting the parameter `loc_rcvr_status` to OK (logic high) or NOT\_OK (logic low). The sequence ( $RA_n, RB_n$ ) is processed to generate the signals `RXD<3:0>`, `RX_DV`, and `RX_ER`, which are presented to the MII. PCS Receive detects the transmission of a stream of data from the remote station and conveys this information to the PCS Carrier Sense and PCS Collision Presence functions via the parameter `receiving`.

#### 32.3.1.3.1 Receiver descrambler polynomials

For side-stream descrambling, the master PHY shall employ the receiver descrambler generator polynomial  $g'_M(x) = 1 + x^{20} + x^{33}$ , and the slave PHY shall employ the receiver descrambler generator polynomial  $g'_S(x) = 1 + x^{13} + x^{33}$ .

#### 32.3.1.3.2 Decoding of quinary symbols

At the beginning of a stream of data, PCS Receive detects “SSD” and asserts the signal `RX_DV`. Detection of “SSD” is achieved by processing two consecutive vectors ( $RA_{n-1}, RB_{n-1}$ ) and ( $RA_n, RB_n$ ) at each time  $n$ . Upon detection of “SSD,” PCS Receive also assigns the value TRUE to the parameter `receiving` that is provided to the PCS Carrier Sense and Collision Presence functions.

Table 32–1 shows the mapping of symbols  $RA_n$  and  $RB_n$  into two-bit words  $Qa_n[1:0]$  and  $Qb_n[1:0]$  that are descrambled and decoded to generate nibbles of data `RXD[3:0]`.

**Table 32–1 – Inverse quinary symbol mapping**

$RA_n/RB_n$	$Qa_n[1:0]/Qb_n[1:0]$
0	00
+1	01 or 10
-1	01 or 10
$\pm 2$	11

The mapping shown in Table 32–1 corresponds to the inverse of the encoding function employed by PCS Transmit. For example, a symbol  $A_n = +1$  is generated by  $Ta_n[1:0]$  being equal to “01” or “10.” Hence, in the above table the value of  $Qa_n[1:0]$  for  $RA_n = +1$  is specified as being equal to “01 or 10.” Similarly for other entries in the table.

During reception of a stream of data, PCS Receive checks that the symbols  $RA_n$  and  $RB_n$  follow the encoding rule defined in 32.3.1.2 whenever they assume values  $\pm 2$ . PCS Receive processes two consecutive vectors at each time  $n$  to detect “ESD.” Upon detection of “ESD,” PCS Receive de-asserts the signal `RX_DV`, and assigns the value FALSE to the parameter `receiving`. If a violation of the encoding rules is detected, PCS

Receive asserts the signal RX\_ER for at least one symbol period. During reception of an idle sequence, PCS Receive checks that the symbols RA<sub>n</sub> and RB<sub>n</sub> follow the encoding rule defined in 32.3.1.2.

#### 32.3.1.4 PCS Carrier Sense function

The PCS Carrier Sense function shall conform to the PCS Carrier Sense state diagram in Figure 32–14.

The PCS Carrier Sense function controls the MII signal CRS according to the rules presented in this clause. While link\_status = OK, CRS is asserted whenever receiving=TRUE or TX\_EN=1.

#### 32.3.1.5 PCS Collision Presence function

A PCS collision is defined as the simultaneous occurrence of TX\_EN=1 and the assertion of the parameter receiving=TRUE while link\_status=OK. While a PCS collision is detected, the MII signal COL shall be asserted. At other times COL shall remain de-asserted.

### 32.3.2 PCS interfaces

#### 32.3.2.1 PCS–MII interface signals

The signals in Table 32–2 are formally defined in 22.2.2. Jabber detection as specified in 22.2.4.2.12 is not required by this standard.

**Table 32–2—MII interface signals**

Signal name	Meaning	Subclause
TX_CLK	Transmit clock	22.2.2.1
RX_CLK	Receive clock	22.2.2.2
TX_EN	Frames transmit data	22.2.2.3
TXD<3:0>	Transmit data	22.2.2.4
TX_ER	Forces transmission of illegal code	22.2.2.5
RX_DV	Frames receive SFD and DATA	22.2.2.6
RXD<3:0>	Receive data	22.2.2.7
RX_ER	Receive error indication	22.2.2.8
CRS	Non-idle medium indication	22.2.2.9
COL	Collision indication	22.2.2.10
MDC	Management data clock	22.2.2.11
MDIO	Management data	22.2.2.12

#### 32.3.2.2 PCS–management entity signals

The management interface has pervasive connections to all functions. Operation of the management control lines MDC and MDIO, and requirements for managed objects inside the PCS and PMA, are specified in Clauses 22 and 30, respectively.

No spurious signals shall be emitted onto the MDI when the PHY is held in power down mode as defined in 22.2.4.1.5, independently of the value of TX\_EN, or when released from power down mode, or when external power is first applied to the PHY.

### 32.3.3 Frame structure

Frames passed from the PCS to the PMA sublayer shall have the structure shown in Figure 32–11. This figure shows the temporal relationship between the signals  $tx\_enable_n$ , and  $TXD[3:0]$  and the sequences of quinary symbol pairs ( $A_n$ ,  $B_n$ ) in correspondence of transitions from the idle mode to transmission of data and vice versa. Time proceeds from left to right in the figure.

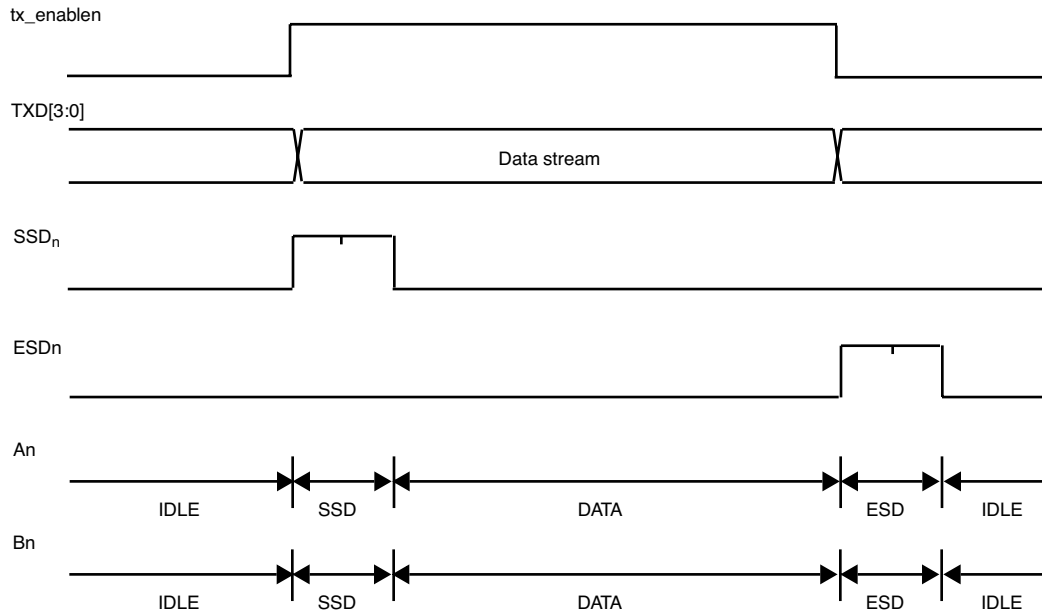


Figure 32–11 – PCS sublayer to PMA sublayer frame structure

### 32.3.4 State variables

#### 32.3.4.1 Variables

COL

The COL signal of the MII as specified in Clause 22.

config

The config parameter set by PHY Control and passed to the PCS via the PHYC\_CONFIG.indicate primitive.

Values: MASTER and SLAVE.

CRS

The CRS signal of the MII as specified in Clause 22.

DATA

A sequence of vectors of two quinary symbols corresponding to valid data, as specified in 32.3.1.2.

ESD

Two consecutive vectors of two quinary symbols corresponding to the End-of-Stream delimiter, as specified in 32.3.1.2.

IDLE

A sequence of vectors of two quinary symbols generated in idle mode, as specified in 32.3.1.2.

**link\_status**

The link\_status parameter set by PMA Link Monitor and passed to the PCS via the PMA\_LINK.indicate primitive.

Values: OK, READY, and FAIL

**loc\_rcvr\_status**

The loc\_rcvr\_status parameter generated by PCS Receive.

Values: OK and NOT\_OK

**pcs\_reset**

The pcs\_reset parameter set by the PCS Reset function.

Values: ON and OFF

**(RA<sub>n</sub>, RB<sub>n</sub>)**

The vector of the two correctly aligned most recently received quinary symbols generated by PCS Receive.

**receiving**

The receiving parameter generated by the PCS Receive function.

Values: TRUE and FALSE

**rxerror\_status**

The rxerror\_status parameter set by the PCS Receive function. Although this variable is set by PCS Receive, it achieves the same function as the variable rxerror\_status of Clause 24 that is set by PMA and communicated through the PMA\_RXERROR.indicate primitive.

Values: ERROR and NO\_ERROR

**RX\_DV**

The RX\_DV signal of the MII as specified in Clause 22.

**RX\_ER**

The RX\_ER signal of the MII as specified in Clause 22.

**rx\_symb\_vector**

A vector of two quinary symbols received by the PMA and passed to the PCS via the PMA\_UNITDATA.indicate primitive.

Value: SYMB\_PAIR

**RXD[3:0]**

The RXD<3:0> signal of the MII as specified in Clause 22.

**SSD**

Two consecutive vectors of two quinary symbols corresponding to the Start-of-Stream delimiter, as specified in 32.3.1.2.

**tx\_enable**

The tx\_enable parameter generated by PCS Transmit as specified in 32.3.1.2.3.

**TX\_ER**

The TX\_ER signal of the MII as specified in Clause 22.

**tx\_mode**

The tx\_mode parameter set by PHY Control and passed to the PCS via the

PHYC\_TXMODE.indicate primitive.  
 Values: SEND\_N and SEND\_I

tx\_symb\_vector

A vector of two quinary symbols generated by the PCS Transmit function and passed to the PMA via the PMA\_UNITDATA.request primitive.  
 Value: SYMB\_PAIR

### 32.3.4.2 Timer

symb\_timer

A continuous free-running timer. The condition symb\_timer\_done becomes true upon timer expiration.

Restart time: Immediately after expiration; timer restart resets the condition symb\_timer\_done.

Duration: 40 ns nominal.

TX\_CLK shall be generated synchronously with symb\_timer (see tolerance required for TX\_CLK in 32.6.1.2.6). In the PCS Transmit state diagram, the message PMA\_UNITDATA.request is issued concurrently with symb\_timer\_done.

### 32.3.4.3 Messages

PMA\_UNITDATA.indicate (rx\_symb\_vector)

A signal sent by PMA Receive indicating that a vector of two quinary symbols is available in rx\_symb\_vector.

PMA\_UNITDATA.request (tx\_symb\_vector)

A signal sent to PMA Transmit indicating that a vector of two quinary symbols is available in tx\_symb\_vector.

## 32.3.5 State diagrams

### 32.3.5.1 PCS Transmit

PCS Transmit sends vectors of two quinary symbols to the PMA via the tx\_symb\_vector parameter. In normal mode of operation, between streams indicated by the parameter tx\_enable, PCS Transmit generates sequences of vectors using the encoding rules defined for the idle mode. Upon assertion of tx\_enable, PCS Transmit passes an “SSD” of two consecutive vectors of two quinary symbols to the PMA replacing the preamble bits of a stream of data during these two symbol periods. Following the “SSD,” each TXD<3:0> nibble is encoded into a vector of two quinary symbols until tx\_enable is de-asserted. If, while tx\_enable is asserted, the TX\_ER signal is also asserted, PCS Transmit passes to the PMA vectors indicating a transmit error. Note that if the signal TX\_ER is asserted while “SSD” is being sent, the transmission of the error condition is delayed until transmission of “SSD” has been completed. Following the de-assertion of tx\_enable, an “ESD” of two consecutive vectors of two quinary symbols is generated, after which the transmission of a sequence indicating the idle mode is resumed.

Collision detection is implemented by noting the occurrence of carrier receptions during transmissions, following the model of 10BASE-T. The PCS shall implement the Transmit process as depicted in Figure 32–12 including compliance with the associated state variables as specified in 32.3.4.

### 32.3.5.2 PCS Receive

PCS Receive accepts vectors of two quinary symbols from the PMA via the rx\_symb\_vector parameter. After correct receiver operation has been achieved, the loc\_rcvr\_status parameter assumes the value OK, and PCS

Receive continuously checks that the received sequence satisfies the encoding rule used in idle mode. As soon as a violation is detected, PCS Receive asserts the parameter receiving and determines whether the violation is due to reception of “SSD” or to a receiver error by examining the last two received vectors ( $RA_{n-1}$ ,  $RB_{n-1}$ ) and ( $RA_n$ ,  $RB_n$ ). In the first case, during the two symbol periods corresponding to “SSD,” PCS Receive replaces “SSD” by preamble bits. Following “SSD,” the signal RX\_DV is asserted and each received vector is decoded into a data nibble RXD<3:0> until “ESD” is detected. De-assertion of RX\_DV and transition to the IDLE state take place upon detection of “ESD”. The signal RX\_ER is asserted if a receiver error occurs before proper stream termination. In the second case, the signal RX\_ER is asserted and the parameter rxerror\_status assumes the value ERROR. De-assertion of RX\_DV and transition to the IDLE state (rxerror\_status=NO\_ERROR) takes place upon detection of a sequence generated in idle mode.

A premature stream termination is caused by the detection of four consecutive vectors satisfying the encoding rule used in idle mode prior to the detection of “ESD”, provided that the first vector is not a valid data vector. Note that RX\_DV remains asserted during the symbol periods corresponding to the first three idle vectors, while RX\_ER=TRUE is signaled on the MII. The signal RX\_ER is also asserted in the LINK FAILED state, which ensures that RX\_ER remains asserted for at least one symbol period. The PCS shall implement the Receive process as depicted in Figure 32–13 including compliance with the associated state variables as specified in 32.3.4.

The parameters receiving and rxerror\_status are communicated to the PCS’s clients by the following primitives:

PCS\_CARRIER.indicate (carrier\_status)

A signal generated by PCS Receive to indicate reception of non-idle quinary symbols. The purpose of this primitive is to give clients indication of activity on the underlying continuous-signaling channel.

PCS\_RXERROR.indicate (rxerror\_status)

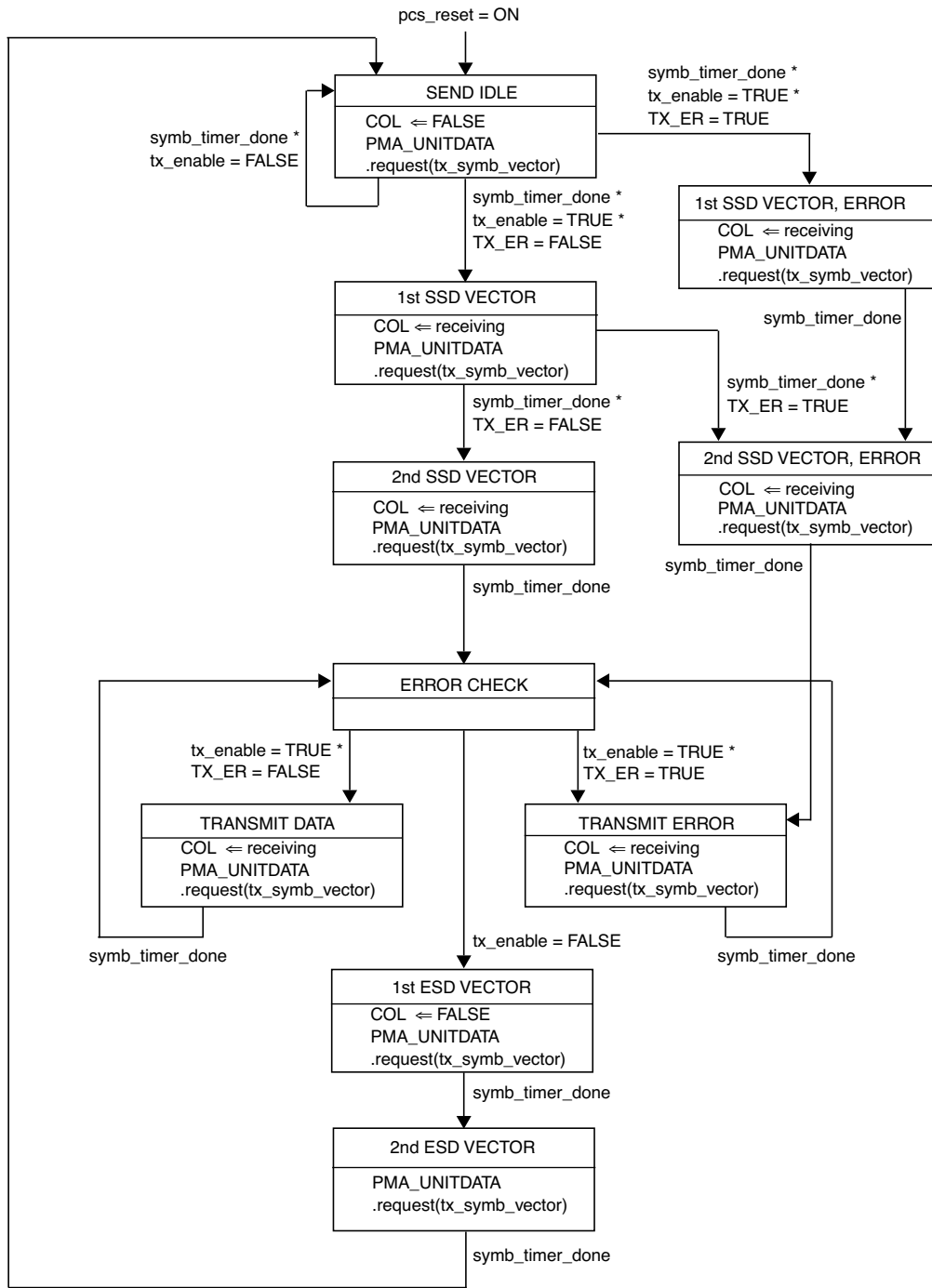
A signal generated by PCS Receive to indicate a reception of non-idle symbols that does not start with “SSD.”

### 32.3.5.3 PCS Carrier Sense

The PCS Carrier Sense process generates the signal CRS on the MII, which the MAC uses via the Reconciliation sublayer for frame receptions and for deferral. The process operates by performing logical operations on TX\_EN and receiving. The PCS shall implement the Carrier Sense process as depicted in Figure 32–14 including compliance with the associated state variables as specified in 32.3.4.

### 32.3.6 PCS electrical specifications

The interface between PCS, PMA and PHY Control is an abstract message-passing interface, having no specified electrical properties. Electrical characteristics of the signals passing between the PCS and MII may be found in Clause 22.



NOTE—The generation of PMA\_UNITDATA.request(tx\_symb\_vector) depends on the parameters config, tx\_mode, and loc\_rcvr\_status as defined in 32.3.1.2 and is not shown explicitly in the state diagram.

Figure 32–12—PCS Transmit state diagram



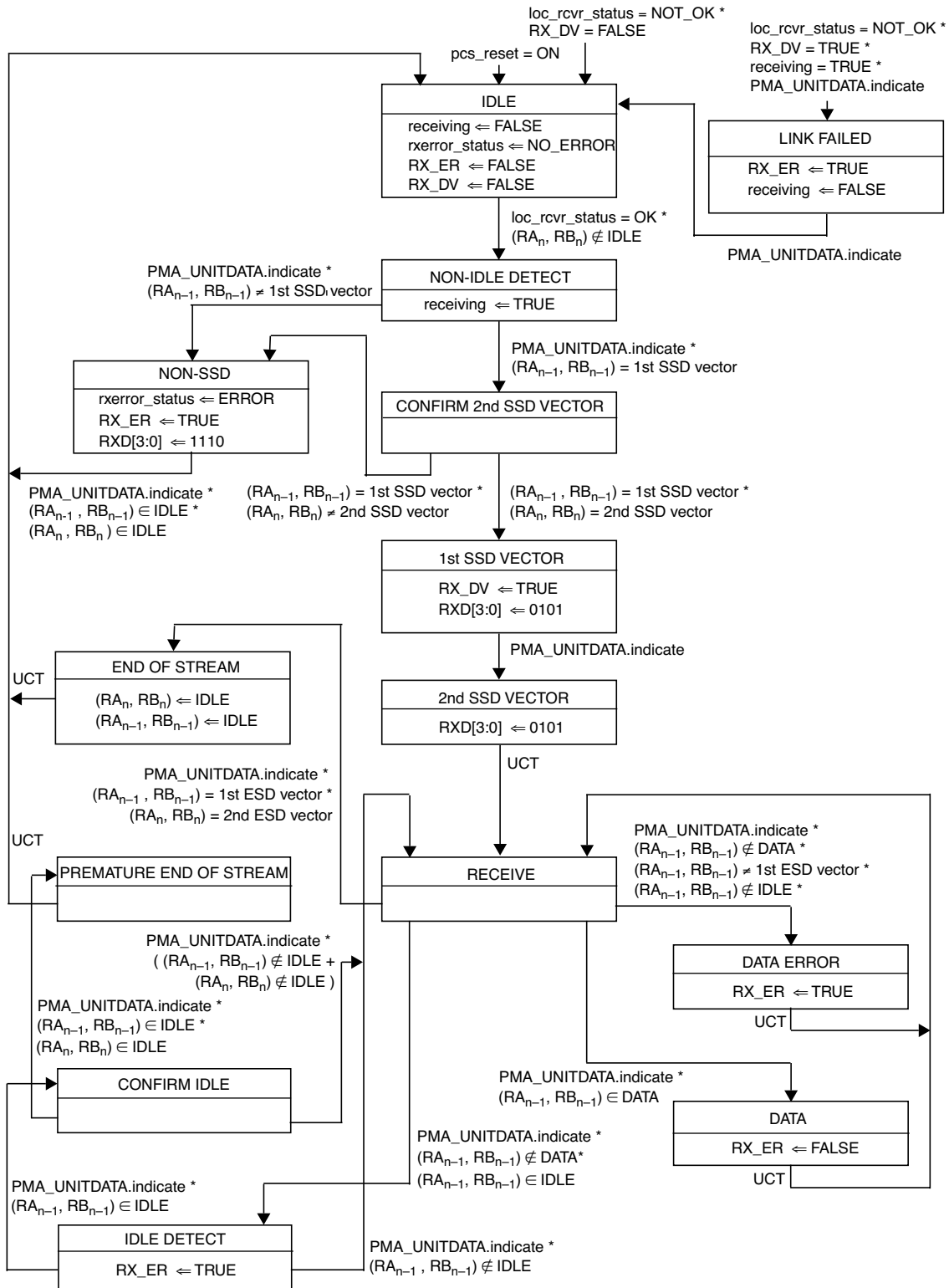


Figure 32–13—PCS Receive state diagram

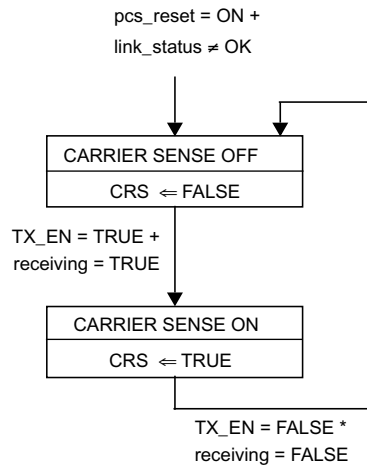


Figure 32–14—PCS Carrier Sense state diagram

### 32.4 PMA functional specifications and service interface

#### 32.4.1 PMA functional specifications

The PMA couples messages from a PMA service interface specified in 32.4.2 to the 100BASE-T2 baseband medium, specified in 32.7. The interface between PMA and the baseband medium is the Medium Dependent Interface (MDI), specified in 32.8.

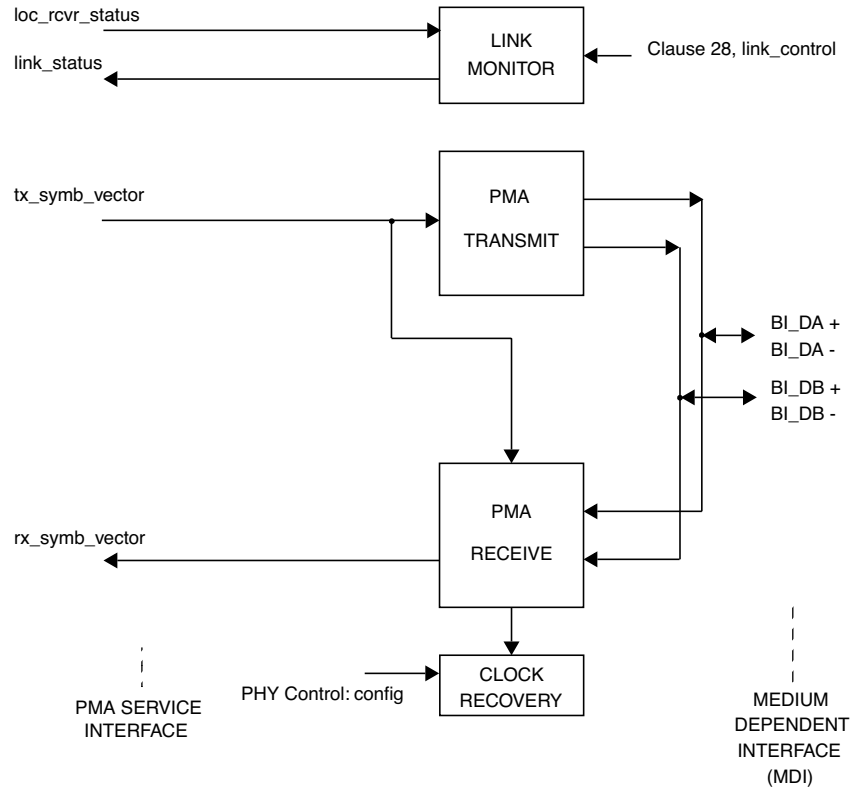


Figure 32–15—PMA reference diagram

### 32.4.1.1 PMA functions

The PMA sublayer comprises one PMA Reset function and four simultaneous and asynchronous operating functions. The PMA operating functions are PMA Transmit, PMA Receive, Link Monitor, and Clock Recovery. All operating functions are started immediately after the successful completion of the PMA Reset function. The Reset function may be shared between PMA and PCS sublayers.

The PMA reference diagram, Figure 32–15, shows how the operating functions relate to the messages of the PMA Service interface, PHY Control Service interface, and the signals of the MDI. Connections from the management interface, comprising the signals MDC and MDIO, to other layers are pervasive, and are not shown in Figure 32–15. The management interface and its functions are specified in Clause 22.

#### 32.4.1.1.1 PMA Reset function

The PMA Reset function shall be executed any time either of two conditions occurs. These two conditions are “power on” and the receipt of a reset request from the management entity. The PMA Reset function initializes all PMA functions and forces Auto-Negotiation to be executed. The PMA Reset function sets `pma_reset = ON` for the duration of its reset function. All state diagrams take the open ended `pma_reset` branch upon execution of the PMA Reset function. The reference diagrams do not explicitly show the PMA Reset function.

#### 32.4.1.1.2 PMA Transmit function

The PMA Transmit function comprises two independent transmitters to generate five-level pulse-amplitude modulated signals on each of the two pairs `BI_DA` and `BI_DB`. PMA Transmit shall continuously transmit onto the MDI pulses modulated by the quinary symbols given by `tx_symb_vector[BI_DA]` and `tx_symb_vector[BI_DB]`, respectively. The two transmitters shall be driven by the same transmit clock. The signals generated by PMA Transmit will follow the mathematical description given in 32.4.1.2.1, and shall comply with the electrical specifications given in 32.6.

#### 32.4.1.1.3 PMA Receive function

The PMA Receive function comprises two independent receivers for quinary pulse-amplitude modulated signals on each of the two pairs `BI_DA` and `BI_DB`. PMA Receive contains the circuits necessary to detect quinary symbol sequences from the signals received at the MDI over receive pairs `BI_DA` and `BI_DB` and present these sequences to the PCS Receive function. The signals received at the MDI are described mathematically in 32.4.1.2.2. The PHY shall translate the signals received on pairs `BI_DA` and `BI_DB` into the `PMA_UNITDATA.indicate` parameter `rx_symb_vector` with a symbol error rate of less than one part in  $10^{10}$ .

To achieve the indicated performance, it is highly recommended that PMA Receive include the functions of signal equalization, suppression of cyclo-stationary interference signals created by alien near-end crosstalk sources, and echo and self near-end crosstalk cancellation. The sequence of quinary transmitted symbols `tx_symb_vector` is needed to perform echo and self near-end crosstalk cancellation.

#### 32.4.1.1.4 Link Monitor function

Link Monitor determines the status of the underlying receive channel. Failure of the underlying receive channel typically causes the PMA's clients to suspend normal operation.

The Auto-Negotiation process notifies Link Monitor whether the device connected to the far end is of type 100BASE-T2. Based on this and other information, Link Monitor sets two important internal variables:

- a) The `pma_type` variable that indicates whether the remote station is of type 100BASE-T2 or not,
- b) The `link_status` variable that is sent across the PMA Service interface.

The Link Monitor function shall comply with the state diagram of Figure 32-16.

Upon power-on, reset, or release from power-down, the Auto-Negotiation algorithm sets `link_control=SCAN_FOR_CARRIER` and sends during this period fast link pulses to signal its presence to a remote station. If the presence of a remote station is sensed through reception of fast link pulses, the Auto-Negotiation algorithm sets `link_control=DISABLE` and exchanges Auto-Negotiation information with the remote station. During this period, `link_status=FAIL` is asserted. If the presence of a remote 100BASE-T2 station is established, the Auto-Negotiation algorithm permits full operation by setting `link_control=ENABLE`. As soon as reliable transmission is achieved, the variable `link_status=OK` is asserted, upon which further PHY operations can take place.

#### 32.4.1.1.5 Clock Recovery function

The Clock Recovery function couples to both receive pairs. It provides a synchronous clock for sampling the signals on the two pairs.

The Clock Recovery function shall provide a clock suitable for synchronous signal sampling on each line so that the symbol-error rate indicated in 32.4.1.1.3 is achieved. The received clock signal must be stable and ready for use when training has been completed (`loc_rcvr_status=OK`).

#### 32.4.1.2 PMA interface messages

The messages between the PMA, PCS, and PHY Control are defined in 32.4.2, PMA Service Interface. Communication through the MDI is summarized below.

##### 32.4.1.2.1 MDI signals transmitted by the PHY

The quinary symbols to be transmitted by the PMA on the two pairs BI\_DA and BI\_DB are denoted by `tx_symb_vector[BI_DA]` and `tx_symb_vector[BI_DB]`, respectively. Five-level Pulse Amplitude Modulation over each pair (PAM5×5) is the modulation scheme employed in 100BASE-T2. It is the function of PMA Transmit to generate on each pair a pulse-amplitude modulated signal of the form

$$s(t) = \sum_k a_k h_1(t - kT)$$

where  $a_k$  represents the quinary symbol from the set  $\{-2, -1, 0, +1, +2\}$  to be transmitted at time  $kT$ , and  $h_1(t)$  denotes the system symbol response at the MDI. This symbol response shall comply with the electrical specifications given in 32.6.

##### 32.4.1.2.2 Signals received at the MDI

Signals received at the MDI can be expressed for each pair as pulse-amplitude modulated signals that are corrupted by noise:

$$r(t) = \sum_k a_k h_2(t - kT) + w(t)$$

In this equation,  $h_2(t)$  denotes the impulse response of the overall channel from the transmit side up to the MDI at the receive side, and  $w(t)$  is a term that includes the contribution of various noise sources. The two signals received on pair BI\_DA and BI\_DB shall be processed within the PMA Receive function to yield the quinary received symbols `rx_symb_vector[BI_DA]` and `rx_symb_vector[BI_DB]`.

### 32.4.1.3 PMA state diagram

#### 32.4.1.3.1 State diagram variables

link\_control

The link\_control parameter as communicated by the PMA\_LINK.request primitive.

Values: See 32.4.2.4.

link\_status

The link\_status parameter as communicated by the Link Monitor function through the PMA\_LINK.indicate primitive.

Values: See 32.4.2.5.

loc\_rcvr\_status

The loc\_rcvr\_status parameter as communicated by the PMA\_RXSTATUS.request primitive.

Values: See 32.2.2.3.1.

pma\_reset

Allows reset of all PMA functions.

Values: ON and OFF

Set by: PMA Reset

#### 32.4.1.3.2 Timers

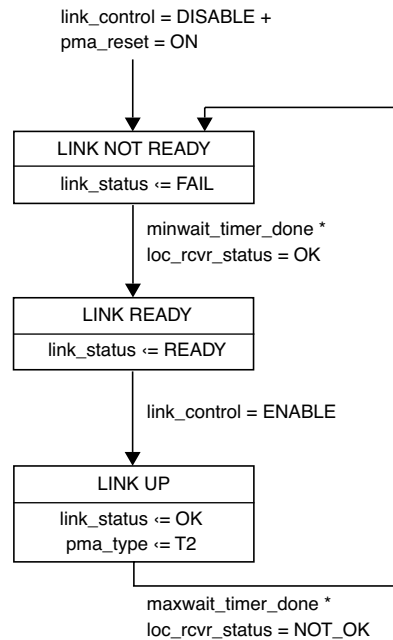
maxwait\_timer

Values: See 32.2.4.

minwait\_timer

Values: See 32.2.4.

#### 32.4.1.3.3 Link Monitor state diagram



NOTE—The variables link\_control and link\_status are designated as link\_control\_[T2] and link\_status\_[T2], respectively, by the Auto-Negotiation Arbitration state diagram (Figure 28–16).

Figure 32-16—Link Monitor state diagram

### 32.4.2 PMA service interface

This subclause specifies the services provided by the PMA. These services are described in an abstract manner and do not imply any particular implementation. The PMA Service Interface supports the exchange of symbol vectors, status indications, and control signals between the PMA, the PCS, and PHY Control. The following primitives are defined:

```
PMA_TYPE.indicate
PMA_UNITDATA.request
PMA_UNITDATA.indicate
PMA_LINK.request
PMA_LINK.indicate
PMA_CARRIER.indicate
PMA_RXERROR.indicate
```

The parameter config is passed from PHY Control to the PMA via the primitive PHYC\_CONFIG.indicate. The definition of this parameter is given for the PHY Control Service interface in 32.2.2.1 and is not repeated here for the PMA Service interface.

#### 32.4.2.1 PMA\_TYPE.indicate

This primitive is generated by the PMA to indicate the nature of the PMA instantiation. Its purpose is to allow clients to support connections to the various types of 100BASE-T PMA entities in a generalized manner.

##### 32.4.2.1.1 Semantics of the service primitive

```
PMA_TYPE.indicate (pma_type)
```

The pma\_type parameter for use with the 100BASE-T2 PMA is T2.

##### 32.4.2.1.2 When generated

The PMA shall continuously generate this primitive to indicate the value of pma\_type.

##### 32.4.2.1.3 Effect of receipt

The client uses the value of pma\_type to define the semantics of the primitives defined at the PMA service interface.

#### 32.4.2.2 PMA\_UNITDATA.request

This primitive defines the transfer of pairs of quinary symbols in the form of the tx\_symb\_vector parameter from the PCS to the PMA. The quinary symbols are obtained in the PCS Transmit function using the encoding rules defined in 32.3.1.2 to represent MII data streams, an idle mode, or other sequences.

##### 32.4.2.2.1 Semantics of the service primitive

```
PMA_UNITDATA.request (tx_symb_vector)
```

During transmission using 100BASE-T2 signaling, the PMA\_UNITDATA.request simultaneously conveys to the PMA via the parameter tx\_symb\_vector the value of the symbols to be sent over each of the two transmit pairs BI\_DA and BI\_DB. The tx\_symb\_vector parameter takes on the form:

```
SYMB_PAIR    A vector of two quinary symbols, one for each of the two transmit pairs BI_DA and
              BI_DB. Each quinary symbol may take on one of the values -2, -1, 0, +1 or +2.
```

The quinary symbols that are elements of `tx_symb_vector` are called, according to the pair on which each will be transmitted, `tx_symb_vector[BI_DA]` and `tx_symb_vector[BI_DB]`.

#### 32.4.2.2.2 When generated

The PCS generates `PMA_UNITDATA.request (SYMB_PAIR)` synchronously with every MII `TX_CLK` cycle.

#### 32.4.2.2.3 Effect of receipt

Upon receipt of this primitive, the PMA transmits on the MDI the signals corresponding to the indicated quinary symbols. The parameter `tx_symb_vector` is also used by the PMA Receive function to process the signals received on pairs `BI_DA` and `BI_DB`.

#### 32.4.2.3 PMA\_UNITDATA.indicate

This primitive defines the transfer of pairs of quinary symbols in the form of the `rx_symb_vector` parameter from the PMA to the PCS.

##### 32.4.2.3.1 Semantics of the service primitive

`PMA_UNITDATA.indicate (rx_symb_vector)`

During reception of PAM5×5 signals using 100BASE-T2 signaling, the `PMA_UNITDATA.indicate` simultaneously conveys to the PCS via the parameter `rx_symb_vector` the values of the symbols detected on each of the two receive pairs `BI_DA` and `BI_DB`. The `rx_symbol_vector` parameter takes on the form:

`SYMB_PAIR` A vector of two quinary symbols, one for each of the two receive pairs `BI_DA` and `BI_DB`. Each quinary symbol may take on one of the values  $-2$ ,  $-1$ ,  $0$ ,  $+1$  or  $+2$ .

The quinary symbols that are elements of `rx_symb_vector` are called, according to the pair upon which each symbol was received, `rx_symbol_vector[BI_DA]` and `rx_symbol_vector[BI_DB]`.

##### 32.4.2.3.2 When generated

The PMA shall generate `PMA_UNITDATA.indicate (SYMB_PAIR)` messages synchronously with signals received at the MDI.

##### 32.4.2.3.3 Effect of receipt

The effect of receipt of this primitive is unspecified.

#### 32.4.2.4 PMA\_LINK.request

This primitive allows the Auto-Negotiation algorithm to enable and disable operation of the PMA.

##### 32.4.2.4.1 Semantics of the service primitive

`PMA_LINK.request (link_control)`

The `link_control` parameter can take on one of three values: `SCAN_FOR_CARRIER`, `DISABLE`, or `ENABLE`.

`SCAN_FOR_CARRIER` Used by the Auto-Negotiation algorithm prior to receiving any fast link pulses. During this mode the PMA reports `link_status=FAIL`. PHY processes are disabled.

DISABLE	Set by the Auto-Negotiation algorithm in the event fast link pulses are detected. <i>PHY processes are disabled</i> . This allows the Auto-Negotiation algorithm to determine how to configure the link.
ENABLE	Used by Auto-Negotiation to turn control over to the PHY for data processing functions.

#### 32.4.2.4.2 When generated

Upon power on, reset, or release from power down, the Auto-Negotiation algorithm issues the message PMA\_LINK.request (SCAN\_FOR\_CARRIER).

#### 32.4.2.4.3 Effect of receipt

While link\_control=SCAN\_FOR\_CARRIER or link\_control=DISABLE, the PMA shall report link\_status=FAIL. While link\_control=ENABLE, PHY Control determines the operations to be performed by the PHY.

#### 32.4.2.5 PMA\_LINK.indicate

This primitive is generated by the PMA to indicate the status of the underlying medium. This primitive informs the PCS, PHY Control and the Auto-Negotiation algorithm about the status of the underlying link.

##### 32.4.2.5.1 Semantics of the service primitive

PMA\_LINK.indicate (link\_status)

The link\_status parameter can take on one of three values: FAIL, READY, or OK.

FAIL	No valid link established.
READY	Training completed after Auto-Negotiation.
OK	The Link Monitor function indicates that a valid 100BASE-T2 link is established. Reliable reception of signals transmitted from the remote PHY is possible.

##### 32.4.2.5.2 When generated

The PMA shall generate this primitive to indicate the value of link\_status in compliance with the state diagrams given in Figure 32-16.

##### 32.4.2.5.3 Effect of receipt

The effect of receipt of this primitive is specified in 32.2 and 32.3.

#### 32.4.2.6 PMA\_CARRIER.indicate

This primitive is identical to PCS\_CARRIER.indicate defined in 32.3.5.2. It is not explicitly shown in the PMA reference diagram.

#### 32.4.2.7 PMA\_RXERROR.indicate

This primitive is identical to PCS\_RXERROR.indicate defined in 32.3.5.2. It is not explicitly shown in the PMA reference diagram.



### 32.4.2.8 PMA\_RXSTATUS.request

This primitive allows the Link Monitor to determine via the parameter `loc_rcvr_status` generated by the PCS Receive function whether reliable receiver operations are established. The parameter `loc_rcvr_status` is also passed from the PCS Receive function to the PHY Control Service interface via the primitive `PHYC_RXSTATUS.request`. The definition of this parameter is given for the PHY Control Service interface in 32.2.2.3 and is not repeated here for the PMA Service interface.

## 32.5 Management functions

100BASE-T2 makes extensive use of the management functions provided by the Media Independent Interface (Clause 22) and the communication and self-configuration functions provided by Auto-Negotiation (Clause 28.)

In addition to the provision of MII Registers 0, 1, 4, 5, 6, and 7, it is required that implementations that support 100BASE-T2 also provide equivalents to MII Registers 8, 9, and 10 (Clause 22). Register 8 is used to provide the Auto-Negotiation Link Partner NEXT Page Register, Register 9 is used to provide the MASTER-SLAVE Control Register, and Register 10 is used to provide the MASTER-SLAVE Status Register. These registers are used to configure PHYs for testing, to manually configure PHYs for MASTER-SLAVE negotiation, to store the contents of next pages during the Auto-Negotiation process, and to store information reporting the results of the master/slave configuration process as described in the next subclause.

### 32.5.1 100BASE-T2 Use of Auto-Negotiation and MII Registers 8, 9, and 10

On power-up, before Auto-Negotiation starts, the Auto-Negotiation Advertisement register shall have the following configuration: The Selector Field (4.4:0) is set to an appropriate code as specified in Annex 28A. The Acknowledge bit (4.14) is set to logic zero. The Technology Ability Field (4:9:5) is set based on the values set in the MII Status Register (Register 1) (1.15:11) or equivalent and (4.11:10) is set based on the values set in the MII Status Register (Register 1) (1.10:9) or equivalent.

When Auto-Negotiation begins, 100BASE-T2 implementations send an Auto-Negotiation base page with bit D15 set to logical one to indicate that a next page follows (see 28.2.1.2.)

The base page is followed by a formatted next page containing the 100BASE-T2 Technology Ability Message Code (7), which indicates that two unformatted next pages containing the 100BASE-T2 Technology Ability fields follow (see Table 28C–1.)

Two unformatted next pages are sent using the 100BASE-T2 Technology Ability fields shown in Table 32–6. Register 8 will be used to store the transmitted information while it is being processed as described below.

Bit U0 of page 1 shall be copied from MII Register 4.10 to indicate 100BASE-T2FD advertised ability.

Bit U1 of page 1 shall be copied from MII Register 4.11 to indicate 100BASE-T2HD advertised ability.

Bit U2 of page 1 shall be copied from MASTER-SLAVE control register 9.10 to indicate that the PHY device is a repeater port or DTE for 100BASE-T2.

Bits U3 and U4 shall be copied from MASTER-SLAVE control register (bits 9.12 and 9.11) for use by the MASTER-SLAVE negotiation process as described below.

Bits U5-U10 of page 1 and U0-U9 of page 2 shall be used to define the seed used for the MASTER-SLAVE negotiation process described as follows.

Using the information described above, the PHY performs a MASTER-SLAVE configuration process as defined in 32.5.4.3. This process is conducted at the entrance to the FLP LINK GOOD CHECK state shown Auto-Negotiation Arbitration State Diagram (Figure 28–16.)

If the local device detects that both the local device and the remote device are of the same type (either repeater or DTE) and that both have generated the same random seed, it sets the Ack2 bit of Register 8 to logical zero and generates and transmits a new random seed for MASTER-SLAVE negotiation.

The MASTER-SLAVE configuration process returns one of the three following outcomes.

**Successful:** Bit 10:15 of the MASTER-SLAVE Status Register is set to logical zero and bit 10.14 is set to logical one. 100BASE-T2 returns control to Auto-Negotiation (at the entrance to the FLP LINK GOOD CHECK state in Figure 28–16) and passes the value of MASTER or SLAVE to PHYC\_CONFIG.indicate (see 32.2.2.)

**Unsuccessful:** link\_status\_T2 is set to FAIL and Auto-Negotiation restarts (see Figure 28–16.)

**Fault detected:** (This happens when both end stations are set for manual configuration and both are set to MASTER or both are set to SLAVE.) Bit 10.15 of the MASTER-SLAVE Status Register is set to logical one to indicate that a manual configuration fault has been detected and bit 10.14 is set to logical one to indicate that MASTER-SLAVE resolution completed with a fault. Because the MASTER-SLAVE relationship was not established, link\_status\_T2 is set to FAIL, causing Auto-Negotiation to restart.

### 32.5.2 Management functions

The management interface specified in Clause 22 provides a simple, two-wire, serial interface to connect a management entity and a managed PHY for the purposes of controlling the PHY and gathering status from the PHY. This interface is referred to as the MII Management Interface. The register definition specifies a basic register set with an extension mechanism.

100BASE-T2 requires the basic register set that consists of two registers referred to as the Control Register (Register 0) and the Status Register (Register 1) and of some PHY-specific registers. The detailed definitions of these registers are given in 22.2.4.

The full set of management registers is listed in Table 22–6 and 100BASE-T2 PHY specific registers are given in Table 32-3.

**Table 32-3—100BASE-T2 Control and Status registers**

Register address	Register name	Basic/Extended
9	MASTER-SLAVE Control register	E
10	MASTER-SLAVE Status register	E

### 32.5.3 PHY specific registers for 100BASE-T2

Some of the extended registers (registers with addresses 2 to 15) are used as PHY specific registers as described in 22.2.4.3. A 100BASE-T2 PHY shall use register addresses 9 and 10 for its control and status functions. The bits in the 100BASE-T2 Control register are used to place the PHY into several possible test modes and to determine the MASTER-SLAVE relationship during Auto-Negotiation. The bits in the

100BASE-T2 Status register are used to report the MASTER-SLAVE relationship determined during Auto-Negotiation, the local and remote receiver status, and provide an idle error counter.

### 32.5.3.1 100BASE-T2 Control register (Register 9)

Register 9 shall provide the following values for 100BASE-T2. The assignment of bits in the register is shown in Table 32–4. The default value for each bit of the register should be chosen so that the initial state of the PHY upon power up or reset is a normal operational state without management intervention.

**Table 32–4—100BASE-T2 Control register (MII management Register 9) bit definition**

Bit(s)	Name	Description	R/W
9.15:14	Transmitter test mode	Default bit values are “00”	R/W
9.13	Receiver test mode	Default bit value is “0”	R/W
9.12	MASTER-SLAVE Manual Configuration Enable	1 = Enable MASTER-SLAVE Manual Configuration value 0 = Disable MASTER-SLAVE Manual Configuration value (default)	R/W
9.11	MASTER-SLAVE Manual Configuration Value	1 = Configure PHY as MASTER during MASTER-SLAVE negotiation, only when 9.12 is set to logical one. 0 = Configure PHY as SLAVE during MASTER-SLAVE negotiation, only when 9.12 is set to logical one.	R/W
9.10	T2_Repeater/DTE bit	1 = Repeater device port 0 = DTE device	R/W
9.9:0	Reserved		

#### 32.5.3.1.1 Transmitter test mode

For a PHY with 100BASE-T2 capability, the PHY shall be placed in transmitter test mode 1 operation (described in 32.6.1.2.1) when bit 9.15 is set to logical zero and bit 9.14 is set to logical one. When bit 9.15 is set to logical one and bit 9.14 is set to logical zero, the PHY shall be placed in transmitter test mode 2 operation as described in 32.6.1.2.1. When bit 9.15 is set to logical one and bit 9.14 is set to logical one, the PHY shall be placed in the transmitter test mode 3 operation as described in 32.6.1.2.1.

The default value for bits 9.15:14 are all zero.

#### 32.5.3.1.2 Receive test mode

The PHY shall be placed in the receiver test mode as described in 32.6.1.3.2 when the bit 9.13 is set to logical one.

The default value for bit 9.13 is zero.

#### 32.5.3.1.3 MASTER-SLAVE Manual Configuration Enable

The MASTER-SLAVE relationship is established during Auto-Negotiation via either automatic MASTER-SLAVE configuration or manual configuration. If bit 9.12 is set to logical zero, then the MASTER-SLAVE configuration negotiation function will determine the PHY configuration. If bit 9.12 is set to logical one, then manual MASTER-SLAVE configuration is enabled, using 9.11 to specify the value. (Usage of this bit is further described in 32.5.3.1.)

The default value of bit 9.12 is zero.

### 32.5.3.1.4 MASTER-SLAVE Manual Configuration Value

MASTER-SLAVE Manual configuration is enabled by setting bit 9.12 to logical one. When manual configuration mode is enabled, setting bit 9.11 to logical one configures the PHY as MASTER, and setting bit 9.11 to logic zero configures the PHY as SLAVE during MASTER-SLAVE negotiation process and shall be used to report the result of the MASTER-SLAVE configuration resolution for that PHY. Detailed description of the use of this bit in MASTER-SLAVE configuration resolution is provided in 32.5.3.1.

The default value of bit 9.11 is zero.

### 32.5.3.1.5 T2\_Repeater/DTE Bit

Bit 9.10 shall be set to logical zero if the PHY is a DTE device and shall be set to a logical one if the PHY is a repeater device port (usage of this bit is described in 32.5.2.)

### 32.5.3.1.6 Reserved bits

Bits 9.9:0 are reserved for future standardization. They shall be written as zero and shall be ignored when read; however, a PHY shall return the value zero in these bits.

### 32.5.3.2 100BASE-T2 Status register (Register 10)

Register 10 shall provide the following values for 100BASE-T2. The assignment of bits in the register is shown in Table 32–5. The default value for each bit of the register should be chosen so that the initial state of the PHY upon power up or reset is a normal operational state without management intervention.

**Table 32–5—100BASE-T2 Status register (MII management Register 10) bit definition**

Bit(s)	Name	Description	R/W <sup>a</sup>
10.15	MASTER-SLAVE manual configuration fault	1 = MASTER-SLAVE manual configuration fault detected 0 = No MASTER-SLAVE manual configuration fault detected	RO/SC
10.14	MASTER-SLAVE configuration resolution complete	1 = MASTER-SLAVE configuration resolution has completed 0 = MASTER-SLAVE configuration resolution has not completed	RO
10.13	Local Receiver Status	1 = Local Receiver OK 0 = Local Receiver not OK	RO
10.12	Remote Receiver Status	1 = Remote Receiver OK 0 = Remote Receiver not OK	RO
10.11:8	Reserved		
10.7:0	Idle Error Count	Idle Error count	RO/SC

<sup>a</sup>R/W = Read/Write, SC = Self Clearing, RO = Read Only

#### 32.5.3.2.1 MASTER-SLAVE Manual Configuration Fault

When read as a logical one, bit 10.15 indicates that a MASTER-SLAVE Manual Configuration Fault condition has been detected. The type of fault as well as the criteria and method of fault detection is PHY specific. The MASTER-SLAVE Manual Configuration Fault bit shall be implemented with a latching function, such that the occurrence of a MASTER-SLAVE Manual Configuration Fault will cause the MASTER-SLAVE Manual Configuration Fault bit to be set and remain set until it is cleared. The MASTER-SLAVE Manual Configuration Fault bit shall be cleared each time Register 10 is read via the management interface and shall also be cleared by a 100BASE-T2 PMA reset.

For 100BASE-T2, this fault condition will occur when both PHYs are forced to be MASTER or SLAVE at the same time using bits 9.12 and 9.11. Bit 10.15 should be set via the MASTER-SLAVE Configuration Resolution function described in 32.5.4.

#### **32.5.3.2.2 MASTER-SLAVE Configuration Resolution Complete**

When read as a logical one, bit 10.14 indicates that the MASTER-SLAVE Resolution process has been completed and that the contents of Registers 9 and 10 related to MASTER-SLAVE are valid. When read as a logic zero, bit 10.14 indicates that the MASTER-SLAVE Configuration Resolution process has not been completed and that the contents of Registers 9 and 10 which are related to MASTER-SLAVE resolution are invalid. Bit 10.14 should be set via the MASTER-SLAVE Configuration Resolution function described in 32.5.4.

#### **32.5.3.2.3 Local Receiver Status**

Bit 10.13 indicates the status of the local receiver. Local receiver status is defined by the value of the `loc_rcvr_status` variable described in 32.2.3.

#### **32.5.3.2.4 Remote Receiver Status**

Bit 10.12 indicates the status of the remote receiver. Remote receiver status is defined by the value of the `rem_rcvr_status` variable described in 32.2.3.

#### **32.5.3.2.5 Reserved bits**

Bit 10.11:8 are reserved for future standardization. They shall be written as zero and shall be ignored when read; however, a PHY shall return the value zero in these bits.

#### **32.5.3.2.6 Idle Error count**

Bits 10.7:0 indicate the Idle Error count, where 10.7 is the most significant bit. During normal operation these bits contain a cumulative count of the errors detected when the receiver is receiving idles and the PHY Control parameter `tx_mode` is equal to `SEND_N` (indicating that both local and remote receiver status have been detected to be OK). When the PHY has receiver test mode (bit 9.13) enabled, these bits contain a cumulative count of the errors detected at all times when the local receiver status is OK. These bits are reset to all zeroes when the error count is read by the management function or upon execution of the PCS Reset function and they are held at all ones in case of overflow (see 30.5.1.1.11).

### **32.5.4 Changes and additions to Auto-Negotiation (Clause 28)**

#### **32.5.4.1 Change to 28.2.4.1.3 (Auto-Negotiation Advertisement register)**

For implementations which support 100BASE-T2, the Technology Ability Field (4:9:5) is set based on the values set in the MII Status Register (Register 1) (1.15:11) or equivalent and (4.11:10) is set based on the values set in the MII Status Register (Register 1) (1.10:9) or equivalent. Use of Register 4 is defined in 28.2.4.1.3.

#### **32.5.4.2 Use of Auto-Negotiation Next Page codes for 100BASE-T2 PHYs**

For a PHY capable of 100BASE-T2 transmission, during Auto-Negotiation the Base Page will be followed by a Next Page with a message code containing the 100BASE-T2 Technology Ability Message Code (7) as shown in Table 28C–1. This Message Next Page indicates that two Unformatted Message Next Pages will follow which contain the 100BASE-T2 Technology Ability Fields as described in Table 32–6.

**Table 32–6 – Bit assignments for Unformatted Next Pages containing  
100BASE-T2 Technology Ability Field**

Bit	Technology	MII register bit/source
PAGE 1		
U0	100BASE-T2 Half Duplex (1 = Half Duplex and 0 = no Half Duplex)	MII Register 4.10
U1	100BASE-T2 Full Duplex (1=Full Duplex and 0 = no Full Duplex)	MII Register 4.11
U2	100BASE-T2 Repeater/DTE bit (1 = Repeater and 0 = DTE)	MII Register 9.10 (MASTER-SLAVE Control register)
U3	100BASE-T2 MASTER-SLAVE Manual Configuration Enable (1 = Manual Configuration Enable); intended to be used for manual selection in a particular MASTER-SLAVE mode. To be used in conjunction with bit U4	MII Register 9.12 (MASTER-SLAVE Control register)
U4	100BASE-T2 MASTER-SLAVE Manual Configuration value 1 = MASTER and 0 = SLAVE. This bit is ignored if U3 = 0.	MII Register 9.11 (MASTER-SLAVE Control register)
U5	100BASE-T2 MASTER-SLAVE Seed Bit 0 (SB0) (LSB)	MASTER-SLAVE seed value (15.0)
U6	100BASE-T2 MASTER-SLAVE Seed Bit 1 (SB1)	
U7	100BASE-T2 MASTER-SLAVE Seed Bit 2 (SB2)	
U8	100BASE-T2 MASTER-SLAVE Seed Bit 3 (SB3)	
U9	100BASE-T2 MASTER-SLAVE Seed Bit 4 (SB4)	
U10	100BASE-T2 MASTER-SLAVE Seed Bit 5 (SB5)	
PAGE 2		
U0	100BASE-T2 MASTER-SLAVE Seed Bit 6 (SB6)	
U1	100BASE-T2 MASTER-SLAVE Seed Bit 7 (SB7)	
U2	100BASE-T2 MASTER-SLAVE Seed Bit 8 (SB8)	
U3	100BASE-T2 MASTER-SLAVE Seed Bit 9 (SB9)	
U4	100BASE-T2 MASTER-SLAVE Seed Bit 10 (SB10)	
U5	100BASE-T2 MASTER-SLAVE Seed Bit 11 (SB11)	
U6	100BASE-T2 MASTER-SLAVE Seed Bit 12 (SB12)	
U7	100BASE-T2 MASTER-SLAVE Seed Bit 13 (SB13)	
U8	100BASE-T2 MASTER-SLAVE Seed Bit 14 (SB14)	
U9	100BASE-T2 MASTER-SLAVE Seed Bit 15 (SB15)	
U10	unused	

### 32.5.4.3 MASTER-SLAVE Configuration Resolution

Since both PHYs that share a link segment are capable of being MASTER or SLAVE, a prioritization scheme exists to ensure that the correct mode is chosen. The MASTER-SLAVE relationship shall be determined during Auto-Negotiation using Table 32–7 with the 100BASE-T2 Technology Ability Next Page bit values specified in Table 32–7 and information received from the link partner.

**Table 32–7 – 100BASE-T2 MASTER-SLAVE Configuration Resolution table**

Local Device Type	Remote Link Partner Type	Resolution Function result	
		Local Device	Remote Link Partner
DTE (U2=0 & U3=0) or Manual SLAVE (U3=1 & U4=0)	Repeater (U2=1 & U3=0) or Manual MASTER (U3=1 & U4=1)	SLAVE	MASTER
Repeater (U2=1 & U3=0)	Manual MASTER (U3=1 & U4=1)		
Manual SLAVE (U3=1 & U4=0)	DTE (U2=0 & U3=0)		
Repeater (U2=1 & U3=0) or Manual MASTER (U3=1 & U4=1)	DTE (U2=0 & U3=0) or Manual SLAVE (U3=1 & U4=0)	MASTER	SLAVE
DTE (U2=0 & U3=0)	Manual Slave (U3=1 & U4=0)		
Manual Master (U3=1 & U4=1)	Repeater (U2=1 & U3=0)		
Repeater (U2=1 & U3=0)	Repeater (U2=1 & U3=0)	PHY with higher seed value is the MASTER. If the seeds are equal, the MASTER-SLAVE resolution is unsuccessful, set link_status_T2=FAIL, causing Auto-Negotiation to restart.	
DTE (U2=0 & U3=0)	DTE (U2=0 & U3=0)		
Manual SLAVE (U3=1 & U4=0)	Manual SLAVE (U3=1 & U4=0)	Fault detected Set link_status_T2=FAIL, forcing Auto-Negotiation to restart.	
Manual MASTER (U3=1 & U4=1)	Manual MASTER (U3=1 & U4=1)		

The rationale for this hierarchy is straightforward. A 100BASE-T2 repeater has higher priority than a DTE to become the MASTER. In the case where both devices are of the same type, e.g., both devices are Repeaters, the device with the higher MASTER-SLAVE seed bits (SB0..SB15), where SB15 is the MSB, becomes the MASTER and the device with the lower seed value becomes the SLAVE. In case both devices have the same seed value, both should assert link\_status\_T2=FAIL (as defined in 28.3.1) and restart Auto-Negotiation. Successful completion of the MASTER-SLAVE resolution shall be treated as MASTER-SLAVE configuration resolution complete and the 100BASE-T2 Status Register bit 10.14 shall be set to logical one.

The method of generating a random seed is left to the implementor. The generated random seeds should belong to a sequence of independent, identically distributed integer numbers with a uniform distribution in the range of 0 to 65535. The algorithm used to generate the integer should be designed to minimize the correlation between the number generated by any two devices at any given time. A seed counter shall be provided to track the number of seed attempts. The seed counter shall be set to zero at start-up and shall be incremented each time a seed is generated. A MASTER-SLAVE resolution fault shall be declared if resolution is not reached after the generation of seven seeds.

The MASTER-SLAVE Manual Configuration Enable bit (control register bit 9.12) is used to manually set a device to become the MASTER or the SLAVE. In case both devices are manually set to become the MASTER or the SLAVE, this condition shall be flagged as a MASTER-SLAVE Manual Configuration fault condition, thus the MASTER-SLAVE Manual Configuration fault bit (status register bit 10.15) shall be set to logical one. The MASTER-SLAVE Manual Configuration fault condition shall be treated as MASTER-SLAVE configuration resolution complete and status register bit 10.14 shall be set to logical one. In this case, link\_status\_T2 will be set to FAIL, because the MASTER-SLAVE relationship was not resolved. This will force Auto-Negotiation to restart after the link\_fail\_inhibit\_timer has expired. Determination of MASTER-SLAVE values occur on the entrance to the FLP LINK GOOD CHECK state (Figure 28–16) when the highest common denominator (HCD) technology is 100BASE-T2. The resulting MASTER-SLAVE value is used by the 100BASE-T2 PHY control (32.2.2.1).

## 32.6 PMA electrical specifications

This clause defines the electrical characteristics of the PHY at the MDI.

The ground reference point for all common-mode tests is the MII ground circuit. Implementations without an MII use the chassis ground. The values of all components in test circuits shall be accurate to within  $\pm 1\%$  unless otherwise stated.

### 32.6.1 PMA-to-MDI interface characteristics

#### 32.6.1.1 Isolation requirement

The PHY shall provide electrical isolation between the DTE or repeater circuits, including frame ground and all MDI leads. This electrical separation shall withstand at least one of the following electrical strength tests:

- a) 1500 V rms at 50–60 Hz for 60 s, applied as specified in Section 5.3.2 of IEC 60950
- b) 2250 Vdc for 60 s, applied as specified in Section 5.3.2 of IEC 60950
- c) A sequence of ten 2400 V impulses of alternating polarity, applied at intervals of not less than 1 s. The shape of the impulses shall be 1.2/50  $\mu$ s (1.2  $\mu$ s virtual front time, 50  $\mu$ s virtual time or half value), as defined in IEC 60060.

There shall be no insulation breakdown, as defined in Section 5.3.2 of IEC 60950, during the test. The resistance after the test shall be at least 2 M $\Omega$ , measured at 500 Vdc.

#### 32.6.1.2 Transmitter electrical specifications

The PMA shall provide the Transmit function specified in 32.4.1.1.2 in accordance with the electrical specifications of this clause.

Where a load is not specified, the transmitter shall meet requirements of this clause with a 100  $\Omega$  resistive differential load connected to each transmitter output.

The tolerance on the poles of the test filters used in this clause shall be  $\pm 1\%$ .



### 32.6.1.2.1 Transmitter test modes

Since the 100BASE-T2 PCS employs scrambling, synchronization of data at the MII to the scrambled state is virtually impossible. Therefore a special transmit test mode shall be required to allow for testing of the transmitter waveform. Additionally, a test mode for measuring transmitter output jitter is also required.

For a PHY with a MII interface, these modes shall be enabled by setting bits 9.14 and 9.15 (MASTER-SLAVE Control Register) of the MII Management register set as shown in Table 32–8. These test modes shall only change the data symbols provided to the transmitter circuitry and may not alter the electrical characteristics of the transmitter. A PHY without an MII shall provide a means to provide these functions. The vendor shall provide a means to enable these modes for conformance testing.

**Table 32–8—MII management register set**

Bit 9.15	Bit 9.14	Mode
0	0	Normal operation
0	1	TX Test mode 1—waveform test
1	0	TX Test mode 2—jitter test
1	1	TX Test mode 3—

When transmit test mode 1 is enabled, the PHY shall transmit the following sequence of data symbols ( $A_n$  and  $B_n$  of 32.3.1.2.3) continually from both transmitters:

{+2 followed by 127 0 symbols}, {-2 followed by 127 0 symbols}, {+1 followed by 127 0 symbols}, {-1 followed by 127 0 symbols}, {16 +2 symbols, 16 -2 symbols followed by 224 0 symbols}

This sequence is repeated continually without breaks between the repetitions when the transmit test mode is enabled. A typical transmitter output is shown below in Figure 32–17. The transmitter shall time the transmitted symbols from a 25.000 MHz  $\pm$  0.01% clock.

When transmit test mode 2 is enabled, the PHY shall transmit the data symbol sequence {+2,-2} repeatedly on both channels. The transmitter shall time the transmitted symbols from a 25.000 MHz  $\pm$  0.01% clock.

When transmit test mode 3 is enabled, the PHY shall transmit idle data compliant with the idle signaling specified in 32.3 with `loc_rcvr_status=OK`.

### 32.6.1.2.2 Peak differential output voltage and level distortion

When in transmit test mode 1 and observing the differential signal output at the MDI terminated in 100  $\Omega$ , preprocessed by the high pass filter defined below, for each pair, with no intervening cable, the absolute value of the peak of the waveform at points A and B as defined in Figure 32–17 shall fall within 1.71–1.91V (1.81 V  $\pm$  0.5 dB).

The absolute value of the peak of the waveforms at points A and B shall differ by less than 1%.

The absolute value of the peak of the waveform at points C and D as defined in Figure 32–17 shall differ from 0.5 times the average of the absolute values of the peaks of the waveform at points A and B by less than 2%.

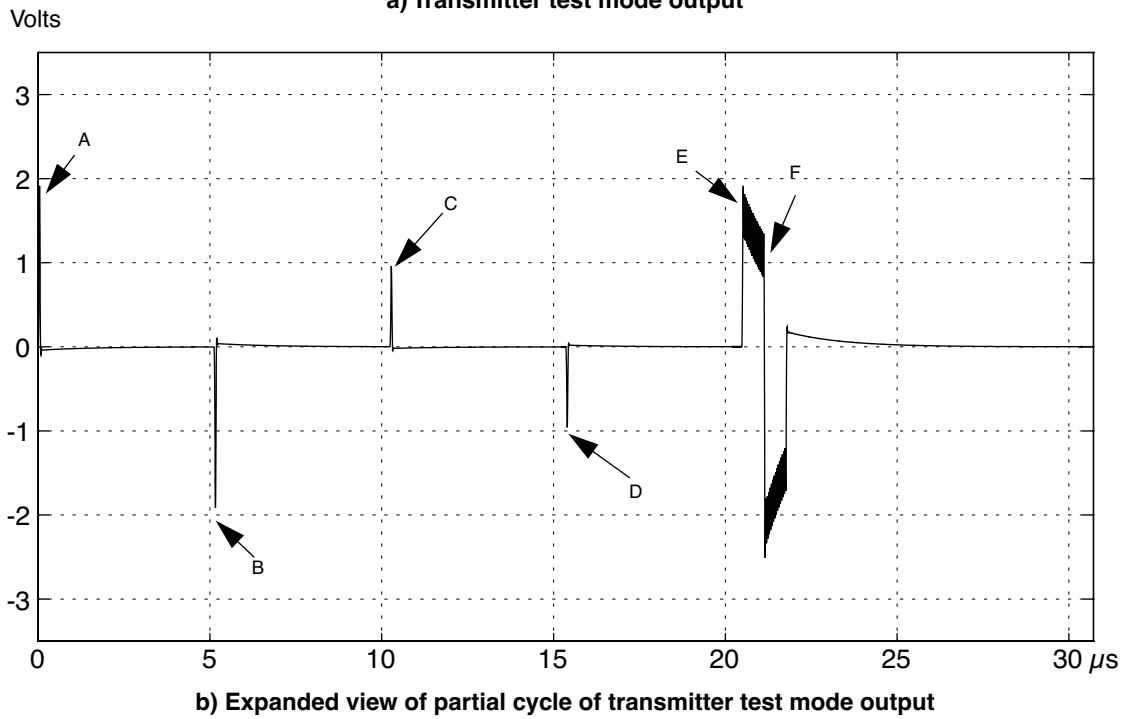
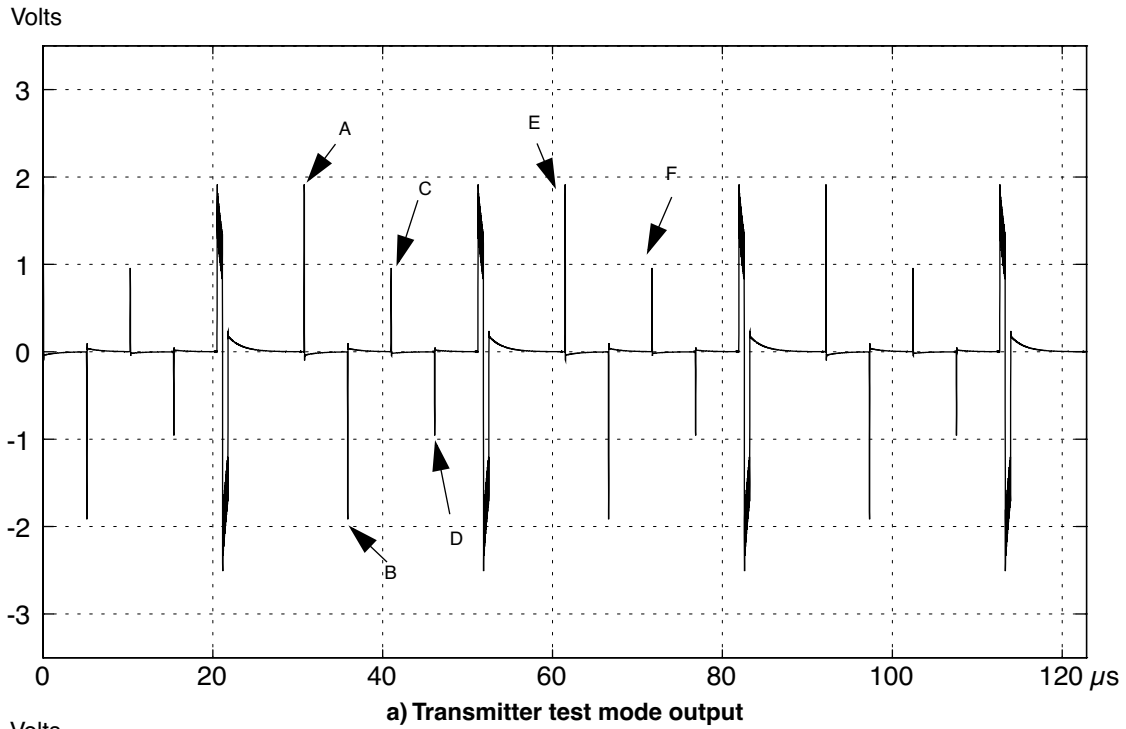


Figure 32-17—Example transmitter test mode transmitter

The preprocessing filter shall have the following transfer function<sup>14</sup>:

$$H_{\text{preprocess}}(f) = \frac{jf}{jf + 200 \times 10^3} \quad f \text{ in Hz}$$

### 32.6.1.2.3 Maximum output droop

When in transmit test mode 1 and observing the differential signal output at the MDI, for either pair, with no intervening cable, the peak value of the waveform at point F as defined in Figure 32–17 shall be greater than 70.5% of the peak value of the waveform at point E. A preprocessing filter is not used for this measurement.

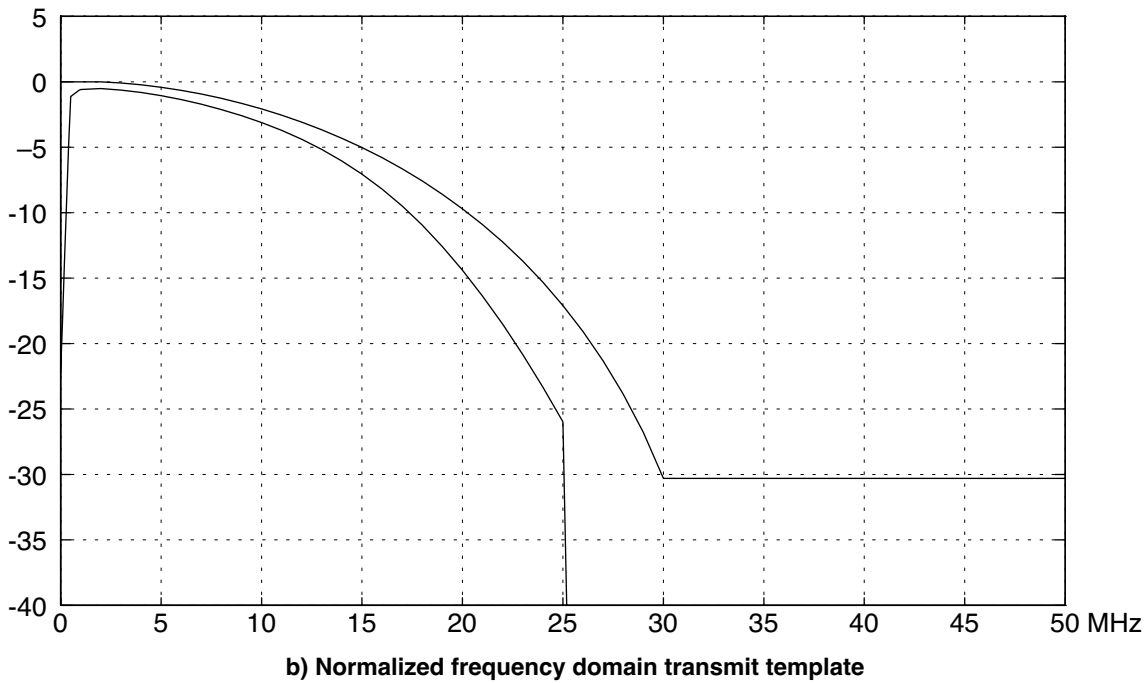
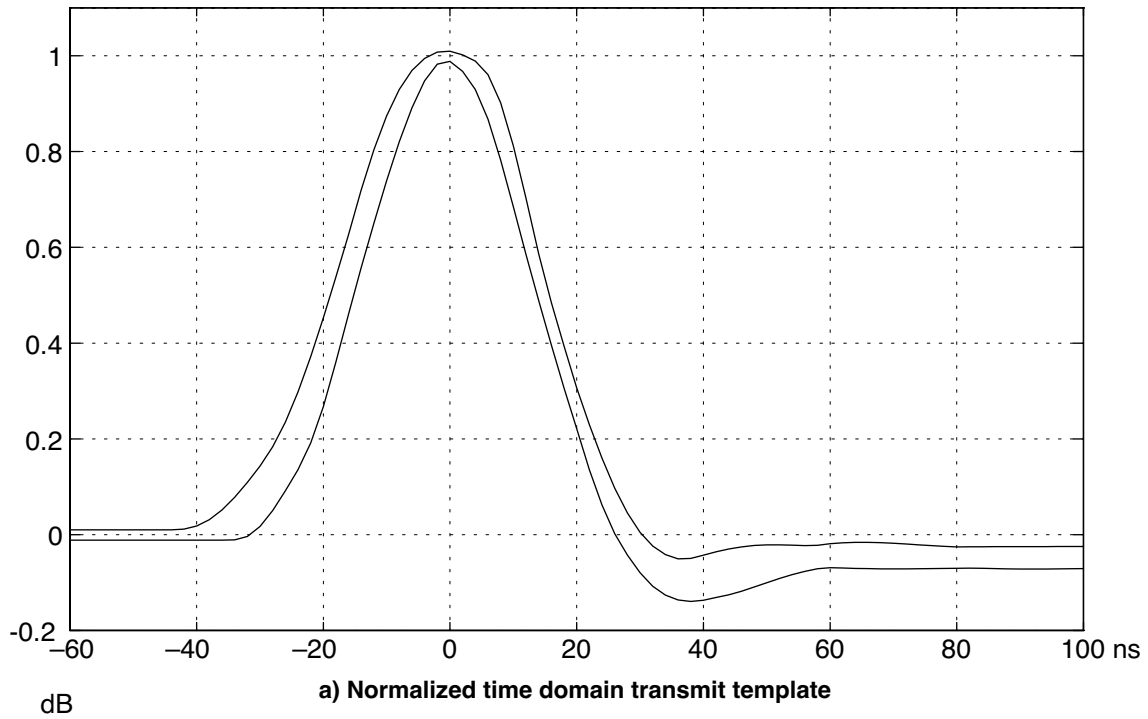
### 32.6.1.2.4 Differential output templates

The transmitter differential output voltage shall be measured at the output of the high pass preprocessing filter defined in 32.6.1.2.2, with no intervening cables. The voltage waveforms A, B, C and D defined in Figure 32–17, after normalization by their respective peak values, shall lie within the time domain template defined in Figure 32–18 and the piecewise linear interpolation between the points in Table 32–9. The waveforms may be shifted in time as appropriate to fit within the template.

Additionally, the magnitude in dB of the Fourier transform of the preprocessed waveforms A, B, C and D shall lie within the frequency domain template defined in Figure 32–18 and the piecewise linear interpolation between the points in Table 32–9. The time span of the waveforms so processed shall be –80 ns to +2000 ns with the 0 ns point of the waveform aligned as for the time domain mask shown in Figure 32–18 and the magnitude of the Fourier transform should be normalized so that the maximum value is at 0 dB.<sup>15</sup>

<sup>14</sup>“j” denotes the positive square root of –1.

<sup>15</sup>A sampling rate of 100 MHz is adequate to generate the frequency domain mask.



NOTE—The frequency domain transmit template is not intended to address electromagnetic radiation limits.

**Figure 32–18—Normalized transmit templates as measured at MDI through preprocessing filter**

**Table 32–9—Normalized time domain voltage template**

Time, ns	Normalized transmit time domain template, upper limit	Normalized transmit time domain template, lower limit	Time, ns	Normalized transmit time domain template, upper limit	Normalized transmit time domain template, lower limit
–60	0.010	–0.011	22	0.230	0.136
–58	0.010	–0.011	24	0.160	0.062
–56	0.010	–0.011	26	0.097	0.002
–54	0.010	–0.011	28	0.045	–0.042
–52	0.010	–0.011	30	0.005	–0.079
–50	0.010	–0.011	32	–0.024	–0.108
–48	0.010	–0.011	34	–0.042	–0.126
–46	0.010	–0.011	36	–0.051	–0.136
–44	0.010	–0.011	38	–0.050	–0.139
–42	0.012	–0.011	40	–0.043	–0.137
–40	0.018	–0.011	42	–0.036	–0.131
–38	0.031	–0.011	44	–0.030	–0.126
–36	0.052	–0.011	46	–0.025	–0.118
–34	0.078	–0.011	48	–0.023	–0.109
–32	0.109	–0.004	50	–0.021	–0.100
–30	0.143	0.017	52	–0.021	–0.091
–28	0.184	0.050	54	–0.022	–0.084
–26	0.235	0.092	56	–0.023	–0.077
–24	0.298	0.136	58	–0.022	–0.071
–22	0.372	0.192	60	–0.019	–0.069
–20	0.453	0.268	62	–0.017	–0.070
–18	0.538	0.360	64	–0.016	–0.070
–16	0.627	0.461	66	–0.016	–0.071
–14	0.720	0.558	68	–0.017	–0.071
–12	0.804	0.650	70	–0.018	–0.071
–10	0.874	0.739	72	–0.020	–0.071
–8	0.930	0.820	74	–0.021	–0.071
–6	0.969	0.891	76	–0.023	–0.071
–4	0.995	0.948	78	–0.024	–0.070
–2	1.008	0.982	80	–0.026	–0.070
0	1.009	0.988	82	–0.025	–0.070

**Table 32–9—Normalized time domain voltage template (continued)**

Time, ns	Normalized transmit time domain template, upper limit	Normalized transmit time domain template, lower limit	Time, ns	Normalized transmit time domain template, upper limit	Normalized transmit time domain template, lower limit
2	1.002	0.967	84	-0.025	-0.070
4	0.989	0.930	86	-0.025	-0.071
6	0.961	0.868	88	-0.025	-0.071
8	0.902	0.782	90	-0.025	-0.072
10	0.812	0.685	92	-0.025	-0.072
12	0.703	0.585	94	-0.025	-0.072
14	0.587	0.489	96	-0.025	-0.071
16	0.485	0.396	98	-0.025	-0.071
18	0.394	0.306	100	-0.025	-0.071
20	0.307	0.221			

**Table 32–10—Normalized frequency domain amplitude spectrum template**

Frequency, MHz	Normalized transmit frequency domain template, upper limit, dB	Normalized transmit frequency domain template, lower limit, dB	Frequency, MHz	Normalized transmit frequency domain template, upper limit, dB	Normalized transmit frequency domain template, lower limit, dB
0.1	0.00	-17.88	22	-12.24	-18.52
0.2	0.00	-13.49	23	-13.70	-20.84
0.3	0.00	-9.09	24	-15.31	-23.32
0.4	0.00	-4.70	25	-17.09	-25.97
0.5	0.00	-1.13	26	-19.08	
0.6	0.00	-1.01	27	-21.31	
0.7	0.00	-0.90	28	-23.84	
0.8	0.00	-0.78	29	-26.78	
0.9	0.00	-0.66	30	-30.29	
1	0.00	-0.59	31	-30.29	
2	-0.00	-0.52	32	-30.29	
3	-0.08	-0.63	33	-30.29	
4	-0.23	-0.82	34	-30.29	

**Table 32–10—Normalized frequency domain amplitude spectrum template (continued)**

Frequency, MHz	Normalized transmit frequency domain template, upper limit, dB	Normalized transmit frequency domain template, lower limit, dB	Frequency, MHz	Normalized transmit frequency domain template, upper limit, dB	Normalized transmit frequency domain template, lower limit, dB
5	-0.42	-1.06	35	-30.29	
6	-0.65	-1.36	36	-30.29	
7	-0.93	-1.71	37	-30.29	
8	-1.26	-2.12	38	-30.29	
9	-1.64	-2.59	39	-30.29	
10	-2.07	-3.12	40	-30.29	
11	-2.55	-3.72	41	-30.29	
12	-3.08	-4.40	42	-30.29	
13	-3.67	-5.17	43	-30.29	
14	-4.31	-6.05	44	-30.29	
15	-5.03	-7.06	45	-30.29	
16	-5.80	-8.20	46	-30.29	
17	-6.65	-9.50	47	-30.29	
18	-7.58	-10.96	48	-30.29	
19	-8.59	-12.59	49	-30.29	
20	-9.70	-14.39	50	-30.29	
21	-10.91	-16.37			

**32.6.1.2.5 Transmitter timing jitter**

When in transmit test mode 2, the peak-to-peak jitter of the zero crossings of the differential signal output at the MDI shall be less than 0.5 ns.

**32.6.1.2.6 Transmit clock frequency**

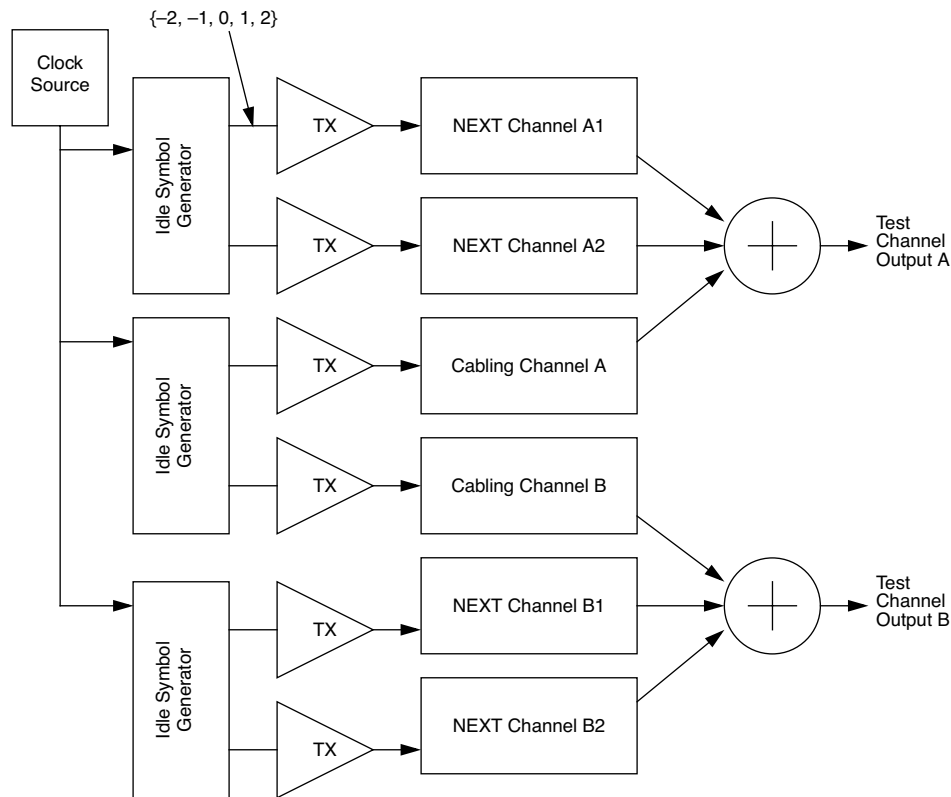
The quinary symbol transmission rate on each pair shall be 25.000 MHz  $\pm$  0.01%.

**32.6.1.3 Receiver electrical specifications**

The PMA shall provide the Receive function specified in 32.4.1.1.3 in accordance with the electrical specifications of this clause. The patch cabling and interconnecting hardware used in test configurations shall meet or exceed ISO/IEC 11801, Category 3 specifications.

### 32.6.1.3.1 Test channel

To perform the Receiver Alien NEXT Tolerance test and Receiver timing jitter test described in this clause, a test channel including transmitter, cabling and NEXT models is required. This test channel is shown conceptually in Figure 32–19.



**Figure 32–19—Conceptual diagram of test channel**

The combined responses of the TX block and NEXT or cabling channel blocks shall be those defined in Table 32–8. The responses of Table 32–10 are shown in Figure 32–20. The responses represent the response of the test channel to isolated “1” symbols in a stream of “0” symbols at the input to the transmitter blocks. The test channel may also include a first order high pass filter with 3 dB cutoff frequency of less than 100 kHz in addition to the tabulated responses. The output impedance of the test channel shall be consistent with 32.6.1.4.1. The idle symbol generator outputs shall be conformant with the idle signaling specified in 32.3 with `loc_rcvr_status=OK`. The clock source shall result in a quinary symbol transmission rate conformant with 32.6.1.2.6. The peak-to-peak jitter on the clock source shall be less than 0.2 ns.

The test channel may be implemented in any fashion consistent with the above specifications and with the further constraint that the ratio of the squared error between the implemented NEXT channel symbol responses and the specified NEXT channel symbol responses to the energy in the specified NEXT channel symbol responses shall be less than 5% and the energy of the implemented NEXT channel symbol responses and the energy of the specified NEXT channel symbol responses shall differ by less than  $\pm 0.25$  dB. If digital filters are used to generate the channel characteristics, care must be taken to ensure that the signal to quantization noise at the channel output is greater than 35 dB.

The NEXT channel impulse responses defined in Table 32–11 have been developed to simulate the attenuation and phase characteristics of worst case 100BASE-T2 alien NEXT.



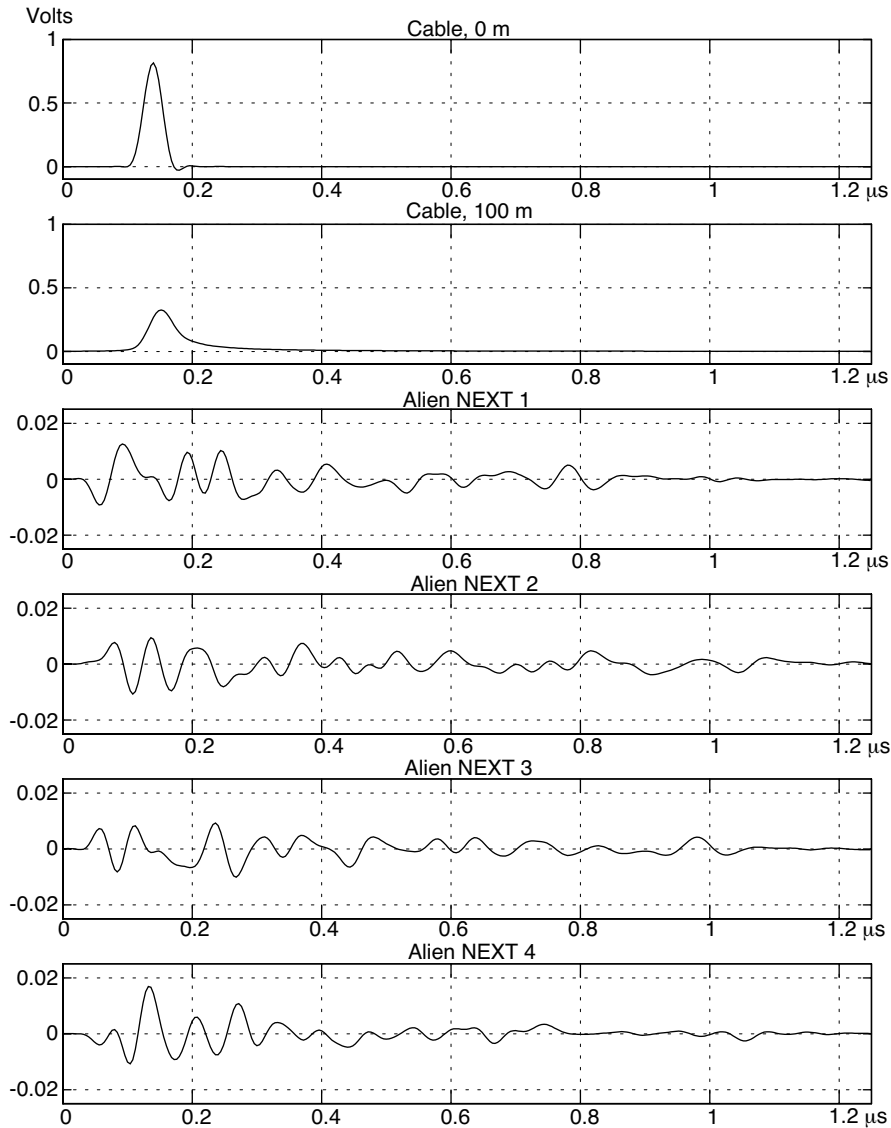
The cabling attenuation and group delay characteristics used to derive the cable symbol responses specified in Table 32–8 at 0 m and 100 m are obtained from the following worst-case model of the cabling attenuation. The model includes 1.2 dB of flat loss simulating three worst-case Category 3 connectors. The group delay of the model is relative and does not include the fixed delay associated with 100 m of Category 3 cabling. An additional 570 ns of fixed delay should be added in order to obtain the absolute group delay; however, it is not necessary to add this fixed delay to the test channel.

$$\gamma(f) = -(1.537 \times 10^{-6} \sqrt{\pi f} + j1.537 \times 10^{-6} \sqrt{\pi f} + 44.5 \times 10^{-12} 2\pi f)$$

$$H(f, l) = e^{\gamma(f)l} l_{10}^{-1.2/20}$$

$f$  in Hz

$l$  in meters



**Figure 32–20—Test channel responses**

**Table 32–11 — Coefficients for worst-case channel and T2 Alien NEXT model**

Time ( $\mu$ s)	Cable 0 m	Cable 100 m	Alien NEXT 1	Alien NEXT 2	Alien NEXT 3	Alien NEXT 4
0.000	4.23e-06	1.48e-03	1.62e-05	1.19e-05	-5.05e-06	3.70e-05
0.004	-4.87e-06	1.55e-03	-5.97e-05	1.67e-05	6.71e-05	2.02e-05
0.008	-6.84e-06	1.62e-03	-8.19e-05	1.15e-05	9.78e-05	1.20e-05
0.012	-1.28e-05	1.69e-03	-3.79e-05	7.58e-06	6.43e-05	1.50e-05
0.016	3.56e-06	1.77e-03	4.53e-05	-1.47e-05	1.52e-06	6.51e-06
0.020	6.97e-06	1.86e-03	1.40e-04	-1.73e-05	-9.59e-05	-8.16e-06
0.024	1.68e-05	1.96e-03	1.86e-04	3.46e-05	-2.01e-04	-3.64e-05
0.028	8.73e-06	2.07e-03	1.07e-04	1.67e-04	-2.67e-04	-8.81e-05
0.032	-1.98e-05	2.19e-03	-2.56e-04	3.80e-04	-1.52e-04	-2.18e-04
0.036	-2.24e-05	2.33e-03	-1.10e-03	5.84e-04	3.94e-04	-5.53e-04
0.040	-2.95e-05	2.48e-03	-2.53e-03	7.54e-04	1.49e-03	-1.14e-03
0.044	3.65e-05	2.64e-03	-4.46e-03	8.74e-04	3.09e-03	-1.95e-03
0.048	7.11e-05	2.83e-03	-6.54e-03	9.73e-04	4.89e-03	-2.83e-03
0.052	6.30e-05	3.04e-03	-8.29e-03	1.13e-03	6.41e-03	-3.56e-03
0.056	-1.42e-04	3.27e-03	-9.25e-03	1.38e-03	7.24e-03	-3.97e-03
0.060	-4.49e-04	3.53e-03	-9.04e-03	1.93e-03	6.96e-03	-3.84e-03
0.064	-2.89e-04	3.87e-03	-7.53e-03	2.90e-03	5.37e-03	-3.06e-03
0.068	-2.72e-04	4.22e-03	-4.73e-03	4.32e-03	2.51e-03	-1.69e-03
0.072	-3.87e-04	4.55e-03	-9.82e-04	5.95e-03	-1.15e-03	-5.29e-05
0.076	-1.39e-04	5.09e-03	3.14e-03	7.23e-03	-4.72e-03	1.19e-03
0.080	4.92e-04	5.83e-03	6.98e-03	7.68e-03	-7.33e-03	1.50e-03
0.084	1.50e-03	6.70e-03	9.98e-03	6.90e-03	-8.27e-03	5.20e-04
0.088	9.97e-04	7.69e-03	1.19e-02	4.74e-03	-7.36e-03	-1.68e-03
0.092	-1.45e-03	8.81e-03	1.26e-02	1.43e-03	-4.82e-03	-4.65e-03
0.096	-3.84e-03	1.04e-02	1.22e-02	-2.63e-03	-1.17e-03	-7.77e-03
0.100	-1.58e-03	1.27e-02	1.10e-02	-6.60e-03	2.75e-03	-1.01e-02
0.104	1.30e-02	1.64e-02	9.20e-03	-9.57e-03	6.05e-03	-1.08e-02
0.108	4.64e-02	2.27e-02	7.06e-03	-1.08e-02	7.97e-03	-9.32e-03
0.112	1.05e-01	3.30e-02	4.91e-03	-9.84e-03	8.20e-03	-5.54e-03
0.116	1.95e-01	4.93e-02	3.07e-03	-7.06e-03	7.00e-03	-2.20e-04
0.120	3.14e-01	7.28e-02	1.72e-03	-3.05e-03	4.92e-03	5.68e-03
0.124	4.54e-01	1.04e-01	9.15e-04	1.34e-03	2.63e-03	1.10e-02
0.128	5.95e-01	1.42e-01	6.23e-04	5.31e-03	6.32e-04	1.50e-02

**Table 32–11 – Coefficients for worst-case channel and T2 Alien NEXT model (continued)**

Time ( $\mu$ s)	Cable 0 m	Cable 100 m	Alien NEXT 1	Alien NEXT 2	Alien NEXT 3	Alien NEXT 4
0.132	7.15e-01	1.84e-01	6.67e-04	8.16e-03	-7.36e-04	1.69e-02
0.136	7.93e-01	2.28e-01	8.34e-04	9.39e-03	-1.30e-03	1.66e-02
0.140	8.15e-01	2.68e-01	7.62e-04	8.84e-03	-1.25e-03	1.44e-02
0.144	7.77e-01	3.01e-01	1.20e-04	6.66e-03	-9.31e-04	1.08e-02
0.148	6.83e-01	3.21e-01	-1.23e-03	3.32e-03	-8.01e-04	6.60e-03
0.152	5.49e-01	3.27e-01	-3.18e-03	-5.48e-04	-1.13e-03	2.50e-03
0.156	3.96e-01	3.20e-01	-5.26e-03	-4.29e-03	-1.86e-03	-1.22e-03
0.160	2.50e-01	3.01e-01	-6.94e-03	-7.31e-03	-2.84e-03	-4.34e-03
0.164	1.30e-01	2.73e-01	-7.68e-03	-9.17e-03	-3.84e-03	-6.80e-03
0.168	4.47e-02	2.40e-01	-7.12e-03	-9.59e-03	-4.72e-03	-8.47e-03
0.172	-5.75e-03	2.06e-01	-5.15e-03	-8.55e-03	-5.39e-03	-9.25e-03
0.176	-2.72e-02	1.75e-01	-1.95e-03	-6.26e-03	-5.80e-03	-9.17e-03
0.180	-2.85e-02	1.49e-01	1.90e-03	-3.26e-03	-6.01e-03	-8.19e-03
0.184	-1.82e-02	1.28e-01	5.60e-03	-1.83e-04	-6.13e-03	-6.35e-03
0.188	-5.94e-03	1.11e-01	8.37e-03	2.37e-03	-6.29e-03	-3.77e-03
0.192	2.81e-03	9.82e-02	9.58e-03	4.08e-03	-6.50e-03	-7.34e-04
0.196	6.25e-03	8.84e-02	9.06e-03	5.05e-03	-6.67e-03	2.21e-03
0.200	5.54e-03	8.06e-02	6.92e-03	5.49e-03	-6.65e-03	4.54e-03
0.204	3.70e-03	7.39e-02	3.66e-03	5.66e-03	-6.24e-03	5.81e-03
0.208	1.64e-03	6.80e-02	6.86e-05	5.68e-03	-5.26e-03	5.91e-03
0.212	5.59e-05	6.26e-02	-3.01e-03	5.52e-03	-3.62e-03	4.88e-03
0.216	-1.02e-03	5.77e-02	-4.83e-03	5.16e-03	-1.33e-03	2.95e-03
0.220	-1.53e-03	5.34e-02	-4.97e-03	4.40e-03	1.42e-03	4.48e-04
0.224	-9.73e-04	4.96e-02	-3.46e-03	3.07e-03	4.28e-03	-2.25e-03
0.228	-3.20e-04	4.63e-02	-6.22e-04	1.04e-03	6.85e-03	-4.71e-03
0.232	2.89e-05	4.33e-02	2.90e-03	-1.55e-03	8.65e-03	-6.56e-03
0.236	1.73e-04	4.06e-02	6.35e-03	-4.20e-03	9.24e-03	-7.53e-03
0.240	1.33e-04	3.83e-02	8.95e-03	-6.40e-03	8.35e-03	-7.48e-03
0.244	1.39e-04	3.62e-02	1.02e-02	-7.73e-03	5.99e-03	-6.40e-03
0.248	9.80e-05	3.42e-02	9.77e-03	-8.11e-03	2.56e-03	-4.30e-03
0.252	4.22e-05	3.25e-02	7.90e-03	-7.70e-03	-1.34e-03	-1.35e-03
0.256	-2.56e-06	3.08e-02	4.95e-03	-6.76e-03	-5.02e-03	2.14e-03
0.260	-3.84e-05	2.93e-02	1.50e-03	-5.62e-03	-7.91e-03	5.67e-03

**Table 32–11 – Coefficients for worst-case channel and T2 Alien NEXT model (continued)**

Time ( $\mu$ s)	Cable 0 m	Cable 100 m	Alien NEXT 1	Alien NEXT 2	Alien NEXT 3	Alien NEXT 4
0.264	-2.83e-05	2.80e-02	-1.85e-03	-4.58e-03	-9.67e-03	8.59e-03
0.268	-2.41e-05	2.67e-02	-4.54e-03	-3.89e-03	-1.01e-02	1.04e-02
0.272	-8.46e-06	2.55e-02	-6.29e-03	-3.61e-03	-9.30e-03	1.08e-02
0.276	4.04e-07	2.44e-02	-7.13e-03	-3.62e-03	-7.62e-03	9.78e-03
0.280	4.91e-06	2.34e-02	-7.25e-03	-3.70e-03	-5.51e-03	7.51e-03
0.284	1.01e-05	2.24e-02	-6.97e-03	-3.61e-03	-3.40e-03	4.44e-03
0.288	3.79e-06	2.15e-02	-6.54e-03	-3.18e-03	-1.47e-03	1.18e-03
0.292	2.18e-06	2.07e-02	-6.11e-03	-2.37e-03	2.01e-04	-1.65e-03
0.296	-2.23e-06	1.99e-02	-5.78e-03	-1.23e-03	1.62e-03	-3.54e-03
0.300	-1.74e-06	1.92e-02	-5.43e-03	6.10e-05	2.78e-03	-4.28e-03
0.304	4.33e-07	1.85e-02	-4.87e-03	1.26e-03	3.62e-03	-3.93e-03
0.308	2.19e-07	1.79e-02	-3.88e-03	2.10e-03	4.14e-03	-2.74e-03
0.312	1.40e-06	1.73e-02	-2.42e-03	2.37e-03	4.25e-03	-1.08e-03
0.316	-5.61e-07	1.67e-02	-7.17e-04	1.97e-03	3.87e-03	6.99e-04
0.320	-4.40e-07	1.62e-02	9.57e-04	9.16e-04	2.95e-03	2.25e-03
0.324	-4.37e-07	1.56e-02	2.28e-03	-5.89e-04	1.53e-03	3.34e-03
0.328	-3.68e-08	1.51e-02	3.07e-03	-2.20e-03	-8.99e-05	3.92e-03
0.332	9.92e-07	1.47e-02	3.23e-03	-3.52e-03	-1.59e-03	4.04e-03
0.336	5.29e-07	1.43e-02	2.74e-03	-4.19e-03	-2.63e-03	3.85e-03
0.340	5.69e-07	1.38e-02	1.73e-03	-4.00e-03	-3.00e-03	3.44e-03
0.344	-1.87e-07	1.34e-02	4.38e-04	-2.97e-03	-2.62e-03	2.85e-03
0.348	-3.47e-07	1.31e-02	-8.80e-04	-1.22e-03	-1.53e-03	2.11e-03
0.352	-9.04e-08	1.27e-02	-2.04e-03	9.56e-04	5.06e-05	1.27e-03
0.356	8.10e-08	1.24e-02	-3.01e-03	3.22e-03	1.79e-03	3.65e-04
0.360	5.29e-07	1.20e-02	-3.78e-03	5.22e-03	3.32e-03	-5.53e-04
0.364	3.23e-07	1.17e-02	-4.36e-03	6.64e-03	4.36e-03	-1.41e-03
0.368	1.82e-07	1.14e-02	-4.67e-03	7.34e-03	4.82e-03	-2.03e-03
0.372	-6.93e-08	1.11e-02	-4.60e-03	7.31e-03	4.75e-03	-2.26e-03
0.376	-1.46e-07	1.09e-02	-4.11e-03	6.65e-03	4.31e-03	-1.98e-03
0.380	6.66e-08	1.06e-02	-3.17e-03	5.50e-03	3.64e-03	-1.29e-03
0.384	1.71e-07	1.03e-02	-1.84e-03	4.02e-03	2.88e-03	-3.97e-04
0.388	3.12e-07	1.01e-02	-2.24e-04	2.44e-03	2.18e-03	4.68e-04
0.392	1.95e-07	9.86e-03	1.51e-03	9.58e-04	1.59e-03	1.07e-03

**Table 32–11 – Coefficients for worst-case channel and T2 Alien NEXT model (continued)**

Time ( $\mu$ s)	Cable 0 m	Cable 100 m	Alien NEXT 1	Alien NEXT 2	Alien NEXT 3	Alien NEXT 4
0.396	5.86e-08	9.64e-03	3.13e-03	-2.37e-04	1.17e-03	1.30e-03
0.400	-2.48e-08	9.43e-03	4.40e-03	-1.02e-03	8.60e-04	1.08e-03
0.404	-3.03e-08	9.22e-03	5.16e-03	-1.34e-03	5.83e-04	4.43e-04
0.408	1.02e-07	9.02e-03	5.37e-03	-1.16e-03	2.87e-04	-4.57e-04
0.412	1.68e-07	8.83e-03	5.08e-03	-5.37e-04	-8.75e-05	-1.43e-03
0.416	1.93e-07	8.64e-03	4.41e-03	4.06e-04	-5.80e-04	-2.27e-03
0.420	1.20e-07	8.46e-03	3.49e-03	1.39e-03	-1.26e-03	-2.94e-03
0.424	3.01e-08	8.29e-03	2.45e-03	2.10e-03	-2.18e-03	-3.47e-03
0.428	5.52e-09	8.12e-03	1.40e-03	2.28e-03	-3.31e-03	-3.92e-03
0.432	2.95e-08	7.96e-03	4.39e-04	1.83e-03	-4.53e-03	-4.33e-03
0.436	1.05e-07	7.80e-03	-4.02e-04	8.46e-04	-5.62e-03	-4.64e-03
0.440	1.40e-07	7.65e-03	-1.11e-03	-4.79e-04	-6.34e-03	-4.79e-03
0.444	1.27e-07	7.51e-03	-1.71e-03	-1.86e-03	-6.51e-03	-4.72e-03
0.448	7.69e-08	7.37e-03	-2.18e-03	-2.95e-03	-6.01e-03	-4.34e-03
0.452	2.73e-08	7.23e-03	-2.53e-03	-3.49e-03	-4.88e-03	-3.66e-03
0.456	2.59e-08	7.10e-03	-2.75e-03	-3.31e-03	-3.20e-03	-2.68e-03
0.460	5.54e-08	6.97e-03	-2.83e-03	-2.57e-03	-1.24e-03	-1.56e-03
0.464	9.74e-08	6.85e-03	-2.80e-03	-1.60e-03	7.13e-04	-5.10e-04
0.468	1.11e-07	6.73e-03	-2.65e-03	-7.72e-04	2.36e-03	2.70e-04
0.472	8.93e-08	6.61e-03	-2.40e-03	-3.48e-04	3.50e-03	6.40e-04
0.476	5.48e-08	6.49e-03	-2.08e-03	-3.49e-04	4.12e-03	6.10e-04
0.480	3.08e-08	6.39e-03	-1.72e-03	-6.97e-04	4.26e-03	2.42e-04
0.484	3.87e-08	6.28e-03	-1.36e-03	-1.18e-03	4.05e-03	-3.37e-04
0.488	6.37e-08	6.17e-03	-1.02e-03	-1.52e-03	3.61e-03	-9.74e-04
0.492	8.60e-08	6.07e-03	-7.22e-04	-1.45e-03	3.03e-03	-1.52e-03
0.496	8.68e-08	5.98e-03	-5.08e-04	-7.85e-04	2.45e-03	-1.85e-03
0.500	6.64e-08	5.88e-03	-4.43e-04	4.01e-04	1.89e-03	-1.94e-03
0.504	4.41e-08	5.79e-03	-6.03e-04	1.84e-03	1.38e-03	-1.81e-03
0.508	3.47e-08	5.70e-03	-1.02e-03	3.19e-03	9.06e-04	-1.54e-03
0.512	4.54e-08	5.61e-03	-1.69e-03	4.14e-03	4.56e-04	-1.20e-03
0.516	6.35e-08	5.52e-03	-2.55e-03	4.57e-03	6.45e-05	-7.70e-04
0.520	7.41e-08	5.44e-03	-3.47e-03	4.40e-03	-2.43e-04	-2.64e-04
0.524	6.88e-08	5.36e-03	-4.30e-03	3.69e-03	-4.49e-04	3.11e-04

**Table 32–11 – Coefficients for worst-case channel and T2 Alien NEXT model (continued)**

Time ( $\mu$ s)	Cable 0 m	Cable 100 m	Alien NEXT 1	Alien NEXT 2	Alien NEXT 3	Alien NEXT 4
0.528	5.25e-08	5.28e-03	-4.86e-03	2.59e-03	-5.97e-04	9.07e-04
0.532	3.91e-08	5.20e-03	-4.98e-03	1.30e-03	-7.31e-04	1.46e-03
0.536	3.76e-08	5.13e-03	-4.57e-03	4.15e-05	-8.65e-04	1.90e-03
0.540	4.78e-08	5.05e-03	-3.67e-03	-1.04e-03	-9.88e-04	2.14e-03
0.544	5.97e-08	4.98e-03	-2.47e-03	-1.87e-03	-1.04e-03	2.11e-03
0.548	6.32e-08	4.91e-03	-1.16e-03	-2.42e-03	-9.79e-04	1.74e-03
0.552	5.59e-08	4.84e-03	4.24e-05	-2.70e-03	-7.35e-04	1.07e-03
0.556	4.40e-08	4.78e-03	9.93e-04	-2.70e-03	-2.60e-04	2.15e-04
0.560	3.68e-08	4.71e-03	1.60e-03	-2.43e-03	4.47e-04	-6.81e-04
0.564	3.90e-08	4.65e-03	1.85e-03	-1.93e-03	1.33e-03	-1.46e-03
0.568	4.74e-08	4.59e-03	1.86e-03	-1.22e-03	2.25e-03	-2.02e-03
0.572	5.44e-08	4.53e-03	1.78e-03	-3.47e-04	3.03e-03	-2.28e-03
0.576	5.40e-08	4.47e-03	1.74e-03	6.36e-04	3.49e-03	-2.22e-03
0.580	4.68e-08	4.41e-03	1.79e-03	1.66e-03	3.55e-03	-1.86e-03
0.584	3.86e-08	4.36e-03	1.86e-03	2.66e-03	3.17e-03	-1.27e-03
0.588	3.55e-08	4.30e-03	1.85e-03	3.54e-03	2.41e-03	-5.17e-04
0.592	3.92e-08	4.25e-03	1.67e-03	4.23e-03	1.41e-03	2.65e-04
0.596	4.55e-08	4.19e-03	1.23e-03	4.64e-03	3.37e-04	9.51e-04
0.600	4.89e-08	4.14e-03	5.18e-04	4.74e-03	-6.03e-04	1.45e-03
0.604	4.65e-08	4.09e-03	-4.25e-04	4.50e-03	-1.23e-03	1.72e-03
0.608	4.03e-08	4.04e-03	-1.43e-03	3.96e-03	-1.42e-03	1.79e-03
0.612	3.52e-08	3.99e-03	-2.30e-03	3.21e-03	-1.13e-03	1.73e-03
0.616	3.46e-08	3.95e-03	-2.85e-03	2.33e-03	-4.11e-04	1.62e-03
0.620	3.84e-08	3.90e-03	-2.96e-03	1.44e-03	6.15e-04	1.56e-03
0.624	4.26e-08	3.86e-03	-2.66e-03	6.40e-04	1.76e-03	1.61e-03
0.628	4.37e-08	3.81e-03	-2.01e-03	1.25e-05	2.81e-03	1.78e-03
0.632	4.06e-08	3.77e-03	-1.16e-03	-4.14e-04	3.60e-03	2.01e-03
0.636	3.58e-08	3.72e-03	-2.62e-04	-6.60e-04	4.00e-03	2.11e-03
0.640	3.29e-08	3.68e-03	5.11e-04	-7.79e-04	3.95e-03	1.94e-03
0.644	3.37e-08	3.64e-03	1.04e-03	-8.48e-04	3.47e-03	1.38e-03
0.648	3.69e-08	3.60e-03	1.32e-03	-9.56e-04	2.68e-03	4.76e-04
0.652	3.95e-08	3.56e-03	1.38e-03	-1.17e-03	1.71e-03	-6.44e-04
0.656	3.91e-08	3.52e-03	1.32e-03	-1.52e-03	7.06e-04	-1.80e-03

**Table 32–11 – Coefficients for worst-case channel and T2 Alien NEXT model (continued)**

Time ( $\mu$ s)	Cable 0 m	Cable 100 m	Alien NEXT 1	Alien NEXT 2	Alien NEXT 3	Alien NEXT 4
0.660	3.60e-08	3.48e-03	1.24e-03	-1.97e-03	-2.19e-04	-2.78e-03
0.664	3.25e-08	3.45e-03	1.22e-03	-2.47e-03	-1.02e-03	-3.39e-03
0.668	3.12e-08	3.41e-03	1.34e-03	-2.93e-03	-1.66e-03	-3.51e-03
0.672	3.26e-08	3.38e-03	1.58e-03	-3.22e-03	-2.14e-03	-3.12e-03
0.676	3.51e-08	3.34e-03	1.92e-03	-3.27e-03	-2.45e-03	-2.36e-03
0.680	3.64e-08	3.31e-03	2.26e-03	-3.01e-03	-2.58e-03	-1.40e-03
0.684	3.52e-08	3.27e-03	2.53e-03	-2.48e-03	-2.54e-03	-4.72e-04
0.688	3.24e-08	3.24e-03	2.67e-03	-1.77e-03	-2.30e-03	2.96e-04
0.692	3.01e-08	3.21e-03	2.66e-03	-1.04e-03	-1.85e-03	8.20e-04
0.696	2.98e-08	3.17e-03	2.47e-03	-4.69e-04	-1.20e-03	1.07e-03
0.700	3.13e-08	3.14e-03	2.17e-03	-1.76e-04	-4.09e-04	1.10e-03
0.704	3.31e-08	3.11e-03	1.78e-03	-2.25e-04	4.27e-04	9.85e-04
0.708	3.34e-08	3.08e-03	1.38e-03	-6.12e-04	1.21e-03	8.49e-04
0.712	3.19e-08	3.05e-03	9.74e-04	-1.24e-03	1.87e-03	7.97e-04
0.716	2.96e-08	3.02e-03	5.40e-04	-1.94e-03	2.35e-03	8.91e-04
0.720	2.83e-08	2.99e-03	4.50e-05	-2.52e-03	2.66e-03	1.15e-03
0.724	2.86e-08	2.96e-03	-5.39e-04	-2.82e-03	2.80e-03	1.56e-03
0.728	2.99e-08	2.94e-03	-1.21e-03	-2.76e-03	2.82e-03	2.06e-03
0.732	3.10e-08	2.91e-03	-1.93e-03	-2.34e-03	2.78e-03	2.56e-03
0.736	3.07e-08	2.88e-03	-2.63e-03	-1.64e-03	2.70e-03	2.99e-03
0.740	2.91e-08	2.86e-03	-3.19e-03	-8.03e-04	2.59e-03	3.27e-03
0.744	2.75e-08	2.83e-03	-3.49e-03	2.08e-05	2.40e-03	3.36e-03
0.748	2.68e-08	2.80e-03	-3.41e-03	6.61e-04	2.08e-03	3.25e-03
0.752	2.74e-08	2.78e-03	-2.90e-03	9.96e-04	1.62e-03	2.96e-03
0.756	2.85e-08	2.75e-03	-1.98e-03	9.72e-04	1.01e-03	2.54e-03
0.760	2.89e-08	2.73e-03	-7.19e-04	6.06e-04	3.03e-04	2.04e-03
0.764	2.83e-08	2.71e-03	7.27e-04	-1.29e-05	-4.51e-04	1.55e-03
0.768	2.69e-08	2.68e-03	2.19e-03	-7.52e-04	-1.17e-03	1.09e-03
0.772	2.57e-08	2.66e-03	3.49e-03	-1.45e-03	-1.76e-03	6.97e-04
0.776	2.55e-08	2.64e-03	4.45e-03	-1.95e-03	-2.16e-03	3.84e-04
0.780	2.62e-08	2.61e-03	4.96e-03	-2.13e-03	-2.35e-03	1.50e-04
0.784	2.70e-08	2.59e-03	4.94e-03	-1.91e-03	-2.33e-03	-9.21e-06
0.788	2.70e-08	2.57e-03	4.41e-03	-1.30e-03	-2.16e-03	-1.09e-04

**Table 32–11 – Coefficients for worst-case channel and T2 Alien NEXT model (continued)**

Time ( $\mu$ s)	Cable 0 m	Cable 100 m	Alien NEXT 1	Alien NEXT 2	Alien NEXT 3	Alien NEXT 4
0.792	2.62e-08	2.55e-03	3.43e-03	-3.53e-04	-1.88e-03	-1.67e-04
0.796	2.50e-08	2.53e-03	2.14e-03	7.94e-04	-1.52e-03	-1.97e-04
0.800	2.43e-08	2.51e-03	6.97e-04	1.99e-03	-1.11e-03	-2.12e-04
0.804	2.44e-08	2.48e-03	-7.29e-04	3.07e-03	-6.76e-04	-2.16e-04
0.808	2.50e-08	2.46e-03	-1.98e-03	3.92e-03	-2.42e-04	-2.11e-04
0.812	2.55e-08	2.44e-03	-2.95e-03	4.47e-03	1.71e-04	-1.93e-04
0.816	2.52e-08	2.43e-03	-3.56e-03	4.69e-03	5.47e-04	-1.58e-04
0.820	2.44e-08	2.41e-03	-3.79e-03	4.59e-03	8.49e-04	-1.06e-04
0.824	2.35e-08	2.39e-03	-3.68e-03	4.24e-03	1.04e-03	-4.15e-05
0.828	2.31e-08	2.37e-03	-3.29e-03	3.71e-03	1.09e-03	3.17e-05
0.832	2.34e-08	2.35e-03	-2.72e-03	3.10e-03	9.78e-04	1.09e-04
0.836	2.38e-08	2.33e-03	-2.02e-03	2.46e-03	7.26e-04	1.89e-04
0.840	2.40e-08	2.31e-03	-1.28e-03	1.85e-03	3.61e-04	2.70e-04
0.844	2.36e-08	2.29e-03	-5.54e-04	1.30e-03	-7.48e-05	3.50e-04
0.848	2.29e-08	2.28e-03	9.66e-05	8.67e-04	-5.29e-04	4.35e-04
0.852	2.22e-08	2.26e-03	6.28e-04	5.63e-04	-9.51e-04	5.28e-04
0.856	2.21e-08	2.24e-03	1.02e-03	3.91e-04	-1.30e-03	6.25e-04
0.860	2.24e-08	2.23e-03	1.26e-03	3.27e-04	-1.54e-03	7.05e-04
0.864	2.27e-08	2.21e-03	1.35e-03	3.18e-04	-1.67e-03	7.38e-04
0.868	2.27e-08	2.19e-03	1.32e-03	3.13e-04	-1.71e-03	7.00e-04
0.872	2.22e-08	2.18e-03	1.20e-03	2.49e-04	-1.65e-03	5.79e-04
0.876	2.15e-08	2.16e-03	1.06e-03	6.50e-05	-1.54e-03	3.86e-04
0.880	2.11e-08	2.15e-03	9.49e-04	-2.75e-04	-1.39e-03	1.42e-04
0.884	2.11e-08	2.13e-03	9.05e-04	-7.77e-04	-1.23e-03	-1.19e-04
0.888	2.14e-08	2.11e-03	9.32e-04	-1.40e-03	-1.08e-03	-3.50e-04
0.892	2.16e-08	2.10e-03	1.01e-03	-2.06e-03	-9.51e-04	-5.12e-04
0.896	2.14e-08	2.08e-03	1.10e-03	-2.69e-03	-8.58e-04	-5.74e-04
0.900	2.09e-08	2.07e-03	1.17e-03	-3.22e-03	-8.19e-04	-5.37e-04
0.904	2.04e-08	2.06e-03	1.17e-03	-3.60e-03	-8.53e-04	-4.30e-04
0.908	2.02e-08	2.04e-03	1.07e-03	-3.81e-03	-9.70e-04	-2.86e-04
0.912	2.02e-08	2.03e-03	8.84e-04	-3.86e-03	-1.17e-03	-1.38e-04
0.916	2.05e-08	2.01e-03	6.43e-04	-3.76e-03	-1.43e-03	-3.73e-06
0.920	2.05e-08	2.00e-03	4.00e-04	-3.57e-03	-1.72e-03	1.07e-04



**Table 32–11 – Coefficients for worst-case channel and T2 Alien NEXT model (continued)**

Time ( $\mu$ s)	Cable 0 m	Cable 100 m	Alien NEXT 1	Alien NEXT 2	Alien NEXT 3	Alien NEXT 4
0.924	2.03e-08	1.99e-03	2.04e-04	-3.32e-03	-2.00e-03	1.97e-04
0.928	1.98e-08	1.97e-03	8.04e-05	-3.05e-03	-2.21e-03	2.83e-04
0.932	1.94e-08	1.96e-03	3.54e-05	-2.77e-03	-2.31e-03	3.84e-04
0.936	1.93e-08	1.95e-03	6.11e-05	-2.49e-03	-2.25e-03	5.12e-04
0.940	1.94e-08	1.93e-03	1.19e-04	-2.19e-03	-2.02e-03	6.60e-04
0.944	1.96e-08	1.92e-03	1.68e-04	-1.87e-03	-1.61e-03	7.99e-04
0.948	1.95e-08	1.91e-03	1.69e-04	-1.51e-03	-1.04e-03	8.98e-04
0.952	1.92e-08	1.90e-03	1.12e-04	-1.11e-03	-3.35e-04	9.23e-04
0.956	1.88e-08	1.88e-03	3.20e-05	-6.75e-04	4.63e-04	8.56e-04
0.960	1.86e-08	1.87e-03	-3.29e-05	-2.21e-04	1.31e-03	6.88e-04
0.964	1.85e-08	1.86e-03	-3.97e-05	2.28e-04	2.16e-03	4.27e-04
0.968	1.86e-08	1.85e-03	3.82e-05	6.47e-04	2.94e-03	1.05e-04
0.972	1.87e-08	1.84e-03	2.03e-04	1.01e-03	3.58e-03	-2.35e-04
0.976	1.86e-08	1.82e-03	4.37e-04	1.30e-03	4.03e-03	-5.42e-04
0.980	1.83e-08	1.81e-03	6.79e-04	1.50e-03	4.23e-03	-7.76e-04
0.984	1.80e-08	1.80e-03	8.54e-04	1.61e-03	4.14e-03	-9.10e-04
0.988	1.78e-08	1.79e-03	8.92e-04	1.65e-03	3.77e-03	-9.27e-04
0.992	1.78e-08	1.78e-03	7.59e-04	1.61e-03	3.13e-03	-8.27e-04
0.996	1.79e-08	1.77e-03	4.78e-04	1.51e-03	2.31e-03	-6.23e-04
1.000	1.79e-08	1.76e-03	9.92e-05	1.35e-03	1.38e-03	-3.43e-04
1.004	1.77e-08	1.75e-03	-3.03e-04	1.14e-03	4.42e-04	-2.81e-05
1.008	1.74e-08	1.74e-03	-6.46e-04	8.71e-04	-4.26e-04	2.80e-04
1.012	1.72e-08	1.73e-03	-8.64e-04	5.31e-04	-1.16e-03	5.35e-04
1.016	1.71e-08	1.71e-03	-9.09e-04	1.17e-04	-1.71e-03	6.95e-04
1.020	1.71e-08	1.70e-03	-7.85e-04	-3.71e-04	-2.06e-03	7.20e-04
1.024	1.72e-08	1.69e-03	-5.39e-04	-9.22e-04	-2.21e-03	5.83e-04
1.028	1.71e-08	1.68e-03	-2.39e-04	-1.51e-03	-2.20e-03	2.80e-04
1.032	1.69e-08	1.67e-03	4.62e-05	-2.08e-03	-2.04e-03	-1.70e-04
1.036	1.67e-08	1.66e-03	2.68e-04	-2.58e-03	-1.78e-03	-7.21e-04
1.040	1.65e-08	1.66e-03	3.93e-04	-2.92e-03	-1.45e-03	-1.31e-03
1.044	1.65e-08	1.65e-03	4.08e-04	-3.07e-03	-1.08e-03	-1.86e-03
1.048	1.65e-08	1.64e-03	3.24e-04	-2.97e-03	-6.97e-04	-2.29e-03
1.052	1.65e-08	1.63e-03	1.64e-04	-2.62e-03	-3.38e-04	-2.53e-03

**Table 32–11 – Coefficients for worst-case channel and T2 Alien NEXT model (continued)**

Time ( $\mu$ s)	Cable 0 m	Cable 100 m	Alien NEXT 1	Alien NEXT 2	Alien NEXT 3	Alien NEXT 4
1.056	1.64e-08	1.62e-03	-2.93e-05	-2.04e-03	-2.07e-05	-2.54e-03
1.060	1.62e-08	1.61e-03	-2.24e-04	-1.30e-03	2.39e-04	-2.32e-03
1.064	1.60e-08	1.60e-03	-3.91e-04	-4.72e-04	4.34e-04	-1.91e-03
1.068	1.59e-08	1.59e-03	-5.13e-04	3.49e-04	5.60e-04	-1.37e-03
1.072	1.59e-08	1.58e-03	-5.79e-04	1.08e-03	6.21e-04	-7.75e-04
1.076	1.59e-08	1.57e-03	-5.84e-04	1.66e-03	6.27e-04	-2.13e-04
1.080	1.59e-08	1.56e-03	-5.35e-04	2.04e-03	5.90e-04	2.42e-04
1.084	1.57e-08	1.56e-03	-4.46e-04	2.24e-03	5.24e-04	5.43e-04
1.088	1.55e-08	1.55e-03	-3.40e-04	2.26e-03	4.42e-04	6.80e-04
1.092	1.54e-08	1.54e-03	-2.40e-04	2.14e-03	3.58e-04	6.70e-04
1.096	1.53e-08	1.53e-03	-1.64e-04	1.95e-03	2.82e-04	5.53e-04
1.100	1.53e-08	1.52e-03	-1.21e-04	1.72e-03	2.23e-04	3.73e-04
1.104	1.53e-08	1.51e-03	-1.09e-04	1.48e-03	1.90e-04	1.72e-04
1.108	1.53e-08	1.51e-03	-1.19e-04	1.27e-03	1.85e-04	-1.10e-05
1.112	1.51e-08	1.50e-03	-1.38e-04	1.09e-03	2.03e-04	-1.51e-04
1.116	1.49e-08	1.49e-03	-1.54e-04	9.49e-04	2.33e-04	-2.31e-04
1.120	1.48e-08	1.48e-03	-1.57e-04	8.36e-04	2.60e-04	-2.52e-04
1.124	1.48e-08	1.47e-03	-1.42e-04	7.46e-04	2.72e-04	-2.18e-04
1.128	1.48e-08	1.47e-03	-1.14e-04	6.77e-04	2.53e-04	-1.33e-04
1.132	1.48e-08	1.46e-03	-8.40e-05	6.30e-04	1.95e-04	-6.35e-06
1.136	1.47e-08	1.45e-03	-6.31e-05	6.05e-04	9.72e-05	1.50e-04
1.140	1.45e-08	1.44e-03	-5.84e-05	5.99e-04	-2.78e-05	3.12e-04
1.144	1.44e-08	1.44e-03	-7.04e-05	6.01e-04	-1.58e-04	4.51e-04
1.148	1.43e-08	1.43e-03	-9.63e-05	6.02e-04	-.69e-04	5.45e-04
1.152	1.43e-08	1.42e-03	-1.29e-04	5.87e-04	-3.41e-04	5.76e-04
1.156	1.43e-08	1.42e-03	-1.60e-04	5.43e-04	-3.62e-04	5.36e-04
1.160	1.42e-08	1.41e-03	-1.81e-04	4.58e-04	-3.29e-04	4.30e-04
1.164	1.41e-08	1.40e-03	-1.83e-04	3.30e-04	-2.48e-04	2.72e-04
1.168	1.40e-08	1.39e-03	-1.65e-04	1.72e-04	-1.35e-04	9.19e-05
1.172	1.39e-08	1.39e-03	-1.28e-04	4.95e-06	-1.13e-05	-7.86e-05
1.176	1.38e-08	1.38e-03	-7.75e-05	-1.48e-04	1.02e-04	-2.10e-04
1.180	1.38e-08	1.37e-03	-1.90e-05	-2.62e-04	1.90e-04	-2.83e-04
1.184	1.38e-08	1.37e-03	4.03e-05	-3.19e-04	2.48e-04	-2.93e-04

**Table 32–11 — Coefficients for worst-case channel and T2 Alien NEXT model (continued)**

Time ( $\mu$ s)	Cable 0 m	Cable 100 m	Alien NEXT 1	Alien NEXT 2	Alien NEXT 3	Alien NEXT 4
1.188	1.37e-08	1.36e-03	9.40e-05	-3.07e-04	2.73e-04	-2.43e-04
1.192	1.36e-08	1.35e-03	1.37e-04	-2.28e-04	2.68e-04	-1.50e-04
1.196	1.35e-08	1.35e-03	1.66e-04	-9.21e-05	2.37e-04	-3.64e-05
1.200	1.34e-08	1.34e-03	1.80e-04	7.95e-05	1.86e-04	7.45e-05
1.204	1.34e-08	1.34e-03	1.79e-04	2.63e-04	1.22e-04	1.62e-04
1.208	1.34e-08	1.33e-03	1.62e-04	4.36e-04	5.03e-05	2.17e-04
1.212	1.33e-08	1.32e-03	1.27e-04	5.81e-04	-2.48e-05	2.37e-04
1.216	1.33e-08	1.32e-03	7.69e-05	6.83e-04	-9.88e-05	2.28e-04
1.220	1.32e-08	1.31e-03	1.24e-05	7.35e-04	-1.66e-04	2.01e-04
1.224	1.31e-08	1.30e-03	-6.03e-05	7.34e-04	-2.22e-04	1.66e-04
1.228	1.30e-08	1.30e-03	-1.36e-04	6.87e-04	-2.61e-04	1.35e-04
1.232	1.29e-08	1.29e-03	-2.14e-04	6.00e-04	-2.84e-04	1.13e-04
1.236	1.29e-08	1.29e-03	-2.89e-04	4.86e-04	-2.93e-04	1.02e-04
1.240	1.29e-08	1.28e-03	-3.66e-04	3.57e-04	-2.97e-04	9.58e-05
1.244	1.29e-08	1.27e-03	-4.41e-04	2.30e-04	-3.00e-04	9.07e-05
1.248	1.29e-08	1.27e-03	-5.17e-04	1.15e-04	-3.10e-04	7.92e-05

### 32.6.1.3.2 Receiver test mode

To facilitate the testing of the receiver in the presence of synchronous 100BASE-T2 alien NEXT, a special receiver test mode shall be required to allow for receiver alien NEXT tolerance and jitter testing. For a PHY with an MII, this mode shall be enabled by setting bit 9.13 (MASTER-SLAVE Control register) of the MII management register set to a 1. A PHY without an MII shall provide a means to enable this test mode. This mode shall not be overridden except by clearing bit 9.13 or resetting the PHY.

When the receive test mode is enabled, the receiver shall configure itself in SLAVE mode, continually attempt to bring its receiver up until successful receiver operation is achieved and transmit symbols in idle mode. For a PHY with an MII, when the receiver is properly detecting the received data (loc\_rcvr\_status=OK), it shall set bit 10.13 of the MII management register to 1 and reset the error count in bits 10.0 through 10.7 (MSB) to zero. The error count shall be incremented for every symbol error detected in the received idle sequence (where rem\_rcvr\_status is assumed to be OK). Upon loss of proper data reception, the receiver shall clear bit 10.13. A PHY without an MII shall provide a means to realize this function. The vendor shall provide a means to enable this mode for conformance testing.

### 32.6.1.3.3 Receiver differential input signals

Differential signals received on the receive inputs that were transmitted within the specifications given in 32.6.1.2, and have then passed through a link as defined in 32.7, shall be translated into one of the

PMA\_UNITDATA.indicate messages with a symbol error rate less than  $10^{-10}$  and sent to the PCS after link bring-up.

Performance shall be tested in at least two configurations: using a 100 m link segment conformant to 32.7 and with a link segment less than 1 m in length between transmitter and receiver.

### 32.6.1.3.4 Receiver Alien NEXT tolerance

Differential signals received from the test channel defined in 32.6.1.3.1 shall be detected with a symbol error rate less than  $10^{-8}$  when the PHY is in receiver test mode for the following combinations of channel and worst-case alien NEXT responses, as shown in Table 32-13.

**Table 32-13—Receiver Alien NEXT test cases**

Case	Cable channels	NEXT Channels			
		A1	A2	B1	B2
1	0 m	Alien NEXT 1	Alien NEXT 3	Alien NEXT 4	Alien NEXT 2
2	0 m	Alien NEXT 1	Alien NEXT 4	Alien NEXT 3	Alien NEXT 2
3	100 m	Alien NEXT 1	Alien NEXT 3	Alien NEXT 4	Alien NEXT 2
4	100 m	Alien NEXT 1	Alien NEXT 4	Alien NEXT 3	Alien NEXT 2

NOTE—Implementors will find it practically impossible to meet the requirements of this subclause without using some form of adaptive equalization and cyclostationary interference suppression.

### 32.6.1.3.5 Receiver timing jitter

For the test channels described below, the peak-to-peak value of RX\_CLK zero-crossing jitter shall be less than 1.3 ns after the receiver is properly receiving the data and has set bit 9.13 of the MII management register set to 1. When the jitter waveform on RX\_CLK is filtered by a high pass filter having the transfer function below,<sup>16</sup> the peak-to-peak value of the resulting filtered timing jitter shall be less than 0.8 ns.

Test channels:

Channels 1–4 are the test channels described in 32.6.1.3.1 with the four combinations of worst-case channel and alien NEXT responses tabulated in 32.6.1.3.4.

Channels 5–6 are the test channels described in 32.6.1.3.1 with the combinations of worst-case channel and alien NEXT responses tabulated in cases 1 and 2 of 32.6.1.3.4 plus the addition of 100 m of Category 3 compliant cable between the test channel fixture and the PHY under test.

$$H_{\text{jitterfilter}}(f) = \frac{jf}{jf + 1000} \quad f \text{ in Hz}$$

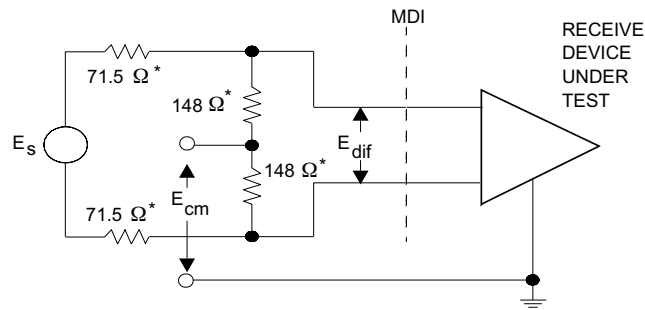
The RX\_CLK of the MII shall be made available for this test. A PHY without an MII shall provide an equivalent clock.

### 32.6.1.3.6 Common-mode noise rejection

While receiving packets from a compliant 100BASE-T2 transmitter, connected to all MDI pins, a receiver shall send the proper PMA\_UNITDATA.indicate messages to the PCS for any differential input signal  $E_s$  that results in a signal  $E_{\text{dif}}$  that meets 32.6.1.3.3 even in the presence of common mode voltages  $E_{\text{cm}}$  (applied

<sup>16</sup>“j” denotes the positive square root of  $-1$ .

as shown in Figure 32–21).  $E_{cm}$  shall be a 25 V peak-to-peak square wave, 500 kHz or lower in frequency, with edges no slower than 4 ns (20%-80%), connected to each of the pairs (BI\_DA+, BI\_DA-) and (BI\_DB+, BI\_DB-).



\* Resistor matching to 1 part in 1 000.

**Figure 32–21 — Receiver common-mode noise rejection test circuit**

### 32.6.1.3.7 Receiver frequency tolerance

The receive feature shall properly receive incoming data with a 5-level symbol rate within the range 25.000 MHz  $\pm$  0.01%.

### 32.6.1.4 MDI Specifications

#### 32.6.1.4.1 MDI differential impedance

The differential impedance as measured at the MDI for each transmit/receive channel shall be such that any reflection due to differential signals incident upon the MDI from a balanced cabling having an impedance of 100  $\Omega$  is at least 17 dB below the incident signal over the frequency range 2.0–6.5 MHz and at least  $12.9 - 20 \log_{10}(f/10)$  dB over the frequency range 6.5–25 MHz ( $f$  in MHz). This return loss shall be maintained at all times when the PHY is transmitting data.

#### 32.6.1.4.2 MDI impedance balance

Over the frequency range 2.0–25.0 MHz, the common-mode to differential-mode impedance balance of each channel of the MDI shall exceed

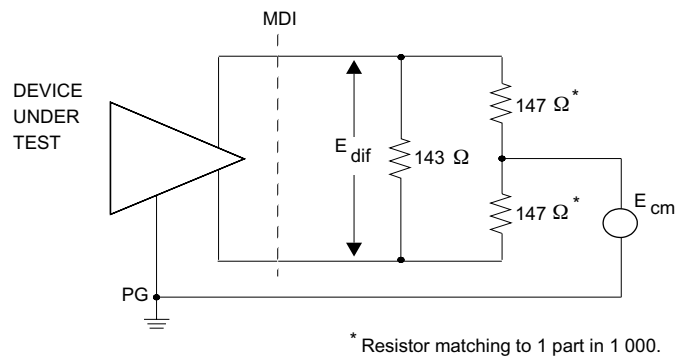
$$29 - 17 \log \left( \frac{f}{10} \right) \text{ dB}$$

where  $f$  is the frequency in MHz when the transmitter is transmitting idle mode data (transmit test mode 3).

The balance is defined as

$$20 \log \left( \frac{E_{cm}}{E_{dif}} \right)$$

where  $E_{cm}$  is an externally applied sine wave voltage as shown in Figure 32–22 and  $E_{dif}$  is the resulting waveform due only to the applied sine wave and not the transmitted data.<sup>17</sup>

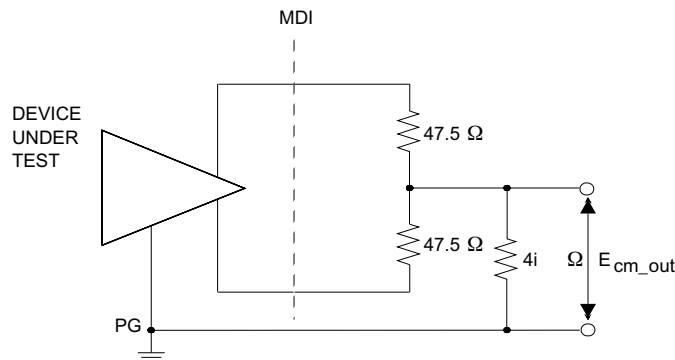


**Figure 32-22—MDI impedance balance test circuit**

NOTE—The balance of the test equipment (such as the matching of the test resistors) must be insignificant relative to the balance requirements.

#### 32.6.1.4.3 MDI common-mode output voltage

The implementor should consider any applicable local, national, or international regulations. Driving balanced cable pairs with high-frequency common mode voltages may cause radiated emissions which may result in interference to other equipment. FCC conducted and radiated emissions tests may require that the magnitude of the total common mode output voltage,  $E_{cm\_out}$ , on any transmit circuit, when measured as shown in Figure 32-23, be less than a few millivolts when transmitting data.



**Figure 32-23—Common-mode output voltage test circuit**

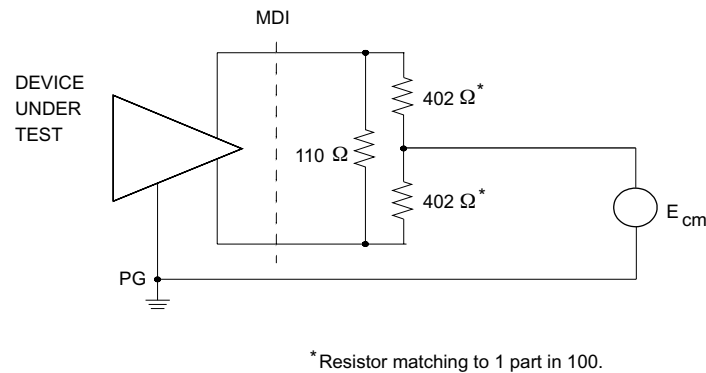
NOTE—The balance of the test equipment (such as the matching of the test resistors) must be insignificant relative to the balance requirements.

#### 32.6.1.4.4 MDI fault tolerance

Transmitters and receivers shall withstand without damage the application of short circuits across any MDI port for an indefinite period of time and shall resume normal operation after such faults are removed. The magnitude of the current through such a short circuit shall not exceed 300 mA.

Transmitters shall withstand without damage a 1000 V common-mode impulse applied at  $E_{cm}$  of either polarity (as indicated in Figure 32-24). The shape of the impulse shall be 0.3/50  $\mu$ s (300 ns virtual front time, 50  $\mu$ s virtual time of half value), as defined in IEC 60060.

<sup>17</sup>Triggered averaging can be used to separate the component due to the applied common-mode sine wave from the transmitted data component.



**Figure 32-24—MDI fault tolerance test circuit**

### 32.6.2 Power consumption

After 100 ms following PowerOn, the current drawn by the PHY shall not exceed 1.0 A when powered through the MII.

The PHY shall be capable of operating from all voltage sources allowed by Clause 22, including those current limited to 1.0 A, as supplied by the DTE or repeater through the resistance of all permissible MII cabling.

The PHY shall not introduce extraneous signals on the MII control circuits during normal power-up and power-down.

While in power-down mode, the PHY is not required to meet any of the 100BASE-T2 performance requirements.

## 32.7 Link segment characteristics

100BASE-T2 employs a dual duplex transmission system, i.e., two full duplex channels are used simultaneously to transmit data. The use of the term *link segment* in this clause refers to two duplex channels and the specifications for a link segment apply individually to each of the two duplex channels. Furthermore, the term duplex channel will be used to refer a single channel of the dual duplex link segment.

100BASE-T2 is designed to allow use of the pairs of the cabling other than the two used for the full duplex channels of the 100BASE-T2 service. Services supported for use in the other pairs are as follows:

- a) 100BASE-T2
- b) 10BASE-T
- c) Digital phone services compliant with the ITU-T Recommendation I.430 and ANSI T1.605 and T1.601

### 32.7.1 Cabling

Cabling and installation practices generally suitable for use with this standard appear in ISO/IEC 11801. Exceptions, notes, and additional requirements are as listed below.

- a) 100BASE-T2 uses a star topology. Balanced cabling is used to connect PHY entities.
- b) 100BASE-T2 is an ISO 11801 class C application, with additional installation requirements and transmission parameters specified in 32.7.2–32.7.4. The width of the PAM5 × 5 transmit spectrum is

approximately 25 MHz (as shown in Figure 32–19). The aggregate data rate for two pairs using PAM5 × 5 coding is 100 Mb/s.

- c) 100BASE-T2 shall use 2 pairs of balanced cabling, Category 3 or better, with a nominal characteristic impedance of 100 Ω.
- d) When using Category 3 cabling for the link segment, Clause 32 recommends, but does not require, the use of Category 4 or better connecting hardware, patch cords and jumpers. The use of Category 4 or better connecting hardware increases the link segment composite NEXT loss, composite ELFEXT loss and reduces the link segment insertion loss. This lowers the link segment crosstalk noise which in turn decreases the probability of errors.
- e) The use of shielding is outside the scope of this standard.
- f) The use of other cabling systems is discussed in Annex 32A.

### 32.7.2 Link transmission parameters

Unless otherwise specified, link segment testing shall be conducted using source and load impedances of 100 Ω.

The tolerance on the poles of the test filter used in this clause shall be ± 1%.

#### 32.7.2.1 Insertion loss

The insertion loss of a link segment shall be no more than 14.6 dB at all frequencies between 2 and 16 MHz. This consists of the attenuation of the balanced cabling pairs, connector losses, and reflection losses due to impedance mismatches between the various components of the link segment. The insertion loss specification shall be met when the link segment is terminated in source and load impedances that satisfy 32.6.1.4.1.

NOTE—The loss of PVC-insulated cabling exhibits significant temperature dependence. At temperatures greater than 40 °C, it may be necessary to use a less temperature-dependent cabling, such as many Fluorinated Ethylene Propylene (FEP), Polytetrafluoroethylene (PTFE), or Perfluoroalkoxy (PFA) plenum-rated cabling.

#### 32.7.2.2 Differential characteristic impedance

The cable used in the links shall meet the requirements for characteristic impedance specified in ISO/IEC 11801. Connecting hardware shall meet the return loss requirements for connecting hardware specified in ISO/IEC 11801.

#### 32.7.2.3 Coupling parameters

In order to limit the noise coupled into a duplex channel from an adjacent duplex channel, Near-End Crosstalk (NEXT) loss and Equal Level Far-End Crosstalk (ELFEXT) loss are specified for each link segment. In addition, since two dual-duplex connections may co-exist in a 4-pair cabling and a receiver on a duplex channel will be disturbed by crosstalk from one to three other duplex (or simplex) channels, Multiple-Disturber NEXT loss and Multiple-Disturber ELFEXT loss are also specified. When a 10BASE-T service is used within the same cabling, a restriction on the allowable NEXT loss to Insertion Loss (NIR) of the cabling is required and is specified in 32.7.2.3.5.

##### 32.7.2.3.1 Differential near-end crosstalk (NEXT) loss

The differential Near-End Crosstalk (NEXT) loss between the two duplex channels of a link segment is specified in order to limit the crosstalk noise at the near end of a link segment to meet the symbol error rate objective specified in 32.1 and the noise specifications of 32.7.3. The NEXT loss between the two duplex channels of a link segment shall be at least  $19.3 - 16.6 \log_{10}(f/16)$  (where  $f$  is the frequency in MHz) over the frequency range 2–16 MHz.



### 32.7.2.3.2 Multiple-disturber NEXT (MDNEXT) loss

Since two dual duplex applications (connections) may exist in a 4-pair cabling system, a received signal may be disturbed by multiple alien NEXT signals. The MDNEXT loss between each link segment duplex channel and the two alien data carrying duplex channels shall be at least  $19.0-16.6\log_{10}(f/16)$  dB (where  $f$  is the frequency in MHz) over the frequency range 2.0–16 MHz. MDNEXT is computed as the power sum of the individual NEXT losses. This specification is consistent with two disturbers, each with a NEXT loss of at least  $22.0-16.6\log_{10}(f/16)$ .

NOTE—Since the self NEXT noise from the other duplex channel of a connection can be cancelled using digital signal processing techniques whereas the alien NEXT noise from an alien connection can not be cancelled in the same fashion, the self NEXT noise is treated differently than the alien NEXT noise and is not included in the MDNEXT calculation.

### 32.7.2.3.3 Equal level far-end crosstalk loss (ELFEXT)

Equal Level Far-End Crosstalk (ELFEXT) loss is specified in order to limit the crosstalk noise at the far end of a link segment to meet the symbol error rate objective specified in 32.1 and the noise specifications of 32.7.3. Far-End Crosstalk (FEXT) noise is the crosstalk noise that appears at the far end of one of the duplex channels which is coupled from one of the duplex channels with the noise source (transmitters) at the near end. ELFEXT loss is the ratio of the data signal to FEXT noise at the far end output of a duplex channel. To limit the FEXT noise from an adjacent duplex channel, the ELFEXT loss between each duplex channel shall be greater than  $20.9-20\log_{10}(f/16)$  dB (where  $f$  is the frequency in MHz) over the frequency range 2–16 MHz. ELFEXT loss at frequency  $f$  and distance  $l$  is defined as

$$\text{ELFEXT\_Loss}(f,l) = 20\log_{10}(V_{pds}/V_{pcn}) - \text{SLS\_Loss}(\text{dB})$$

where  $V_{pds}$  = peak voltage of disturbing signal (near-end transmitter),  $V_{pcn}$  = peak crosstalk noise at far-end of disturbed channel, and  $\text{SLS\_Loss}$  = insertion loss of the disturbing channel.

### 32.7.2.3.4 Multiple-disturber ELFEXT (MDELNEXT) loss

Since two duplex channels are used to transfer data between PHYs and two connections can exist in a 4-pair cabling, the FEXT noise that is coupled into a data carrying duplex channel is from one to three disturbers. The MDELNEXT loss between a duplex channel and the other data carrying duplex channels shall be greater than  $19.9-20\log_{10}(f/16)$  (where  $f$  is the frequency in MHz) over the frequency range 2–16 MHz. This specification is consistent with three disturbers, one with a FEXT loss of at least  $20.9-20\log_{10}(f/16)$  and two with a FEXT loss of at least  $27.0-20\log_{10}(f/16)$ .

MDELNEXT is computed as the power sum of the individual FEXT losses.

### 32.7.2.3.5 10BASE-T NEXT loss to insertion loss ratio requirement

The objective of this specification is to support the coexistence of a 100BASE-T2 link segment and a 10BASE-T link segment in a 4-pair cable. When a 100BASE-T2 link segment operates in the same 4-pair cable with a 10BASE-T link segment, each 100BASE-T2 duplex channel will receive alien NEXT noise signals from the 10BASE-T link segment. To ensure reliable operation, a minimum signal to noise ratio must be maintained. This minimum signal to noise ratio is assured by meeting the following NEXT loss to insertion loss ratio (NIR).

NIR is defined by the following equation:

$$\text{NIR (dB)} = (\text{AdjustedNEXTLoss} - \text{InsertionLoss}_{6 \text{ MHz}})$$

where  $InsertionLoss_{6\text{ MHz}}$  is the maximum of the insertion loss at 6 MHz of the two duplex channels of the 100BASE-T2 link segment and  $AdjustedNEXTLoss$  is determined by the following algorithm:

#### AdjustedNEXTLoss Algorithm

*Step 1.* Measure the NEXT loss as a function of frequency over the range 1–16 MHz for each of the six pair combinations between the four pairs of the cabling. The maximum spacing in frequency of the samples shall be 250 kHz.

*Step 2.* Add  $16.6\log_{10}(f/16)$  to the NEXT loss measurements (where  $f$  is frequency in MHz) to normalize the NEXT loss as a function of frequency.

*Step 3.* Determine the minimum value of the normalized NEXT loss across the frequency range over all pair combinations. The minimum value is the AdjustedNEXTLoss.

The NIR shall be greater than 19.4 dB.

#### 32.7.2.4 Delay

Since 100BASE-T2 sends information over two duplex channels in parallel, the propagation delay of each channel and the difference in delay are specified to comply with network round-trip delay of the two channels and ensure proper decoding by receivers, respectively.

##### 32.7.2.4.1 Maximum link delay

The propagation delay of a link segment shall not exceed 5.7 ns/m at all frequencies between 2 and 25 MHz.

##### 32.7.2.4.2 Difference in link delays

The difference in propagation delay, or skew, under all conditions, between the two duplex channels of a link segment shall not exceed 90 ns at all frequencies between 2 and 25 MHz. It is a further functional requirement that, once installed, the skew between the duplex links due to environmental conditions shall not vary more than  $\pm 20$  ns, within the above requirement.

#### 32.7.3 Noise

The noise level on the link segments shall be such that the cabling noise requirements which follow are met. The noise environment consists generally of a main contributor, self-induced and alien near-end crosstalk noise, and a lesser contributor, far-end crosstalk noise.

The noise environment for 100BASE-T2 can consist of the following elements:

- a) Echo from the local transmitter on the same pair (duplex channel). Echo is caused by the hybrid used to achieve simultaneous bidirectional transmission of data in the T2 system and by impedance mismatches in the link segment. It is practically impossible to achieve robust performance without using echo cancellation to reduce this noise to a small residual. Echo cancellation is possible since the symbols transmitted from the disturbing local transmitter are available to the cancellation processor.
- b) Near-end crosstalk (NEXT) noise from the local transmitter on the other pair (duplex channel) of the link segment. This is often referred to as self NEXT noise since the source is from the same link segment. NEXT noise cancellation is typically used to reduce this noise to a small residual. NEXT noise cancellation is possible since the symbols transmitted from the disturbing local transmitter are available to the cancellation processor.
- c) Far-end crosstalk (FEXT) noise from the remote transmitters on the other pair (duplex channel) of the link segment. This is often referred to as self FEXT noise since the source is from the same link segment. Self FEXT noise can not be cancelled in the same way as echo and self NEXT noise since

the symbols from the remote transmitter are not immediately available; however, in the link configurations used for 100BASE-T2, self FEXT noise is much smaller than self NEXT noise and can generally be neglected.<sup>18</sup>

- d) Noise from non-idealities in the duplex channels, transmitters and receivers; for example, DAC/ADC non-linearity, electrical noise (shot and thermal) and non-linear channel characteristics.
- e) Noise from sources outside the cabling which couple into the link segment via electric and magnetic fields.
- f) Noise from services in adjacent wire pairs in the same cable sheath. These services generate near- and far-end crosstalk and are often referred to as alien NEXT noise and alien FEXT noise since the sources are not from the link segment of the disturbed duplex channel. Since the transmitted symbols from an alien NEXT noise source are not available to the PHY of the disturbed duplex channel, it is not possible to cancel the alien NEXT noise as can be done for self NEXT noise. If the alien NEXT noise is from a 100BASE-T2 transceiver, a technique termed cyclostationary interference suppression can be used to suppress the alien NEXT noise. It will be practically impossible achieve reliable operation in the presence of alien NEXT noise meeting the limits of the specifications in subclause 32.6 without using some form of cyclostationary interference suppression. 10BASE-T can not be suppressed and therefore an additional constraint has been placed on the link (see subclause 32.7.2.3.5) to ensure adequate signal to noise levels for reliable performance. Digital phone services compliant with the ITU-T Recommendation I.430 and ANSI T1.605 and T1.601 also can not be suppressed but produce substantially smaller crosstalk than 10BASE-T and thus do not require any additional constraints on the link.

100BASE-T2 supports three types of service in adjacent pairs of the same cable: 100BASE-T2, 10BASE-T, and digital phone service compliant with the ISDN-BR U and S/T interfaces. Analog phone service is not supported since the noise generated during off-hook transitions and ringing source from older PBX equipment can cause bit errors to occur.

NOTE—Due to the use of noise cancellation, cyclostationary interference suppression and the use of adaptive equalization, there is no meaningful way to add up the noises at the input to the receiver into an overall noise level and simulation of a design is required to determine the contribution of each source to the final error at the symbol decision point.

### 32.7.3.1 Near-end crosstalk noise

The MDNEXT (Multiple-Disturber Near-End Crosstalk) noise on a duplex channel from an alien connection depends on the signal spectrum on the alien channels and the crosstalk between the alien channels and the disturbed channel.

The MDNEXT noise on each duplex link of a link segment shall not exceed 182 mVp.

This specification is compatible with the following assumptions:

- a) Two disturbing alien pairs with a NEXT loss greater than 22.0 dB at 16 MHz
- b) All disturbers combined on a power sum basis

The MDNEXT noise is the noise measured at the output of a filter connected to the output of the near end of a disturbed link segment using maximum level 100BASE-T2 transmitters attached to the near end of an alien disturbing link segment. Each continuous transmit signal is generated by a transceiver in idle mode meeting the data scrambling and encoding rules in 32.3, e.g., a transmitter in transmit test mode 3.

<sup>18</sup>Additionally, FEXT noise may be suppressed to some degree via cyclostationary interference suppression; however, in the presence of alien NEXT noise, the equalizer will be primarily suppressing the alien NEXT noise.

### 32.7.3.2 Far-end crosstalk noise

The MDFEXT (Multiple-Disturber Far-End Crosstalk) noise on a duplex channel depends on the signal spectrum on the disturbing channels and the various crosstalk losses between those channels and the disturbed channel.

The MDFEXT noise on a link segment shall not exceed 54.4 mVp.

This specification is compatible with the following assumptions:

- a) One disturbing pair with ELFEXT (Equal Level Far-End Crosstalk) loss greater than 20.9 dB at 16 MHz
- b) Two additional disturbers with ELFEXT (Equal Level Far-End Crosstalk) loss greater than 27.0 dB at 16 MHz
- c) All disturbers combined on a power sum basis

The MDFEXT noise is the noise measured at the output of a filter connected to the output of the far end of a disturbed link segment using maximum level 100BASE-T2 transmitters attached to the near end of the other duplex channel of the link segment and both duplex channels of an alien disturbing link segment. Each continuous transmit signal is generated by a transceiver in idle mode meeting the data scrambling and encoding rules in 32.3, e.g., a transmitter in transmit test mode 3.

The filter is a 5th order Butterworth filter with a 3 dB cutoff at 23 MHz.

### 32.7.3.3 External coupled noise

Noise coupled from external sources, measured at the output of a filter connected to the output of the near end of a disturbed link segment shall not exceed 25 mV peak.<sup>19</sup>

The filter is a 5th order Butterworth filter with a 3 dB cutoff at 23 MHz.

## 32.7.4 Installation practice

### 32.7.4.1 Connector installation practices

The amount of untwisting in a pair as a result of termination to connecting hardware should be no greater than 25 mm (1.0 in.) for Category 3 cabling. This is the same value recommended in ISO/IEC 11801 for Category 4 connectors.

### 32.7.4.2 Restrictions on use of Category 3 cabling with more than four pairs

Jumper cabling, or horizontal runs, made from more than four pairs of Category 3 cabling shall not be used.

### 32.7.4.3 Restrictions on use of Category 5 cabling with up to 25 pairs

Cables made from up to 25 pairs of Category 5 cable are allowed. Such cables, if used, shall be limited in length to no more than 90 m total. The services in the cable shall be limited to any combination 100BASE-T2, 10BASE-T and digital phone services compliant with the ITU-T Recommendation I.430 and ANSI T1.605 and T1.601 interfaces up to a total of 12 services in the cable.

---

<sup>19</sup>This assumes the link has worst-case attenuation and alien NEXT and that the noise has the worst possible properties. In the absence of alien NEXT the tolerance to external noise sources is substantially increased. Tolerance to stationary noise such as continuous wave interference from AM radio can be substantially higher since the equalizer can notch out frequencies with poor signal to noise ratios. Tolerance to isolated impulse noise events is also typically much higher and dependent on the shape of the impulse.

## 32.8 MDI specification

This subclause defines the MDI. The link topology requires a crossover function between PMAs. Implementation and location of this crossover are also defined in this subclause.

### 32.8.1 MDI connectors

Eight-pin connectors meeting the requirements of section 3 and Figures 1–4 of IEC 60603-7, Detail Specification for Connectors, 8-Way shall be used as the mechanical interface to the balanced cabling. The plug connector shall be used on the balanced cabling and the jack on the PHY. These connectors are depicted (for informational use only) in Figures 32–25 and 32–26. Table 32–14 shows the assignment of PMA signals to connector contacts for PHYs.

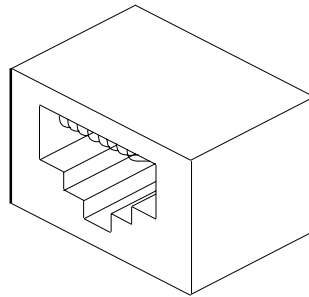


Figure 32–25—MDI connector

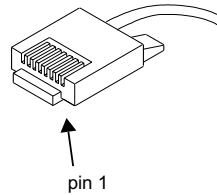


Figure 32–26—Balanced cabling connector

Table 32–14—Assignment of PMA signals to MDI pin-outs

Contact	PHY without internal crossover (100BASE-T2 operation)	PHY with internal crossover (Auto-Negotiation operation)	MDI labeling requirement
1	BI_DA+	BI_DB+	BI_DA+
2	BI_DA-	BI_DB-	BI_DA-
3	BI_DB+	BI_DA+	BI_DB+
4	Not used	Not used	
5	Not used	Not used	
6	BI_DB-	BI_DA-	BI_DB-
7	Not used	Not used	
8	Not used	Not used	

### 32.8.2 Crossover function

Although the crossover function is not required for successful operation of 100BASE-T2, it is a functional requirement that a crossover function be implemented in every link segment to support the operation of Auto-Negotiation. The crossover function connects the transmitters of one PHY to the receivers of the PHY at the other end of the link segment. Crossover functions may be implemented internally to a PHY or elsewhere in the link segment. For a PHY that does not implement the crossover function, the MDI labels in the last column of Table 32–14 refer to its own internal circuits (second column). For PHYs that do implement the internal crossover, the MDI labels in the last column of Table 32–14 refer to the internal circuits of the remote PHY of the link segment. Additionally, the MDI connector for a PHY that implements the crossover function shall be marked with the graphical symbol X. The crossover function specified here is compatible with the crossover function specified in 14.5.2 for pairs TD and RD.

When a link segment connects a DTE to a repeater, it is recommended the crossover be implemented in the PHY local to the repeater. If both PHYs of a link segment contain internal crossover functions, an additional external crossover is necessary. It is recommended that the crossover be visible to an installer from one of the PHYs. When both PHYs contain internal crossovers, it is further recommended in networks in which the topology identifies either a central backbone segment or a central repeater that the PHY furthest from the central element be assigned the external crossover to maintain consistency.

Implicit implementation of the crossover function within a twisted-pair cable, or at a wiring panel, while not expressly forbidden, is beyond the scope of this standard.

## 32.9 System considerations

The repeater unit specified in Clause 27 forms the central unit for interconnecting 100BASE-T2 twisted-pair links in networks of more than two nodes. It also provides the means for connecting 100BASE-T2 balanced cabling links to other 100 Mb/s baseband segments. The proper operation of a CSMA/CD network requires that network size be limited to control round-trip propagation delay as specified in Clause 29.

When operated in Full Duplex mode where CSMA/CD requirements do not apply, 100BASE-T2 balanced cabling links are limited to 100 m as per ISO/IEC 11801.

## 32.10 Environmental specifications

### 32.10.1 General safety

All equipment meeting this standard shall conform to IEC 60950.

### 32.10.2 Network safety

This clause sets forth a number of recommendations and guidelines related to safety concerns; the list is neither complete nor does it address all possible safety issues. The designer is urged to consult the relevant local, national, and international safety regulations to ensure compliance with the appropriate requirements.

LAN cabling systems described in this clause are subject to at least four direct electrical safety hazards during their installation and use. These hazards are as follows:

- a) Direct contact between LAN components and power, lighting, or communications circuits
- b) Static charge buildup on LAN cabling and components
- c) High-energy transients coupled onto the LAN cabling system
- d) Voltage potential differences between safety grounds to which various LAN components are connected

Such electrical safety hazards must be avoided or appropriately protected against for proper network installation and performance. In addition to provisions for proper handling of these conditions in an operational system, special measures must be taken to ensure that the intended safety features are not negated during installation of a new network or during modification or maintenance of an existing network.

### 32.10.2.1 Installation

It is a mandatory functional requirement that sound installation practice, as defined by applicable local codes and regulations, be followed in every instance in which such practice is applicable.

### 32.10.2.2 Grounding

Any safety grounding path for an externally connected PHY shall be provided through the circuit ground of the MII connection.

#### WARNING

It is assumed that the equipment to which the PHY is attached is properly grounded, and not left floating nor serviced by a “doubly insulated, ac power distribution system.” The use of floating or insulated equipment, and the consequent implications for safety, are beyond the scope of this standard.

### 32.10.2.3 Installation and maintenance guidelines

It is a mandatory functional requirement that, during installation and maintenance of the cabling plant, care be taken to ensure that non-insulated network cabling conductors do not make electrical contact with unintended conductors or ground.

### 32.10.2.4 Telephony voltages

The use of building wiring brings with it the possibility of wiring errors that may connect telephony voltages to 100BASE-T2 equipment. Other than voice signals (which are low voltage), the primary voltages that may be encountered are the “battery” and ringing voltages. Although there is no universal standard, the following maximums generally apply.

Battery voltage to a telephone line is generally 56 Vdc applied to the line through a balanced 400  $\Omega$  source impedance.

Ringing voltage is a composite signal consisting of an AC component and a DC component. The ac component is up to 175 V peak at 20 Hz to 60 Hz with a 100  $\Omega$  source resistance. The DC component is 56 Vdc with 300–600  $\Omega$  source resistance. Large reactive transients can occur at the start and end of each ring interval.

Although 100BASE-T2 equipment is not required to survive such wiring hazards without damage, application of any of the above voltages shall not result in any safety hazard.

NOTE—Wiring errors may impose telephony voltages differentially across 100BASE-T2 transmitters or receivers. Because the termination resistance likely to be present across a receiver’s input is of substantially lower impedance than an off-hook telephone instrument, receivers will generally appear to the telephone system as off-hook telephones. Therefore, full-ring voltages will be applied for only short periods. Transmitters that are coupled using transformers will similarly appear like off-hook telephones (though perhaps a bit more slowly) due to the low resistance of the transformer coil.

### 32.10.3 Environment

#### 32.10.3.1 Electromagnetic emission

The twisted-pair link shall comply with applicable local and national codes for the limitation of electromagnetic interference.

#### 32.10.3.2 Temperature and humidity

The twisted-pair link is expected to operate over a reasonable range of environmental conditions related to temperature, humidity, and physical handling (such as shock and vibration). Specific requirements and values for these parameters are considered to be beyond the scope of this standard.

It is recommended that manufacturers indicate in the literature associated with the PHY the operating environmental conditions to facilitate selection, installation, and maintenance.

### 32.10.4 Cabling specifications

All equipment subject to this clause shall conform to the requirements of 14.7 and applicable sections of ISO/IEC 11801.

### 32.11 PHY labeling

It is recommended that each PHY (and supporting documentation) be labeled in a manner visible to the user with at least the following parameters:

- a) Data rate capability in Mb/s
- b) Power level in terms of maximum current drain (for external PHYs)
- c) Port type (i.e., 100BASE-T2)
- d) Any applicable safety warnings

See also 32.8.2.

### 32.12 Delay constraints

Proper operation of a CSMA/CD LAN, operating in half duplex mode, demands that there be an upper bound on the propagation delays through the network. This implies that MAC, PHY and repeater implementors must conform to certain delay minima and maxima, and that network planners and administrators conform to constraints regarding the cabling topology and concatenation of devices. MAC constraints are contained in Clause 21. Topological constraints are contained in Clause 29. In the full duplex mode of operation, DTEs are not required to conform to the constraints specified in this clause.

The reference point for all MDI measurements is the peak point of the mid-cell transition corresponding to the reference code-bit, as measured at the MDI. Although 100BASE-T2 output is scrambled, it is assumed that these measurements are made via apparatuses that appropriately account for this.

#### 32.12.1 PHY delay constraints (exposed MII)

Every 100BASE-T2 PHY with an exposed MII shall comply with the bit delay constraints specified in Table 32–15. These figures apply for all 100BASE-T2 PMDs.



**Table 32–15—MDI to MII delay constraints (exposed MII)**

Sublayer Measurement Points	Event	Min (bits)	Max (bits)	Input Timing Reference	Output Timing Reference
MII ⇔ MDI	TX_EN Sampled to MDI Output	7	9	TX_CLK rising	1st symbol of SSD
	MDI input to CRS assert		25	1st symbol of SSD	
	MDI input to CRS de-assert		29	1st ONE	
	MDI input to COL assert		25	1st symbol of SSD	
	MDI input to COL de-assert		29	1st symbol of SSD	
	TX_EN sampled to CRS assert	0	4	TX_CLK rising	
	TX_EN sampled to CRS de-assert	0	16	TX_CLK rising	

**32.12.2 DTE delay constraints (unexposed MII)**

Every 100BASE-T2 DTE with no exposed MII shall comply with the bit delay constraints specified in Table 32–16. These figures apply for all 100BASE-T2 PMDs.

**Table 32–16—DTE delay constraints (unexposed MII)**

Sublayer Measurement Points	Event	Min (bits)	Max (bits)	Input Timing Reference	Output Timing Reference
MAC ⇔ MDI	MAC transmit start to MDI output		13		1st symbol of SSD
	MDI input to MDI output (worst-case non-deferred transmit)		50	1st symbol of SSD	1st symbol of SSD
	MDI input to collision detect		33	1st symbol of SSD	
	MDI input to MDI output = Jam (worst-case collision response)		50	1st symbol of SSD	1st symbol of SSD

**32.13 Protocol Implementation Conformance Statement (PICS) proforma for Clause 32, Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) sublayer and baseband medium, type 100BASE-T2<sup>20</sup>**

The supplier of a protocol implementation that is claimed to conform to Clause 32, Physical Coding Sublayer (PCS), Physical Medium Attachment (PMA) sublayer and baseband medium, type 100BASE-T2, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

Instructions for interpreting and filling out the PICS proforma may be found in Clause 21.

<sup>20</sup>Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this subclause so that it can be used for its intended purpose and may further publish the completed PICS.

**32.13.1 Identification****32.13.1.1 Implementation identification**

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Name(s)	
NOTE 1— Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.	
NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).	

**32.13.1.2 Protocol summary**

Identification of protocol specification	IEEE Std 802.3-2002 <sup>®</sup> , Clause 32, Physical Coding Sub-layer (PCS), Physical Medium Attachment (PMA) sub-layer and baseband medium, type 100BASE-T2
Identification of amendments and corrigenda to this PICS proforma which have been completed as part of this PICS	
Have any Exceptions items been required? No [ ] Yes [ ] (See Clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3-2002 <sup>®</sup> .)	

Date of Statement	
-------------------	--

**32.13.2 Major capabilities/options**

Item	Feature	Subclause	Status	Support	Value/Comment
*MII	Exposed MII interface	32.1.3.2	O	Yes [ ] No [ ]	Devices supporting this option must also support the PCS option.
PC	PHY Control functions	32.2	M	Yes [ ]	Required for proper operation of the PHY.
*PCS	PCS functions	32.3	O	Yes [ ] No [ ]	Required for integration with DTE or MII.
*PMA	Exposed PMA service interface	32.4.2	O	Yes [ ] No [ ]	Required for integration into symbol level repeater core.
NWY	Support for Auto-Negotiation (Clause 28)	32.1.3.4	M	Yes [ ]	Required
*INS	Installation/cabling	32.7.4	O	Yes [ ] No [ ]	Items marked with INS include installation practices and cabling specifications not applicable to a PHY manufacturer.

**32.13.3 Compatibility considerations**

Item	Feature	Subclause	Status	Support	Value/Comment
CCO1	Compatibility at the MDI	32.1.3.1	M	Yes [ ]	
CCO2	Auto-Negotiation required	32.1.3.4	M	Yes [ ]	
CCO3	State diagrams prevail	32.1.4	M	Yes [ ]	In discrepancy between text and state diagram.

**32.13.4 PHY control function**

Item	Feature	Subclause	Status	Support	Value/Comment
PC01	PHY Control shall	32.2.1	M	Yes [ ]	Comply with the state diagram descriptions given in Figure 32-5.
PC02	PHY Control shall	32.2.2.1.2	M	Yes [ ]	Generate PHYC_CONFIG.indicate messages synchronously with every MII TX_CLK cycle.
PC03	Upon receipt of the PHYC_CONFIG primitive, PCS Transmit and PMA Clock Recovery shall	32.2.2.1.3	M	Yes [ ]	Perform their functions in master or slave configuration according to the value of the parameter config.
PC04	PHY Control shall	32.2.2.2.2	M	Yes [ ]	Generate PHYC_LOCRXSTATUS.indicate messages synchronously with every MII TX_CLK cycle.
PC05	Upon reception of the PHYC_LOCRXSTATUS.indicate primitive, PCS Transmit shall	32.2.2.2.3	M	Yes [ ]	Perform its function according to the value assumed by the parameter loc_cvr_status.
PC06	PHY Control shall	32.2.2.3.2	M	Yes [ ]	Generate PHYC_TXMODE.indicate messages synchronously with every MII TX_CLK cycle.
PC07	Upon receipt of the PHYC_TXMODE.indicate primitive, the PCS shall	32.2.2.3.3	M	Yes [ ]	Perform its Transmit function as described in 32.3.1.2.
PC8	The PCS shall	32.2.2.3.2	M	Yes [ ]	Generate PHYC_RXSTATUS.request messages synchronously with signals received at the MDI.
PC9	The PCS shall	32.2.2.4.2	M	Yes [ ]	Generate PHYC_REMRXSTATUS.request messages synchronously with signals received at the MDI.
PC10	PCS Transmit shall	32.2.3	M	Yes [ ]	Send quinary symbols according to the value assumed by tx_mode.
PC11	When tx_mode assumes the value of SEND_N	32.2.3	M	Yes [ ]	Transmission of sequences of quinary symbols representing an MII data stream, the idle mode, or control sequences shall take place.
PC12	When tx_mode assumes the value of SEND_I	32.2.3	M	Yes [ ]	Transmission of sequences of quinary symbols representing the idle mode shall take place.

**32.13.5 Physical Coding Sublayer (PCS) or Physical Medium Attachment (PMA) sublayer****32.13.5.1 PCS transmit functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PCT01	The PCS shall	32.3.5.1	M	Yes [ ]	Implement the Transmit process as depicted in Figure 32–12 including compliance with the associated state variables specified in 32.3.4.
PCT02	PCS Transmit function shall	32.3.1.2	M	Yes [ ]	Conform to the PCS Transmit state diagram in Figure 32–12.
PCT03	If the parameter config provided to the PCS by the PHY Control function via the PHYC_CONFIG.indicate message assumes the value MASTER, PCS Transmit shall	32.3.1.2.1	M	Yes [ ]	Employ the transmitter side-stream scrambler generator polynomial specified for use with MASTER in 32.3.1.2.1.
PCT04	If the parameter config provided to the PCS by the PHY Control function via the PHYC_CONFIG.indicate message assumes the value SLAVE, PCS Transmit shall	32.3.1.2.1	M	Yes [ ]	Employ the transmitter side-stream scrambler generator polynomial specified for use with SLAVE in 32.3.1.2.1.
PCT05	Transmission of quinary symbol pairs over wire pairs	32.3.1.2	M	Yes [ ]	Symbols $A_n$ and $B_n$ are transmitted over BI_DA and BI_DB respectively.

**32.13.5.2 PCS receive functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PCR01	The PCS shall	32.3.5.2	M	Yes [ ]	Implement the Receive process as depicted in Figure 32–13 including compliance with the associated state variables as specified in 32.3.4.
PCR02	PCS Receive function shall	32.3.1.3	M	Yes [ ]	Conform to the PCS Receive state diagram shown in Figure 32–13.
PCR03	For side-stream descrambling, the MASTER PHY shall employ	32.3.1.3.1	M	Yes [ ]	The receiver scrambler generator polynomial specified for MASTER operation in 32.3.1.3.1.
PCR04	For side-stream descrambling, the SLAVE PHY shall employ	32.3.1.3.1	M	Yes [ ]	The receiver scrambler generator polynomial specified for SLAVE operation in 32.3.1.3.1.

**32.13.5.3 Other PCS functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PCO1	The PCS Reset function shall	32.3.1.1	M	Yes [ ]	Be executed any time “power on,” receipt of a request for reset from the management entity, or receipt of a request for reset from the PHY Control occur.
PCO2	THE PCS shall	32.3.1.4	M	Yes [ ]	Implement the PCS Carrier Sense function shown in Figure 32–14.
PCO3	PCS Carrier Sense function shall	32.3.5.3	M	Yes [ ]	Comply with the PCS Carrier Sense state diagram shown in Figure 32–14 including compliance with the associated state variables specified in 32.3.4.
PCO4	MII COL signal shall be asserted while	32.3.1.5	M	Yes [ ]	A PCS collision is being detected.
PCO5	The MII signal COL shall remain de-asserted.	32.3.1.5	M	Yes [ ]	A PCS collision is not being detected.
PCO6	No spurious PCS management entity signals shall be emitted onto the MDI while the PHY is held in power down mode as defined in 22.2.4.1.5, independently of the value of TX_EN, or when released from power down mode, or when power is first applied to the PHY.	32.3.2.2	M	Yes [ ]	
PCO7	Frames passed from the PCS to the PMA sublayer shall	32.3.3	M	Yes [ ]	Have the structure shown in Figure 32–11.
PCO8	TX_CLK shall be generated	32.3.4.2	M	Yes [ ]	Synchronously with symb_timer with tolerance as specified in 32.6.1.2.6.

**32.13.5.4 PMA functions**

Item	Feature	Subclause	Status	Support	Value/Comment
PMF1	PMA Reset function shall be executed	32.4.1.1.1	M	Yes [ ]	At power on and upon receipt of a reset request from the management entity.
PMF2	PMA transmit shall continuously transmit	32.3.1.2	M	Yes [ ]	Onto the MDI pulses modulated by the quinary symbols given by tx_symb_vector[BI_DA] and tx_symb_vector[BI_DB] respectively.
PMF3	The two transmitters shall be driven by the same transmit clock.	32.4.1.1.2	M	Yes [ ]	
PMF4	PMA Transmit function will follow	32.4.1.1.2	M	Yes [ ]	The mathematical description given in 32.4.1.2.1.
PMF5	PMA Transmit shall comply with	32.4.1.1.2	M	Yes [ ]	The electrical specifications given in 32.6.
PMF6	PMA Receive function shall	32.4.1.1.3	M	Yes [ ]	Translate the signals received on pairs BI_DA and BI_DB into the PMA_UNITDATA.indicate parameter rx_symb_vector with a symbol error rate of less than one part in 10 <sup>10</sup> .
PMF7	The Link Monitor function shall	32.4.1.1.4	M	Yes [ ]	Comply with the state diagram shown in Figure 32–17.
PMF8	Clock Recovery function shall provide	32.4.1.1.5	M	Yes [ ]	A clock suitable for synchronous signal sampling on each line so that the symbol-error rate indicated in 32.4.1.1.3 is achieved.
PMF9	The waveform obtained at the MDI shall comply with	32.4.1.2.1	M	Yes [ ]	The electrical specifications given in 32.6.
PMF10	The two signals received on pair BI_DA and BI_DB shall be processed within the PMA Receive function to yield	32.4.1.2.2	M	Yes [ ]	The quinary received symbols rx_symb_vector[BI_DA] and rx_symb_vector[BI_DB].

**32.13.5.5 PMA service interface**

Item	Feature	Subclause	Status	Support	Value/Comment
PMS1	Continuous generation of PMA_TYPE	32.4.1.2.2	M	Yes [ ]	
PMS2	Generation of PMA_UNITDATA.indicate (SYMB_PAIR) messages	32.4.2.3.2	M	Yes [ ]	Synchronous with data received at the MDI.
PMS3	The PMA shall generate	32.4.2.5.2	M	Yes [ ]	PMA_LINK.indicate to indicate the value of link_status.
PMS4	Effect of receipt of PMA_LINK.request messages	32.4.2.4.3	M	Yes [ ]	While link_control=SCAN_FOR_CARRIER or link_control=DISABLE, the PMA shall report link_status=fail; while link_control=ENABLE, PHY determines operations to be performed by the PHY.



**32.13.5.6 Management functions**

Item	Feature	Subclause	Status	Support	Value/Comment
MF1	A 100BASE-T2 PHY shall	32.5	M	Yes [ ]	Use register addresses 9 and 10 for its control and status functions.
MF2	Register 9 shall	32.5.3.1	M	Yes [ ]	Provide values as specified in Table 32–4.
MF3	A PHY with 100BASE-T2 capability shall	32.5.3.1.1	M	Yes [ ]	Be placed in transmitter test mode 1 when bit 9.15 is set to logic zero and bit 9.14 is set to logic one.
MF4	A PHY with 100BASE-T2 capability shall	32.5.3.1.1	M	Yes [ ]	Be placed in transmitter test mode 2 when bit 9.15 is set to logic one and bit 9.14 is set to logic zero.
MF5	A PHY with 100BASE-T2 capability shall	32.5.3.1.1	M	Yes [ ]	Be placed in transmitter test mode 3 when bit 9.15 is set to logic one and bit 9.14 is set to logic one.
MF6	A PHY with 100BASE-T2 capability shall	32.5.3.1.2	M	Yes [ ]	Be placed in receiver test mode when bit 9.13 is set to logic one.
MF7	MASTER-SLAVE configuration negotiation will determine PHY configuration if	32.5.3.1.3	M	Yes [ ]	Bit 9.12 is set to logic zero.
MF8	Manual MASTER-SLAVE configuration will be set manually using bit 9.11 to set the value if	32.5.3.1.3	M	Yes [ ]	Bit 9.12 is set to logic one.
MF9	Bit 9.11 shall be used to report the results of manual MASTER-SLAVE configuration.	32.5.3.1.4	M	Yes [ ]	
MF10	Bit 9.10 shall	32.5.3.1.5	M	Yes [ ]	Be set to logic zero if the PHY is a DTE device and be set to logic one if the PHY is a repeater device port.
MF11	Bits 9.9:0 shall	32.5.3.1.6	M	Yes [ ]	Be set to logic zero.
MF12	Bits 9.9:0 shall be ignored when read	32.5.3.1.6	M	Yes [ ]	
MF13	A PHY shall return a value of zero for bits 9.9:0.	32.5.3.1.6	M	Yes [ ]	
MF14	Register 10 shall	32.5.3.1.2	M	Yes [ ]	Be used as shown in Table 32–5.
MF15	Bits 10.11:8 shall	32.5.3.2.5	M	Yes [ ]	Be set to logic zeros by default.
MF16	The MASTER-SLAVE Manual Configuration Fault bit shall be implemented	32.5.3.2.1	M	Yes [ ]	With a latching function, such that the occurrence of a MASTER-SLAVE Manual Configuration Fault will cause the MASTER-SLAVE Manual Configuration Fault bit to be set and remain set until cleared.
MF17	The MASTER-SLAVE Manual Configuration Fault bit shall be cleared	32.5.3.2.1	M	Yes [ ]	Each time register 10 is read by the management interface.

**32.13.5.6 Management functions (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
MF18	The MASTER-SLAVE Management Configuration Fault bit shall also be cleared by	32.5.3.2.1	M	Yes [ ]	A 100BASE-T2 PMA reset.
MF19	Bits 10:11:8 shall	32.5.3.2.5	M	Yes [ ]	Be ignored when read.
MF20	A PHY shall return a value of zero for bits 10:11:8.	32.5.3.2.5	M	Yes [ ]	

**32.13.5.7 100BASE-T2 specific Auto-Negotiation requirements**

Item	Feature	Subclause	Status	Support	Value/Comment
AN1	Base Page will be followed with	32.5.4.2	M	Yes [ ]	A Next Page with a message code containing the 100BASE-T2 Technology Ability Message Code (7).
AN2	Message Next Page shall be followed with	32.5.4.2	M	Yes [ ]	Unformatted Message Next Page containing the 100BASE-T2 Technology Ability Fields as described in Table 32-6.
AN3	MASTER-SLAVE relationship shall be determined by	32.5.4.3	M	Yes [ ]	Using Table 32-7 with the 100BASE-T2 Technology Ability Next Page bit values specified in Table 32-6.
AN4	A seed counter shall be provided to	32.5.4.3	M	Yes [ ]	Track the generation of seeds.
AN5	At start-up, the seed counter shall be set to	32.5.4.3	M	Yes [ ]	Zero.
AN6	The seed counter shall be incremented	32.5.4.3	M	Yes [ ]	Every time a new random seed is sent.
AN7	Maximum seed attempts before declaring a MASTER_SLAVE configuration Resolution Fault	32.5.4.3	M	Yes [ ]	Seven.
AN8	During MASTER_SLAVE configuration, the device with the lower seed value shall	32.5.4.3	M	Yes [ ]	Become the SLAVE.
AN9	Both PHYs set in manual mode to be either MASTER or SLAVE shall be treated as	32.5.4.3	M	Yes [ ]	MASTER-SLAVE resolution fault (Failure) condition
AN10	MASTER-SLAVE resolution fault (failure) condition shall result in	32.5.4.3	M	Yes [ ]	MASTER-SLAVE Configuration Resolution Fault bit (10.15) to be set
AN11	MASTER-SLAVE Configuration resolution fault condition shall be treated as	32.5.4.3	M	Yes [ ]	MASTER-SLAVE Configuration Resolution complete

**32.13.5.8 PMA electrical specifications**

Item	Feature	Subclause	Status	Support	Value/Comment
PME1	The value of all components in test circuits shall be accurate to within	32.6	M	Yes [ ]	±1%.
PME2	The PHY shall provide electrical isolation between	32.6.1.1	M	Yes [ ]	The DTE or repeater circuits including frame ground, and all MDI leads.
PME3	PHY-provided electrical separation shall withstand at least one of three electrical strength tests	32.6.1.1	M	Yes [ ]	a) 1500 V rms at 50–60Hz for 60 s, applied as specified in section 5.3.2 of IEC 60950. b) 2250 Vdc for 60 s, applied as specified in Section 5.3.2 of IEC 60950. c) a sequence of ten 2400 V impulses of alternating polarity, applied at intervals of not less than 1 s. The shape of the impulses shall be 1.2/50 μs. (1.2 μs virtual front time, 50 μs virtual time or half value), as defined in IEC 60950.
PME4	There shall be no insulation breakdown as defined in Section 5.3.2 of IEC 60950, during the test.	32.6.1.1	M	Yes [ ]	
PME5	The resistance after the test shall be at least	32.6.1.1	M	Yes [ ]	2 MΩ, measured at 500 Vdc.
PME6	The PMA shall provide the Transmit function specified in 32.4.1.1.2 in accordance with the electrical specifications of this clause.	32.6.1.2	M	Yes [ ]	
PME7	Where a load is not specified, the transmitter shall meet all the requirements of this clause with a 100 Ω resistive differential load connected to each transmitter output.	32.6.1.2	M	Yes [ ]	
PME8	The tolerance on the poles of the test filters used in 32.6 shall be	32.6.1.2	M	Yes [ ]	±1%
PME9	A special transmit test mode shall be required to allow for testing of the transmitter waveform	32.6.1.2.1	M	Yes [ ]	
PME10	A test mode for measuring transmitter output jitter is required.	32.6.1.2.1	M	Yes [ ]	
PME11	For a PHY with a MII interface, the transmit test mode shall be enabled by	32.6.1.2.1	M	Yes [ ]	Setting bit 9.15 and 9.14 (MASTER-SLAVE Control register) of the MII management register set as shown in Table 32–6.

**32.13.5.8 PMA electrical specifications (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
PME12	These test modes shall only change the data symbols provided to the transmitter circuitry and may not alter the electrical characteristics of the transmitter	32.6.1.2.1	M	Yes [ ]	
PME13	When transmit test mode 1 is enabled, the PHY shall transmit	32.6.1.2.1	M	Yes [ ]	The sequence of data symbols specified in 32.6.1.2.1 continuously from both transmitters.
PME14	When in test mode 1, the transmitter shall time the transmitted symbols	32.6.1.2.1	M	Yes [ ]	From a 25.000 MHz $\pm$ 0.01% clock.
PME15	When test mode 2 is enabled, the PHY shall transmit	32.6.1.2.1	M	Yes [ ]	The data symbol sequence {+2,-2} repeatedly on both channels.
PME16	When in test mode 2, the transmitter shall time the transmitted symbols	32.6.1.2.1	M	Yes [ ]	From a 25.000 MHz $\pm$ 0.01% clock.
PME17	A PHY without a MII shall provide a means to enter this test mode.	32.6.1.2.1	M	Yes [ ]	
PME18	The vendor shall	32.6.1.2.1	M	Yes [ ]	Provide a means to enable these modes for testing.
PME19	When in transmit test mode 1 and observing the differential signal output at the MDI, terminated in 100 $\Omega$ , preprocessed by the high pass filter defined below, for either pair, with no intervening cable, the absolute value of the peak of the waveform at points A and B as defined in Figure 32-18 shall fall within	32.6.1.2.2	M	Yes [ ]	The range of 1.71V to 1.91 V (1.81 V $\pm$ 0.5 dB).
PME20	The absolute value of the peak of the waveforms at points A and B shall	32.6.1.2.2	M	Yes [ ]	Differ by less than 2%.
PME21	The absolute value of the peak of the waveform at points C and D as defined in Figure 32-18 shall differ	32.6.1.2.2	M	Yes [ ]	From 0.5 times the average of the absolute values of the peaks of the waveform at points A and B by less than 2%.
PME22	The preprocessing filter shall have	32.6.1.2.2	M	Yes [ ]	The transfer function specified in 32.6.1.2.2.
PME23	When in transmit test mode 1 and observing the differential transmitted output at the MDI, for either pair, with no intervening cabling, the peak value of the waveform at point F as defined in Figure 32-17 shall be	32.6.1.2.3	M	Yes [ ]	Greater than 70.5% of the peak value of the waveform at point E. A preprocessing filter is not used for this measurement.

**32.13.5.8 PMA electrical specifications (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
PME24	The transmitter differential output voltage shall be measured at the output of the high pass filter defined in 32.6.1.2.2 with no intervening cables.	32.6.1.2.4	M	Yes [ ]	
PME25	The voltage waveforms at points A, B, C and D as defined in Figure 32–17, when normalized by their respective peak values, shall	32.6.1.2.4	M	Yes [ ]	Lie within the time domain template defined in Figure 32–18 and the piecewise linear interpolation between the points in Table 32–6
PME26	The magnitude in dB of the Fourier transform of the waveforms at points A, B, C and D shall	32.6.1.2.4	M	Yes [ ]	Lie within the transmit frequency domain template defined in Figure 32–18 and the piecewise linear interpolation between the points in Table 32–7.
PME27	The time span of the waveforms so processed shall be	32.6.1.2.4	M	Yes [ ]	–80 ns to +2000 ns with the 0 ns point of the waveform aligned as for the time domain mask shown in Figure 32–18 and the magnitude of the Fourier transform should be normalized so that the maximum value is at 0 dB.
PME28	When in transmit mode 2, the peak-to-peak jitter of the zero crossings of the differential signal output at the MDI shall	32.6.1.2.5	M	Yes [ ]	Be less than 0.5 ns.
PME29	The quinary symbol transmission rate on each pair shall be	32.6.1.2.6	M	Yes [ ]	25.000 MHz $\pm$ 0.01%.
PME30	The PMA shall provide	32.6.1.3	M	Yes [ ]	The Receive function specified in 32.4.1.3 in accordance with the electrical specifications of this clause.
PME31	The patch cabling and interconnecting hardware used in test configurations shall	32.6.1.3	M	Yes [ ]	Meet or exceed ISO/IEC 11801 Category 3 specifications.
PME32	The combined responses of the test fixture TX block and NEXT or cabling channel blocks shall be	32.6.1.3.1	M	Yes [ ]	Those defined in Table 32–8.
PME33	The output impedance of the test channel shall be	32.6.1.3	M	Yes [ ]	Consistent with 32.6.1.4.1.
PME34	The idle symbol generator outputs shall be	32.6.1.3	M	Yes [ ]	Conformant with the idle signaling specified in 32.3 with loc_rcvr_status=OK.
PME35	The clock source shall	32.6.1.3	M	Yes [ ]	Result in a quinary symbol transmission rate conformant with 32.6.1.2.6.
PME36	The jitter on the clock source shall be	32.6.1.3	M	Yes [ ]	Less than 0.2 ns.

**32.13.5.8 PMA electrical specifications (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
PME37	The test channel implementation shall ensure that the ratio of the squared error between the implemented NEXT channel symbol responses and the specified NEXT channel symbol responses to the energy in the specified NEXT channel symbol responses shall be	32.6.1.3	M	Yes [ ]	Less than 5%.
PME38	The test channel implementation shall ensure that the energy of the implemented NEXT channel impulse responses and the energy of the specified NEXT channel impulse responses shall	32.6.1.3	M	Yes [ ]	Differ by less than $\pm 0.25$ dB.
PME39	A special receiver test mode shall be required to allow for receiver alien NEXT tolerance and jitter testing.	32.6.1.3.2	M	Yes [ ]	
PME40	For a PHY with a MII interface, this mode shall be enabled by	32.6.1.3.2	M	Yes [ ]	Setting bit 9.13 (MASTER-SLAVE Control Register) of the MII management register set to a 1.
PME41	A PHY without an MII shall provide	32.6.1.3.2	M	Yes [ ]	A means to enable this test mode.
PME42	This mode shall not be overridden except by	32.6.1.3.2	M	Yes [ ]	Clearing bit 9.13 or resetting the PHY.
PME43	When the receive test mode is enabled, the receiver shall	32.6.1.3.2	M	Yes [ ]	Configure itself in SLAVE mode, continually attempt to bring its receiver up until successful receiver operation is achieved and transmit symbols in idle mode.
PME44	For a PHY with a MII interface, when the receiver is properly detecting the received data, it shall set	32.6.1.3.2	M	Yes [ ]	Bit 10.13 of the MII management register set to 1 and reset the error count in bit 10.0–10.7 (MSB) to zero.
PME45	The error count shall be incremented	32.6.1.3.2	M	Yes [ ]	For every symbol error detected in the received idle sequence.
PME46	Upon loss of proper data reception, the receiver shall	32.6.1.3.2	M	Yes [ ]	Clear bit 10.13.
PME47	A PHY without an MII shall provide	32.6.1.3.2	M	Yes [ ]	A means to provide the functions defined in PME43 through PME46.
PME48	The vendor shall provide a means to enable this mode for conformance testing.	32.6.1.3.2	M	Yes [ ]	

**32.13.5.8 PMA electrical specifications (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
PME49	Differential signals received on the receive inputs that were transmitted within the constraints of 32.6.1.2, and have then passed through a link as defined in 32.7, shall be translated into	32.6.1.3.3	M	Yes [ ]	One of the PMA_UNITDATA.indicate messages with a bit error rate less than $10^{-10}$ and sent to the PCS after link bring-up.
PME50	Performance shall be tested	32.6.1.3.3	M	Yes [ ]	In at least two configurations: using a 100 m link segment conformant to 32.7 and with a link segment less than one meter in length between transmitter and receiver.
PME51	Differential signals received from the test channel defined in 32.6.1.3.1 shall be detected	32.6.1.3.4	M	Yes [ ]	With a symbol error rate less than $10^{-10}$ when the PHY is in receiver test mode for the combinations of channel and worst-case alien NEXT responses specified in 32.6.1.2.
PME52	In the test configuration described in 32.6.1.3.1 and for all combinations of worst-case channel and alien NEXT coefficients tabulated in 32.6.1.3.4, the peak-to-peak value of the RX_CLK zero-crossing jitter shall be less than	32.6.1.3.5	M	Yes [ ]	1.3 ns after the receiver is properly receiving the data and has set bit 9.13 of the MII management register set to 1.
PME53	When the jitter waveform is filtered by a high pass filter having the transfer function specified in 32.6.1.3.4, the peak-to-peak value of the jitter waveform shall be less than	32.6.1.3.5	M	Yes [ ]	0.8 ns.
PME54	The RX_CLK of the MII shall be made available for the receiver jitter test specified in 32.6.1.3.5.	32.6.1.3.5	M	Yes [ ]	
PME55	A PHY without an MII shall provide an equivalent to the MII RX-CLK clock for the receiver jitter test specified in 32.6.1.3.5.	32.6.1.3.5	M	Yes [ ]	
PME56	While receiving packets from a compliant 100BASE-T2 transmitter connected to all MDI pins, a receiver shall send the	32.6.1.3.6	M	Yes [ ]	Proper PMA_UNITDATA.indicate messages to the PCS for any differential input signal $E_s$ that results in a signal $E_{dif}$ that meets 32.6.1.3.3 even in the presence of common mode voltages $E_{cm}$ (applied as shown in Figure 32-21).

## 32.13.5.8 PMA electrical specifications (continued)

Item	Feature	Subclause	Status	Support	Value/Comment
PME57	$E_{cm}$ shall be	32.6.1.3.6	M	Yes [ ]	A 25 V peak-to-peak square wave, 500 kHz or lower in frequency, with edges no slower than 4 ns (20%–80%), connected to each of the pairs BI_DA+, BI_DA-, BI_DB+ and BI_DB-.
PME58	The receive feature shall properly receive	32.6.1.3.7	M	Yes [ ]	Incoming data with a 5-level symbol rate within the range 25.000 MHz $\pm$ 0.01%.
PME59	The differential impedance as measured at the MDI for each transmit/receive channel shall be such that	32.6.1.4.1	M	Yes [ ]	Any reflection due to differential signals incident upon the MDI from a balanced cabling having an impedance of 100 $\Omega$ is at least 17 dB below the incident signal, over the frequency range of 2.0 MHz to 25.0 MHz and at least 12.9–20log <sub>10</sub> (f/10) dB over the frequency range 6.5 MHz to 25 MHz (f in MHz).
PME60	This return loss shall be maintained	32.6.1.4.1	M	Yes [ ]	At all times when the PHY is transmitting data.
PME61	The common-mode to differential-mode impedance balance of each transmit output shall exceed	32.6.1.4.2	M	Yes [ ]	The value specified by the equations specified in 32.6.1.2.6 over the range 2.0–25.0 MHz.
PME62	Transmitters and receivers shall tolerate	32.6.1.4.4	M	Yes [ ]	The application of short circuits between the leads of any receive input for an indefinite period of time without damage.
PME63	Transmitters and receivers shall resume	32.6.1.4.4	M	Yes [ ]	Normal operation after such faults are removed.
PME64	The magnitude of the current through the short circuit specified in PME62 shall not exceed	32.6.1.4.4	M	Yes [ ]	300 mA.
PME65	Transmitters shall withstand without damage	32.6.1.4.4	M	Yes [ ]	A 1000 V common mode impulse of either polarity ( $E_{cm}$ as indicated in Figure 32–24).
PME66	The shape of the impulse shall be	32.6.1.4.4	M	Yes [ ]	0.3/50 $\mu$ s (300 ns virtual front time, 50 $\mu$ s virtual time of half value), as defined in IEC 60060.
PME67	After 100 ms following PowerOn, the current drawn by the PHY shall not exceed	32.6.2	M	Yes [ ]	1.0 A when powered through the MII.



**32.13.5.8 PMA electrical specifications (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
PME68	The PHY shall	32.6.2	M	Yes [ ]	Be capable of operating from all voltage sources allowed by Clause 22, including those current limited to 1.0 A, as supplied by the DTE or repeater through the resistance of all permissible MII cabling.
PME69	The PHY shall not introduce	32.6.2	M	Yes [ ]	Extraneous signals on the MII control circuits during normal power-up and power-down.

## 32.13.5.9 Characteristics of the link segment

Item	Feature	Subclause	Status	Support	Value/Comment
LKS1	100BASE-T2 links shall use	32.7.1	M	Yes [ ]	2 pair of balanced cabling, CAT 3 or better, with a nominal impedance of 100 $\Omega$ .
LKS2	Unless otherwise specified, link segment testing shall be conducted using	32.7.2	M	Yes [ ]	Source and load impedances of 100 $\Omega$ .
LKS3	The tolerance on the poles of the test filter used in this section shall be	32.7.2	M	Yes [ ]	$\pm 1\%$ .
LKS4	The insertion loss of a link segment shall be no more than	32.7.2.1	M	Yes [ ]	14.6 dB at all frequencies between 2 and 16 MHz.
LKS5	The insertion loss specification shall be met when	32.7.2.1	M	Yes [ ]	The link segment is terminated in source and load impedances that satisfy 32.6.1.4.1.
LKS6	The magnitude of the differential characteristic impedance of a 3 m segment of balanced cabling pair used in a link shall be	32.7.2.2	M	Yes [ ]	Between 85 and 115 $\Omega$ for all frequencies between 2 and 16 MHz.
LKS7	The NEXT loss between each of the two duplex channels of a link segment shall be	32.7.2.3.1	M	Yes [ ]	At least $19.3 - 16.6 \log_{10}(f/16)$ (where $f$ is the frequency in MHz) over the frequency range 2.0 to 16 MHz.
LKS8	The NEXT loss between link segments of two different connections shall be	32.7.2.3.1	M	Yes [ ]	At least $22.0 - 16.6 \log_{10}(f/16)$ (where $f$ is the frequency in MHz) over the frequency range 2.0 to 16 MHz.
LKS9	The MDNEXT loss between a link segment and the two alien data carrying channels shall be	32.7.2.3.2	M	Yes [ ]	At least $19.0 - 16.6 \log_{10}(f/16)$ dB (where $f$ is the frequency in MHz) over the frequency range 2.0 to 16 MHz.
LKS10	To limit the FEXT noise from an adjacent channel, the ELFEXT loss between channels shall be	32.7.2.3.3	M	Yes [ ]	Greater than $20.9 - 20 \log_{10}(f/16)$ dB as defined in 32.7.2.3.3
LKS11	The MDELNEXT loss between a duplex channel and the other data carrying duplex channels shall be	32.7.2.3.4	M	Yes [ ]	Greater than $19.9 - 20 \log_{10}(f/16)$ (where $f$ is the frequency in MHz) over the frequency range 2.0 to 16 MHz.
LKS12	The maximum spacing of the frequency in the sample shall be	32.7.2.3.5	M	Yes [ ]	250 kHz.

**32.13.5.9 Characteristics of the link segment (continued)**

Item	Feature	Subclause	Status	Support	Value/Comment
LKS13	When 10BASE-T service is used in adjacent pairs, the channel shall provide	32.7.2.3.5	M	Yes [ ]	A NEXT loss to Insertion loss Ratio (NIR) greater than 19.4 dB.
LKS14	The propagation delay of a link segment shall not exceed	32.7.2.4.1	M	Yes [ ]	5.7 ns/meter at all frequencies between 2.0–25.0 MHz.
LKS15	The difference in propagation delay, or skew, under all conditions, between the fastest and the slowest channel in a link segment shall not exceed	32.7.2.4.2	M	Yes [ ]	50 ns at all frequencies between 2.0–25.0 MHz.
LKS16	Once installed, the skew between pairs due to environmental conditions shall not vary	32.7.2.4.2	M	Yes [ ]	More than $\pm 10$ ns.
LKS17	The noise level on the link segments shall be such that	32.7.3	M	Yes [ ]	The objective error rate is met.
LKS18	The MDNEXT noise on a link segment shall not exceed	32.7.3.1	M	Yes [ ]	182 mVp.
LKS19	The MDFEXT noise on a link segment shall not exceed	32.7.3.2	M	Yes [ ]	54.4 mVp.
LKS20	Cables made from up to 25 pairs of Category 5 cabling, if used, shall be limited in length to no more than	32.7.4.3	O	Yes [ ]	90 m total.
LKS21	The services in 25 pair Category 5 cabling shall be limited to any combination of 100BASE-T2, 10BASE-T and digital phone service compliant with the ISDN-BR U and S/T interfaces up to a total of 12 services in the cable.	32.7.4.3	M	Yes [ ]	
LKS22	Jumper cabling, or horizontal runs, made from more than 4 pairs of Category 3 cabling shall not be used	32.7.4.3	M	Yes [ ]	

**32.13.5.10 MDI requirements**

Item	Feature	Subclause	Status	Support	Value/Comment
MDI1	MDI connector	32.8.1	M	Yes [ ]	8-Way connector as per IEC 60603-7.
MDI2	Connector used on cabling	32.8.1	M	Yes [ ]	Plug.
MDI3	Connector used on PHY	32.8.1	M	Yes [ ]	Jack (as opposed to plug).
MDI4	AN MDI connector for the PHY that implements the crossover function shall be marked	32.8.2	M	Yes [ ]	With the graphical symbol "X".

**32.13.5.11 General safety and environmental requirements**

Item	Feature	Subclause	Status	Support	Value/Comment
ENV1	Conformance to safety specifications	32.10.1	M	Yes [ ]	IEC 60950.
ENV2	Installation practice	32.10.2.1	M	Yes [ ]	Sound practice, as defined by applicable local codes.
ENV3	Any safety grounding path for an externally connected PHY shall be provided through the circuit ground of the MII connection.	32.10.2.2	M	Yes [ ]	
ENV4	Care taken during installation to ensure that non-insulated network cabling conductors do not make electrical contact with unintended conductors or ground.	32.10.2.3	M	Yes [ ]	
ENV5	Application of voltages specified in 32.10.2.4 does not result in any safety hazard.	32.10.2.4	M	Yes [ ]	
ENV6	Conformance with local and national codes for the limitation of electromagnetic interference.	32.10.3.1	M	Yes [ ]	
ENV7	All equipment subject to this clause shall conform to	32.10.4	M	Yes [ ]	The requirements of 14.7 and the applicable sections of ISO/IEC 11801.

**32.13.5.12 Timing requirements exposed MII**

Item	Feature	Subclause	Status	Support	Value/Comment
TME1	TX-EN (sampled to MDI output)	32.12.1	M	Yes [ ]	7–9 bit times
TME2	MDI input to CRS assert	32.12.1	M	Yes [ ]	25 bit times
TME3	MDI input to CRS de-assert	32.12.1	M	Yes [ ]	29 bit times
TME4	MDI input to COL assert	32.12.1	M	Yes [ ]	25 bit times
TME5	MDI input to COL de-assert	32.12.1	M	Yes [ ]	29 bit times
TME6	TX_EN sampled to CRS assert	32.12.1	M	Yes [ ]	0–4 bit times
TME7	TX_EN sampled to CRS de-assert	32.12.1	M	Yes [ ]	0–16 bit times

**32.13.5.13 Timing requirements unexposed MII**

Item	Feature	Subclause	Status	Support	Value/Comment
TMU1	MAC transmit start to MDI output	32.12.2	M	Yes [ ]	13 bit times
TMU2	MDI input to MDI output (worst-case non-deferred transmit)	32.12.2	M	Yes [ ]	50 bit times
TMU3	MDI input to collision detect	32.12.2	M	Yes [ ]	33 bit times
TMU4	MDI input to MDI output = Jam (worst-case collision response)	32.12.2	M	Yes [ ]	50 bit times

**32.13.5.14 Timing requirements: carrier assertion/deassertion constraint**

Item	Feature	Subclause	Status	Support	Value/Comment
TMC1	Each DTE shall satisfy the relationship	36.5.3	M	Yes [ ]	(MAX MDI to MAC Carrier De-assert Detect)–(MIN MDI to MAC Carrier Assert Detect).

**33. Clause 33 is reserved for future use.**

NOTE—The remaining annexes in the standard are numbered in correspondence to their associated clauses; e.g., Annexes 22A, 22B, and 22C correspond to Clause 22.

## Annex 22A

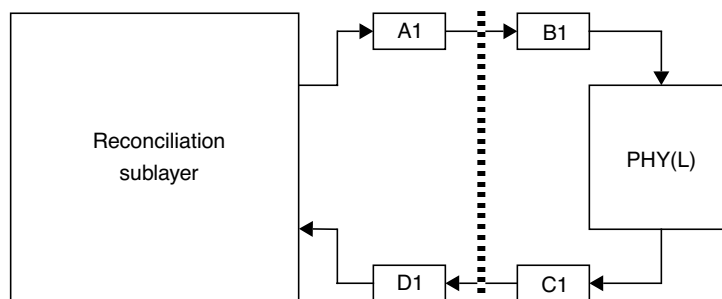
(informative)

### MII output delay, setup, and hold time budget

#### 22A.1 System model

The discussion of signal timing characteristics that follows will refer to the system model depicted in Figure 22A-1, Figure 22A-2, and Figure 22A-3. This system model can be applied to each of the three application environments defined in 22.2.1.

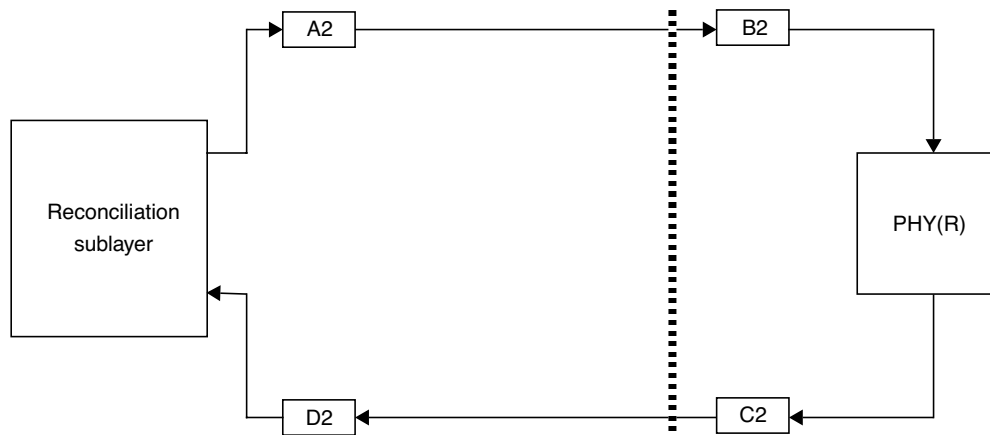
Figure 22A-1 depicts a simple system model in which the MII is used to interconnect two integrated circuits on the same circuit assembly. In this model the Reconciliation sublayer comprises one integrated circuit, and the PHY comprises the other. A Reconciliation sublayer or a PHY may actually be composed of several separate integrated circuits. The system model in Figure 22A-1 includes two unidirectional signal transmission paths, one from the Reconciliation sublayer to the PHY and one from the PHY to the Reconciliation sublayer. The path from the Reconciliation sublayer to the PHY is separated into two sections, labeled A1 and B1. The path from the PHY to the Reconciliation sublayer is separated into two sections, labeled C1 and D1.



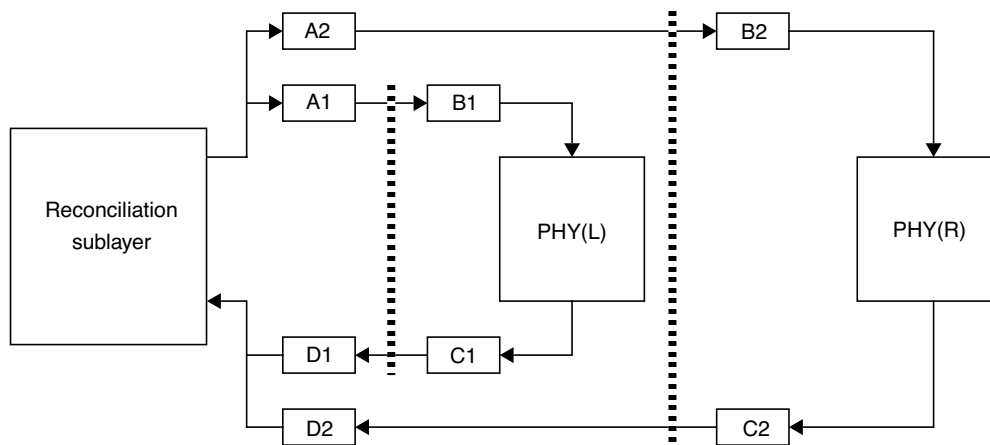
**Figure 22A-1—Model for integrated circuit to integrated circuit connection**

Figure 22A-2 depicts a system model for the case where the MII is used to interconnect two circuit assemblies. The circuit assemblies may be physically connected in a motherboard/daughterboard arrangement, or they may be physically connected with the cable defined in 22.4.5 and the line interface connector defined in 22.6. The system model in Figure 22A-2 includes two unidirectional signal transmission paths, one from the Reconciliation sublayer to the PHY and one from the PHY to the Reconciliation sublayer. The path from the Reconciliation sublayer to the PHY is separated into two sections, labeled A2 and B2. The path from the PHY to the Reconciliation sublayer is separated into two sections, labeled C2 and D2.

Figure 22A-3 depicts a system model in which the MII is used to interconnect both integrated circuits and circuit assemblies. This system model allows for separate signal transmission paths to exist between the Reconciliation sublayer and a local PHY(L), and between the Reconciliation sublayer and a remote PHY(R). The unidirectional paths between the Reconciliation sublayer and the PHY(L) are composed of sections A1, B1, C1, and D1. The unidirectional paths between the Reconciliation sublayer and the remote PHY(R) are composed of sections A2, B2, C2, and D2.



**Figure 22A-2—Model for circuit assembly to circuit assembly connection**



**Figure 22A-3—Combined model**

Each of these system models assumes a set of common timing and electrical characteristics that shall be met at the input and output ports of the Reconciliation sublayer and PHY devices. The characteristics of the signal transmission paths are identified for each of the sections A1, B1, C1, D1, A2, B2, C2, and D2.

## 22A.2 Signal transmission path characteristics

The signal transmission path characteristics are specified for each of the path sections defined in 22A.1. The characteristics for these sections are specified so as to allow sections A1, B1, C1, and D1 to be implemented in the form of printed circuit board traces, while sections A2, B2, C2, and D2 may be implemented with a combination of printed circuit board traces and wire conductors in a cable assembly.

The signal transmission path characteristics are stated in terms of their maximum delay and their characteristic impedance. These values are summarized in Table 22A-1.



**Table 22A-1—Signal transmission path characteristics**

Section	Maximum delay (ns)	Impedance ( $\Omega$ )
A1, D1	5	$68 \pm 15\%$
B1, C1	2.5	$68 \pm 15\%$
A2, D2	5	$68 \pm 10\%$
B2, C2	2.5	$68 \pm 10\%$

The driver characteristics specified in 22.4.3, the receiver characteristics specified in 22.4.4, and the signal transmission path characteristics specified in Table 22A-1 can be applied to the system models shown in Figure 22A-1 or Figure 22A-2. The combination of loads presented in Figure 22A-3 cannot be adequately driven by an output buffer that meets the driver characteristics specified in 22.4.3 while being sampled by an input buffer that meets the receiver characteristics specified in 22.4.4.

To address the system model depicted in Figure 22A-3, it is permissible to incorporate an additional stage of buffering into path sections A1, A2, D1, and D2, provided that the resulting maximum delay characteristic for those path sections does not exceed the value stated in Table 22A-1. The delay characteristic for transmission path sections A2 and D2 includes an allowance for the delay that results from the presence of a lumped capacitive load at the end of the path. For a transmission path section with a characteristic impedance  $Z_o$ , with a lumped capacitive load  $C_L$ , this delay is nominally  $Z_o C_L$ . In the case of a maximum transmission path section impedance of  $78 \Omega$  with a lumped load of 8 pF, the nominal delay is 0.6 ns. Thus the allowable delay for a buffer inserted into transmission path section A2 or D2 is 4.4 ns.

### 22A.3 Budget calculation

A recommended timing budget is shown in Table 22A-2. This budget assumes that the combined system model shown in Figure 22A-3 represents a worst case.

**Table 22A-2—Round-trip delay budget**

Description	Incremental delay (ns)	Cumulative delay (ns)
TX_CLK output at PHY(R)	0.0	0.0
Transmission path section C2	2.5	2.5
Transmission path section D2	5.0	7.5
clock to output in Reconciliation sublayer	15.0	22.5
Transmission path section A2	5.0	27.5
Transmission path section B2	2.5	30.0
Setup time at PHY(R)	10.0	40.0

## Annex 22B

(informative)

### MII driver ac characteristics

#### 22B.1 Implications of CMOS ASIC processes

For MII drivers that drive rail to rail, such as those commonly used in CMOS ASICs (complimentary metal oxide semiconductor application-specific integrated circuits), the ac characteristic performance requirements of 22.4.3.2 can be met if the  $V_{oh}$  vs.  $I_{oh}$  and  $V_{ol}$  vs.  $I_{ol}$  dc characteristics of the driver stay within the unshaded areas of Figure 22B-1.

The variation in output resistance of a field effect transistor (FET) due to variations in supply voltage, temperature, and process may require that a resistance be placed in series with the output of the FETs to meet this specification. The series resistance can be part of the driver circuit, or external to the driver. If the series resistance is not part of the driver circuit, the driver vendor shall specify the value of series resistance required to meet the specification. A series resistor used to meet this specification is conceptually part of the driver regardless of whether it is physically internal or external to the driver.

The propagation delay of the path between the driver and an external series resistor used to meet the specification shall not exceed 10% of the 10–90% rise/fall time of the driver.

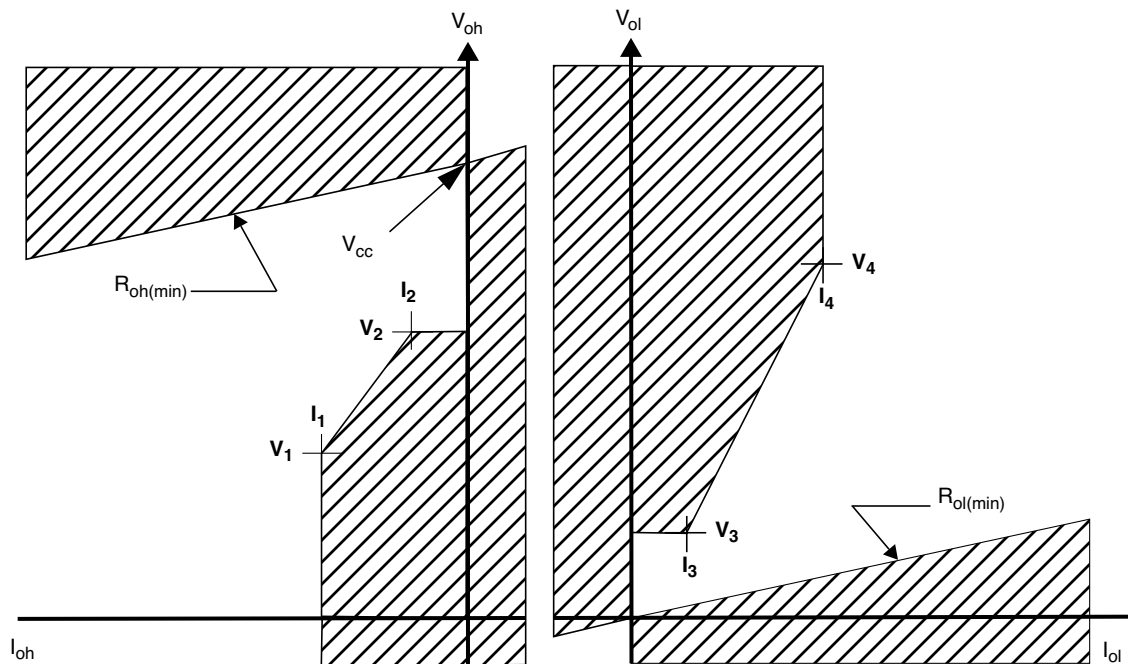


Figure 22B-1 – Driver output V-I curve

### 22B.2 $R_{o(min)}$ and V, I values for operation from $5\text{ V} \pm 10\%$ supply

Referring to Figure 22B-1,  $R_{oh(min)}$  and  $R_{ol(min)}$  both equal  $40\ \Omega$ , and the values for the V-I points on the curve are given in Table 22B-1 for MII drivers that drive rail to rail from a  $+5\text{ V} \pm 10\%$  power supply.

**Table 22B-1 – Values for driver output V-I curve (5 V supply)**

V-I point	I (mA)	V (V)
$I_1, V_1$	-20	1.10
$I_2, V_2$	-4	2.4
$I_3, V_3$	4	0.40
$I_4, V_4$	43	3.05

### 22B.3 $R_{o(min)}$ and V, I values for operation from $3.3 \pm 0.3\text{ V}$ supply

Referring to Figure 22B-1,  $R_{oh(min)}$  and  $R_{ol(min)}$  both equal  $33\ \Omega$ , and the values for the V-I points on the curve are given in Table 22B-2 for MII drivers that drive rail to rail from a  $+3.3 \pm 0.3\text{ V}$  power supply.

**Table 22B-2 – Values for driver output V-I curve (3.3 V supply)**

V-I point	I (mA)	V (V)
$I_1, V_1$	-20	1.10
$I_2, V_2$	-4	2.4
$I_3, V_3$	4	0.40
$I_4, V_4$	26	2.10

## Annex 22C

(informative)

### Measurement techniques for MII signal timing characteristics

#### 22C.1 Measuring timing characteristics of source terminated signals

The measurement of timing relationships between MII signals at the MII connector is complicated by the use of driver output impedance to control transmission line reflections on point-to-point transmission paths passing through the connector. The voltage waveforms on point-to-point transmission paths can be different at the MII connector and at the end of the paths. A clean transition (or step) from one logic state to the other at the end of a point to point path can appear as two half-steps at the MII connector.

To eliminate ambiguity as to where on a two half-step state transition to measure timing, all timing measurements on point-to-point transmission paths will be at the end of the path. In some cases, an end of path must be artificially created.

#### 22C.2 Measuring timing characteristics of transmit signals at the MII

The timing of TX\_EN, TX\_ER, and TXD<3:0> relative to TX\_CLK at the MII connector is measured as follows.

Use the time base for TX\_CLK as a timing reference. Break the TX\_CLK path at the MII connector, forcing the TX\_CLK point-to-point transmission path to end at the connector. Measure when the rising edge of TX\_CLK passes through  $V_{ih(min)}$  at the MII connector. Call this time  $T_{clk}$ . Reconnect the TX\_CLK path at the MII connector and break the paths of TX\_EN, TX\_ER, and TXD<3:0> at the MII connector, forcing the paths to end at the connector. Measure when TX\_EN, TX\_ER, and TXD<3:0> exit the switching region at the MII connector. Call these times  $T_{en}$ ,  $T_{er}$ , and  $T_{<3:0>}$ , respectively.

The timing relationships at the MII connector for TX\_EN, TX\_ER, and TXD<3:0> relative to TX\_CLK are met if  $(T_{en} - T_{clk})$ ,  $(T_{er} - T_{clk})$ ,  $(T_3 - T_{clk})$ ,  $(T_2 - T_{clk})$ ,  $(T_1 - T_{clk})$ , and  $(T_0 - T_{clk})$ , respectively, meet the timing relationships specified in 22.3.1.

#### 22C.3 Measuring timing characteristics of receive signals at the MII

The timing of RX\_DV, RX\_ER, and RXD<3:0> relative to RX\_CLK at the MII connector is measured as follows.

Break the paths of RX\_CLK, RX\_DV, RX\_ER, and RXD<3:0> at the MII connector, forcing the paths to end at the connector. Measure when RX\_DV, RX\_ER, and RXD<3:0> exit the switching region at the MII connector relative to when the rising edge of RX\_CLK passes through  $V_{il(max)}$ . Also measure when RX\_DV, RX\_ER, and RXD<3:0> reenter the switching region relative to when the rising edge of RX\_CLK passes through  $V_{ih(min)}$ .

The timing relationships at the MII connector for RX\_DV, RX\_ER, and RXD<3:0> relative to RX\_CLK are met if the times measured in the previous step meet the timing relationships specified in 22.3.2.

## 22C.4 Measuring timing characteristics of MDIO

The MDIO and MDC signal timing characteristics cannot be measured using the techniques defined for the transmit and receive signals since MDIO and MDC may connect a single station management entity to multiple PHY devices. The MDIO and MDC timing characteristics are measured with a PHY connected to the MII connector. The signal timing characteristics for MDC and MDIO must be met over the range of conditions which occur when from one to 32 PHYs are connected to an STA. When 32 PHYs are connected to an STA, the total capacitance can be as large as 390 pF on MDC, and as large as 470 pF on MDIO.

## Annex 23A

(normative)

### 6T code words

The *leftmost* ternary symbol of each 6T Code group shown in Table 23A-1 (broken into 23A-1a and 23A-1b for pagination) shall be transmitted *first*. The *leftmost* nibble of each data octet is the *most significant*.

**Table 23A-1a—100BASE-T4 8B6T code table**

Data octet	6T code group	Data octet	6T code group	Data octet	6T code group	Data octet	6T code group
00	+ - 0 0 + -	20	0 0 - + + -	40	+ 0 + 0 0 -	60	0 - 0 + + 0
01	0 + - + - 0	21	- - + 0 0 +	41	+ + 0 0 - 0	61	0 0 - + 0 +
02	+ - 0 + - 0	22	+ + - 0 + -	42	+ 0 + 0 - 0	62	0 - 0 + 0 +
03	- 0 + + - 0	23	+ + - 0 - +	43	0 + + 0 - 0	63	- 0 0 + 0 +
04	- 0 + 0 + -	24	0 0 + 0 - +	44	0 + + 0 0 -	64	- 0 0 + + 0
05	0 + - - 0 +	25	0 0 + 0 + -	45	+ + 0 - 0 0	65	0 0 - 0 + +
06	+ - 0 - 0 +	26	0 0 - 0 0 +	46	+ 0 + - 0 0	66	0 - 0 0 + +
07	- 0 + - 0 +	27	- - + + + -	47	0 + + - 0 0	67	- 0 0 0 + +
08	- + 0 0 + -	28	- 0 - + + 0	48	0 0 0 + 0 0	68	- + - + + 0
09	0 - + + - 0	29	- - 0 + 0 +	49	0 0 0 - + +	69	- - + + 0 +
0A	- + 0 + - 0	2A	- 0 - + 0 +	4A	0 0 0 + - +	6A	- + - + 0 +
0B	+ 0 - + - 0	2B	0 - - + 0 +	4B	0 0 0 + + -	6B	+ - - + 0 +
0C	+ 0 - 0 + -	2C	0 - - + + 0	4C	0 0 0 - + 0	6C	+ - - + + 0
0D	0 - + - 0 +	2D	- - 0 0 + +	4D	0 0 0 - 0 +	6D	- - + 0 + +
0E	- + 0 - 0 +	2E	- 0 - 0 + +	4E	0 0 0 + - 0	6E	- + - 0 + +
0F	+ 0 - - 0 +	2F	0 - - 0 + +	4F	0 0 0 + 0 -	6F	+ - - 0 + +
10	+ 0 + - - 0	30	+ - 0 0 - +	50	+ 0 + - - +	70	- + + 0 0 0
11	+ + 0 - 0 -	31	0 + - - + 0	51	+ + 0 - + -	71	+ - + 0 0 0
12	+ 0 + - 0 -	32	+ - 0 - + 0	52	+ 0 + - + -	72	+ + - 0 0 0
13	0 + + - 0 -	33	- 0 + - + 0	53	0 + + - + -	73	0 0 + 0 0 0
14	0 + + - - 0	34	- 0 + 0 - +	54	0 + + - - +	74	- 0 + 0 0 0
15	+ + 0 0 - -	35	0 + - + 0 -	55	+ + 0 + - -	75	0 - + 0 0 0
16	+ 0 + 0 - -	36	+ - 0 + 0 -	56	+ 0 + + - -	76	+ 0 - 0 0 0
17	0 + + 0 - -	37	- 0 + + 0 -	57	0 + + + - -	77	0 + - 0 0 0
18	0 + - 0 + -	38	- + 0 0 - +	58	+ + + 0 - -	78	0 - - + + +
19	0 + - 0 - +	39	0 - + - + 0	59	+ + + - 0 -	79	- 0 - + + +
1A	0 + - + + -	3A	- + 0 - + 0	5A	+ + + - - 0	7A	- - 0 + + +
1B	0 + - 0 0 +	3B	+ 0 - - + 0	5B	+ + 0 - - 0	7B	- - 0 + + 0
1C	0 - + 0 0 +	3C	+ 0 - 0 - +	5C	+ + 0 - - +	7C	+ + - 0 0 -
1D	0 - + + + -	3D	0 - + + 0 -	5D	+ + 0 0 0 -	7D	0 0 + 0 0 -
1E	0 - + 0 - +	3E	- + 0 + 0 -	5E	- - + + + 0	7E	+ + - - - +
1F	0 - + 0 + -	3F	+ 0 - + 0 -	5F	0 0 - + + 0	7F	0 0 + - - +

**Table 23A-1b—100BASE-T4 8B6T code table**

Data octet	6T code group	Data octet	6T code group	Data octet	6T code group	Data octet	6T code group
80	+ - + 0 0 -	A0	0 - 0 + + -	C0	+ - + 0 + -	E0	+ - 0 + + -
81	+ + - 0 - 0	A1	0 0 - + - +	C1	+ + - + - 0	E1	0 + - + - +
82	+ - + 0 - 0	A2	0 - 0 + - +	C2	+ - + + - 0	E2	+ - 0 + - +
83	- + + 0 - 0	A3	- 0 0 + - +	C3	- + + + - 0	E3	- 0 + + - +
84	- + + 0 0 -	A4	- 0 0 + + -	C4	- + + 0 + -	E4	- 0 + + + -
85	+ + - - 0 0	A5	0 0 - - + +	C5	+ + - - 0 +	E5	0 + - - + +
86	+ - + - 0 0	A6	0 - 0 - + +	C6	+ - + - 0 +	E6	+ - 0 - + +
87	- + + - 0 0	A7	- 0 0 - + +	C7	- + + - 0 +	E7	- 0 + - + +
88	0 + 0 0 0 -	A8	- + - + + -	C8	0 + 0 0 + -	E8	- + 0 + + -
89	0 0 + 0 - 0	A9	- - + + - +	C9	0 0 + + - 0	E9	0 - + + - +
8A	0 + 0 0 - 0	AA	- + - + - +	CA	0 + 0 + - 0	EA	- + 0 + - +
8B	+ 0 0 0 - 0	AB	+ - - + - +	CB	+ 0 0 + - 0	EB	+ 0 - + - +
8C	+ 0 0 0 0 -	AC	+ - - + + -	CC	+ 0 0 0 + -	EC	+ 0 - + + -
8D	0 0 + - 0 0	AD	- - + - + +	CD	0 0 + - 0 +	ED	0 - + - + +
8E	0 + 0 - 0 0	AE	- + - - + +	CE	0 + 0 - 0 +	EE	- + 0 - + +
8F	+ 0 0 - 0 0	AF	+ - - - + +	CF	+ 0 0 - 0 +	EF	+ 0 - - + +
90	+ - + - - +	B0	0 - 0 0 0 +	D0	+ - + 0 - +	F0	+ - 0 0 0 +
91	+ + - - + -	B1	0 0 - 0 + 0	D1	+ + - - + 0	F1	0 + - 0 + 0
92	+ - + - + -	B2	0 - 0 0 + 0	D2	+ - + - + 0	F2	+ - 0 0 + 0
93	- + + - + -	B3	- 0 0 0 + 0	D3	- + + - + 0	F3	- 0 + 0 + 0
94	- + + - - +	B4	- 0 0 0 0 +	D4	- + + 0 - +	F4	- 0 + 0 0 +
95	+ + - + - -	B5	0 0 - + 0 0	D5	+ + - + 0 -	F5	0 + - + 0 0
96	+ - + + - -	B6	0 - 0 + 0 0	D6	+ - + + 0 -	F6	+ - 0 + 0 0
97	- + + + - -	B7	- 0 0 + 0 0	D7	- + + + 0 -	F7	- 0 + + 0 0
98	0 + 0 - - +	B8	- + - 0 0 +	D8	0 + 0 0 - +	F8	- + 0 0 0 +
99	0 0 + - + -	B9	- - + 0 + 0	D9	0 0 + - + 0	F9	0 - + 0 + 0
9A	0 + 0 - + -	BA	- + - 0 + 0	DA	0 + 0 - + 0	FA	- + 0 0 + 0
9B	+ 0 0 - + -	BB	+ - - 0 + 0	DB	+ 0 0 - + 0	FB	+ 0 - 0 + 0
9C	+ 0 0 - - +	BC	+ - - 0 0 +	DC	+ 0 0 0 - +	FC	+ 0 - 0 0 +
9D	0 0 + + - -	BD	- - + + 0 0	DD	0 0 + + 0 -	FD	0 - + + 0 0
9E	0 + 0 + - -	BE	- + - + 0 0	DE	0 + 0 + 0 -	FE	- + 0 + 0 0
9F	+ 0 0 + - -	BF	+ - - + 0 0	DF	+ 0 0 + 0 -	FF	+ 0 - + 0 0

## Annex 23B

(informative)

### Noise budget

Worst-case values for noise effects in the 100BASE-T4 system are as shown in Tables 23B-1 and 23B-2.

**Table 23B-1—Carrier presence analysis**

Received signal peak amplitude (min.)	792 mVp
NEXT noise	325 mVp

**Table 23B-2—Far-end signal analysis**

Received signal peak amplitude (min.)	796 mVp
Baseline wander	14 mVp
ISI	80 mVp
Reflections	60 mVp
FEXT noise	87 mVp



## Annex 23C

(informative)

### Use of cabling systems with a nominal differential characteristic impedance of 120 $\Omega$

The 100BASE-T4 standard specifies only the use of 100  $\Omega$  link segments for conformance. Since ISO/IEC 11801: 1995 also recognizes 120  $\Omega$  cabling, this informative annex specifies the conditions for using cabling systems with a nominal characteristic impedance of 120  $\Omega$  by 100BASE-T4 conformant stations.

The use of cables with a characteristic impedance outside the range specified in 23.6 will generally increase the mismatching effects in the link components, inducing additional noise in the received signals.

In particular, the use of a homogeneous link segment having a characteristic impedance of 120  $\Omega \pm 15 \Omega$  over the frequency band 1 to 16 MHz may add up to 1.4% of additional noise to the signals at the input of the receivers (worst-case short-length link segment).

Therefore, in order to keep the overall noise (MDFEXT + reflections) at the same value as for a 100  $\Omega$  link segment when using a 120  $\Omega$  link segment, the minimum ELFEXT loss requirement for the cable must be increased by 2 dB (i.e., from 23 dB to 25 dB at 12.5 MHz, see 23.6.3.2). Accordingly, the MDFEXT noise requirement shall be decreased from 87 mV peak to 69 mV peak. In practice, this means that cables rated category 4 or higher, as specified in ISO/IEC 11801: 1995, are required when 120  $\Omega$  cables are used with 100BASE-T4 compliant PMDs.

NOTE 1—The use of 100  $\Omega$  cords at end points in conjunction with 120  $\Omega$  premises cabling may be tolerated provided that all the components of the link are of Category 5, as defined in ISO/IEC 11801: 1995.

NOTE 2—The use of 100  $\Omega$  cords at any intermediate cross-connect points on 120  $\Omega$  links as well as the use of 120  $\Omega$  cords in conjunction with 100  $\Omega$  premises cabling is not allowed since it would result in worst-case jitter greater than that allowed in this standard.

CAUTION—Users of this annex are further advised to check with the manufacturer of the particular 100BASE-T4 couplers they intend to use with a 120  $\Omega$  link to see whether those couplers can operate correctly on cables with  $Z_c$  as high as 120  $\Omega \pm 15 \Omega$ .

## Annex 27A

(normative)

### Repeater delay consistency requirements

Proper operation of the network requires that repeaters do not cause the Inter-Packet Gap (IPG) to disappear by propagating the end of any carrier event to different output ports with greatly different delay times. Maximum port-to-port delays have been assigned as absolute delays to meet requirements for detection of collision within a slot time and limiting the length of collision fragments to less than minimum frame size. To avoid specification of minimum input-to-output propagation time as absolute values that reduce implementation flexibility, these delays are instead implied by imposing a triangular delay inequality relationship.

Consider three ports {A, B, C}. Using the notation SOP(xy) to mean the start-of-packet delay for an input at port x to resulting output on port y, repeaters shall achieve this relationship for all groups of three ports within a repeater set:

$$\text{SOP(AC)} < \text{SOP(AB)} + \text{SOP(BC)}$$

Following a frame transmitted by node A that propagates to nodes B and C, this constraint ensures that node B cannot complete an IPG timer and initiate a transmission that arrives at node C before node C has also advanced its own IPG timer sufficiently that a pending frame can contend for access to the network.

There is a second delay consistency requirement, one that relates to jam propagation by repeaters. Using a notation similar to that above, SOJ(xy) stands for the start-of-jam propagation delay from port x to port y and EOJ(xy) for the end-of-jam delay between same two ports.

To ensure proper detection of collisions and avoid generation of fragments that exceed minimum frame size, maximum values have been imposed on SOJ and EOJ delays through repeaters. No specific minima have been specified as all delays less than the maxima meet the collision detection and fragment length criteria. To prevent the jam pattern from shrinking excessively as it propagates through repeaters, repeaters shall meet this relationship between all pairs of ports:

$$\text{EOJ(AB)} \geq \text{SOJ(AB)} - 4 \text{ bit times}$$

## Annex 28A

(normative)

### Selector Field definitions

The Selector Field, S[4:0] in the Link Code Word, shall be used to identify the type of message being sent by Auto-Negotiation. The following table identifies the types of messages that may be sent. As new messages are developed, this table will be updated accordingly.

The Selector Field uses a 5-bit binary encoding, which allows 32 messages to be defined. All unspecified combinations are reserved. Reserved combinations shall not be transmitted.

**Table 28A–1—Selector Field value mappings**

S4	S3	S2	S1	S0	Selector description
0	0	0	0	0	Reserved for future Auto-Negotiation development
0	0	0	0	1	IEEE Std 802.3 <sup>®</sup>
0	0	0	1	0	IEEE Std 802.9 <sup>®</sup> ISLAN-16T
0	0	0	1	1	IEEE Std 802.5 <sup>®</sup>
1	1	1	1	1	Reserved for future Auto-Negotiation development <sup>a</sup>

<sup>a</sup>For up-to-date information on the allocation of Auto-Negotiation Selector fields see <http://www.ieee802.org/3/selectors/selectors.html>

## Annex 28B

(normative)

### IEEE 802.3® Selector Base Page definition

This annex provides the Technology Ability Field bit assignments, Priority Resolution table, and Message Page transmission conventions relative to the IEEE 802.3 Selector Field value within the base page encoding.

As new IEEE 802.3® LAN technologies are developed, a reserved bit in the Technology Ability field may be assigned to each technology by the standards body.

The new technology will then be inserted into the Priority Resolution hierarchy and made a part of the Auto-Negotiation standard. The relative hierarchy of the existing technologies will not change, thus providing backward compatibility with existing Auto-Negotiation implementations.

It is important to note that the reserved bits are required to be transmitted as logic zeros. This guarantees that devices implemented using the current priority table will be forward compatible with future devices using an updated priority table.

#### 28B.1 Selector field value

The value of the IEEE 802.3 Selector Field is  $S[4:0] = 00001$ .

#### 28B.2 Technology Ability Field bit assignments

The Technology bit field consists of bits D5 through D12 (A0–A7, respectively) in the IEEE 802.3® Selector Base Page. Table 28B–1 summarizes the bit assignments.

Note that the order of the bits within the Technology Ability Field has no relationship to the relative priority of the technologies.

Setting Bit A5 or Bit A6 indicates that the DTE has implemented both the optional MAC control sublayer and the PAUSE function as specified in Clause 31 and Annex 31B. This capability is significant only when the link is configured for full duplex operation, regardless of data rate and medium. The encoding of Bits A5 and A6 is specified in Table 28B–2.

**Table 28B–1—Technology Ability Field bit assignments**

Bit	Technology	Minimum cabling requirement
A0	10BASE-T	Two-pair category 3
A1	10BASE-T full duplex	Two-pair category 3
A2	100BASE-TX	Two-pair category 5
A3	100BASE-TX full duplex	Two-pair category 5
A4	100BASE-T4	Four-pair category 3
A5	PAUSE operation for full duplex links	Not applicable
A6	Asymmetric PAUSE operation for full duplex Links	Not applicable
A7	Reserved for future technology	

**Table 28B-2—Pause encoding**

PAUSE (A5)	ASM_DIR (A6)	Capability
0	0	No PAUSE
0	1	Asymmetric PAUSE toward link partner
1	0	Symmetric PAUSE
1	1	Both Symmetric PAUSE and Asymmetric PAUSE toward local device

The PAUSE bit indicates that the device is capable of providing the symmetric PAUSE functions as defined in Annex 31B. The ASM\_DIR bit indicates that asymmetric PAUSE is supported. The value of the PAUSE bit when the ASM\_DIR bit is set indicates the direction the PAUSE frames are supported for flow across the link. Asymmetric PAUSE configuration results in independent enabling of the PAUSE receive and PAUSE transmit functions as defined by Annex 31B. See 28B.3 regarding PAUSE configuration resolution.

### 28B.3 Priority resolution

Since two devices may have multiple abilities in common, a prioritization scheme exists to ensure that the highest common denominator ability is chosen. The following list shall represent the relative priorities of the technologies supported by the IEEE 802.3 Selector Field value, where priorities are listed from highest to lowest.

- a) 1000BASE-T full duplex
- b) 1000BASE-T
- c) 100BASE-T2 full duplex
- d) 100BASE-TX full duplex
- e) 100BASE-T2
- f) 100BASE-T4
- g) 100BASE-TX
- h) 10BASE-T full duplex
- i) 10BASE-T

The rationale for this hierarchy is straightforward. 10BASE-T is the lowest common denominator and therefore has the lowest priority. Full duplex solutions are always higher in priority than their half duplex counterparts. 1000BASE-T has a higher priority than 100 Mb/s technologies. 100BASE-T2 is ahead of 100BASE-TX and 100BASE-T4 because 100BASE-T2 runs across a broader spectrum of copper cabling and can support a wider base of configurations. 100BASE-T4 is ahead of 100BASE-TX because 100BASE-T4 runs across a broader spectrum of copper cabling. The relative order of the technologies specified herein shall not be changed. As each new technology is added, it shall be inserted into its appropriate place in the list, shifting technologies of lesser priority lower in priority. If a vendor-specific technology is implemented, the priority of all IEEE 802.3<sup>®</sup> standard technologies shall be maintained, with the vendor specific technology inserted at any appropriate priority location.

The use of the PAUSE operation for full duplex links (as indicated by bits A5 and A6) is orthogonal to the negotiated data rate, medium, or link technology. The setting of these bits indicates the availability of additional DTE capability when full duplex operation is in use. The PAUSE function shall be enabled according to Table 28B-3 only if the Highest Common Denominator is a full duplex technology. There is no priority resolution associated with the PAUSE operation.

**Table 28B–3—Pause resolution**

Local device		Link partner		Local device resolution	Link partner resolution
PAUSE	ASM_DIR	PAUSE	ASM_DIR		
0	0	Don't care	Don't care	Disable PAUSE Transmit and Receive	Disable PAUSE Transmit and Receive
0	1	0	Don't care	Disable PAUSE Transmit and Receive	Disable PAUSE Transmit and Receive
0	1	1	0	Disable PAUSE Transmit and Receive	Disable PAUSE Transmit and Receive
0	1	1	1	Enable PAUSE transmit Disable PAUSE receive	Enable PAUSE receive Disable PAUSE transmit
1	0	0	Don't care	Disable PAUSE Transmit and Receive	Disable PAUSE Transmit and Receive
1	Don't care	1	Don't care	Enable PAUSE Transmit and Receive	Enable PAUSE Transmit and Receive
1	1	0	0	Disable PAUSE Transmit and Receive	Disable PAUSE Transmit and Receive
1	1	0	1	Enable PAUSE receive Disable PAUSE transmit	Enable PAUSE transmit Disable PAUSE receive

### 28B.4 Message Page transmission convention

Each series of Unformatted Pages shall be preceded by a Message Page containing a Message Code that defines how the following Unformatted Pages will be used.

Next Page message codes should be allocated globally across Selector Field values so that meaningful communication is possible between technologies using different Selector Field values.

## Annex 28C

(normative)

### Next Page Message Code Field definitions

The Message Code Field of a message page used in Next Page exchange shall be used to identify the meaning of a message. The following table identifies the types of messages that may be sent. As new messages are developed, this table will be updated accordingly.

The Message Code Field uses an 11-bit binary encoding that allows 2048 messages to be defined. All Message Codes not specified shall be reserved for IEEE use or allocation.

**Table 28C–1 – Message code field values**

Message Code #	M 10	M 9	M 8	M 7	M 6	M 5	M 4	M 3	M 2	M 1	M 0	Message Code Description
0	0	0	0	0	0	0	0	0	0	0	0	Reserved for future Auto-Negotiation use
1	0	0	0	0	0	0	0	0	0	0	1	Null Message
2	0	0	0	0	0	0	0	0	0	1	0	One UP with Technology Ability Field follows
3	0	0	0	0	0	0	0	0	0	1	1	Two UPs with Technology Ability Field follow
4	0	0	0	0	0	0	0	0	1	0	0	One UP with Binary coded Remote fault follows
5	0	0	0	0	0	0	0	0	1	0	1	Organizationally Unique Identifier Tagged Message
6	0	0	0	0	0	0	0	0	1	1	0	PHY Identifier Tag Code
7	0	0	0	0	0	0	0	0	1	1	1	100BASE-T2 Technology Message Code. 100BASE-T2 Ability Page to follow using Unformatted Next Page
8	0	0	0	0	0	0	0	1	0	0	0	1000BASE-T Technology Message Code. Two 1000BASE-T Ability Pages to follow using Unformatted Next Pages.
9.....	0	0	0	0	0	0	0	1	0	0	1	Reserved for future Auto-Negotiation use
.....2047	1	1	1	1	1	1	1	1	1	1	1	Reserved for future Auto-Negotiation use

#### 28C.1 Message code #0— Auto-Negotiation reserved code 1

This code is reserved for future Auto-Negotiation function enhancements. Devices shall not transmit this code.

## 28C.2 Message code #1—Null Message code

The Null Message code shall be transmitted during Next Page exchange when the Local Device has no further messages to transmit and the Link Partner is still transmitting valid Next Pages. See 28.2.3.4 for more details.

## 28C.3 Message code #2—Technology Ability extension code 1

This Message Code is reserved for future expansion of the Technology Ability Field and indicates that a defined user code with a specific Technology Ability Field encoding follows.

## 28C.4 Message code #3—Technology Ability extension code 2

This Message Code is reserved for future expansion of the Technology Ability Field and indicates that two defined user codes with specific Technology Ability Field encodings follow.

## 28C.5 Message code #4—Remote fault number code

This Message Code shall be followed by a single user code whose encoding specifies the type of fault that has occurred. The following user codes are defined:

- 0: RF Test  
This code can be used to test Remote Fault operation.
- 1: Link Loss
- 2: Jabber
- 3: Parallel Detection Fault  
This code may be sent to identify when bit 6.4 is set.

## 28C.6 Message code #5—Organizationally Unique Identifier (OUI) tag code

The OUI Tagged Message shall consist of a single message code of 0000 0000 0101 followed by four user codes defined as follows. The first user code shall contain the most significant 11 bits of the OUI (bits 23:13) with the most significant bit in bit 10 of the user code. The second user code shall contain the next most significant 11 bits of the OUI (bits 12:2) with the most significant bit in bit 10 of the user code. The third user code shall contain the remaining least significant 2 bits of the OUI (bits 1:0) with the most significant bit in bit 10 of the user code. Bits 8:0 of the fourth user contain a user-defined user code value that is specific to the OUI transmitted. The fourth and final user code shall contain a user-defined user code value that is specific to the OUI transmitted.

## 28C.7 Message code #6—PHY identifier tag code

The PHY ID tag code message shall consist of a single message code of 0000 0000 0110 followed by four user codes defined as follows. The first user code shall contain the most significant 11 bits of the PHY ID (2.15:5) with the most significant bit in bit 10 of the user code. The second user code shall contain bits 2.4:0 to 3.15:10 of the PHY ID with the most significant bit in bit 10 of the user code. The third user code shall contain bits 3.9:0 of the PHY ID with the most significant bit in bit 10 of the user code. Bit 0 in the third user code shall contain a user-defined user code value that is specific to the PHY ID transmitted. The fourth and final user code shall contain a user-defined user code value that is specific to the PHY ID transmitted.



**28C.8 Message code #2047—Auto-Negotiation reserved code 2**

This code is reserved for future Auto-Negotiation function enhancements. Devices shall not transmit this code.

**28C.9 Message code #7—100BASE-T2 technology message code**

Clause 32 (100BASE-T2) uses Next Page Message Code 7 to indicate that T2 implementations will follow the transmission of this page [the initial, Message (formatted) Next Page] with two unformatted Next Pages which contain information defined in 32.5.4.2.

**28C.10 Message Code #8 - 1000BASE-T technology message code**

Clause 40 (1000BASE-T) uses Next Page Message Code 8 to indicate that 1000BASE-T implementations will follow the transmission of this page [the initial, Message (formatted) Next Page] with two unformatted Next Pages that contain information defined in 40.5.1.2.

## Annex 28D

(normative)

### Description of extensions to Clause 28 and associated annexes

#### 28D.1 Introduction

This annex is to be used to document extensions and modifications to Clause 28 required by IEEE 802.3® clauses and other standards that use Auto-Negotiation and that were approved after June 1995. It provides a single location to define such extensions and modifications without changing the basic contents of Clause 28.

Subclause 28D.2 lists those clauses and standards that require extensions to Clause 28 and provides pointers to the subclauses where those extensions are listed.

#### 28D.2 Extensions to Clause 28

##### 28D.2.1 Extensions required for Clause 31 (full duplex)

Clause 31 (full duplex) requires the use of Auto-Negotiation. Extensions to Clause 28 and associated annexes required for the correct operation of full duplex are shown in 28D.3.

##### 28D.2.2 Extensions required for Clause 32 (100BASE-T2)

Clause 32 (100BASE-T2) requires the use of Auto-Negotiation. Extensions to Clause 28 required for correct operation of 100BASE-T2 are shown in 28D.4.

#### 28D.3 Extensions for Clause 31

Full duplex requires the use of bit A5 in the Technology Ability Field of the IEEE 802.3® Selector Base Page. (This bit is also defined as MII bit 4.10.) This bit was previously defined as “reserved for future technology.”

Bit	Technology	Minimum cabling requirement
A5	PAUSE operation for full duplex links	Not applicable

Bit A5 (PAUSE operation for full duplex links) signifies that the DTE has implemented both the optional MAC Control sublayer and the PAUSE function as specified in Clause 31 and Annex 31B. This capability is significant only when the link is configured for full duplex operation, regardless of data rate and medium.

## 28D.4 Extensions for Clause 32 (100BASE-T2)

Clause 32 (100BASE-T2) makes special use of Auto-Negotiation and requires additional MII registers. This use is summarized below. Details are provided in 32.5.

Auto-Negotiation is mandatory for 100BASE-T2 (32.1.3.4).

100BASE-T2 introduces the concept of MASTER and SLAVE to define DTEs and to facilitate the timing of transmit and receive operations. Auto-Negotiation is used to provide information used to configure MASTER/SLAVE status (32.5.4.3).

100BASE-T2 uses unique next page transmit and receive registers (MII Registers 8, 9 and 10) in conjunction with Auto-Negotiation. These registers are in addition to Registers 0–7 as defined in 28.2.4 (32.5.2).

100BASE-T2 use of Auto-Negotiation generates information which is stored in configuration and status bits defined for the MASTER-SLAVE Control register (MII Register 9) and the MASTER-SLAVE Status register (MII Register 10).

100BASE-T2 requires an ordered exchange of next page messages (32.5.1).

100BASE-T2 parameters are configured based on information provided by the ordered exchange of next page messages.

100BASE-T2 adds new message codes to be transmitted during Auto-Negotiation (32.5.4.2).

100BASE-T2 adds 100BASE-T2 full duplex and half duplex capabilities to the priority resolution table (28B.3) and MII Status Register (22.2.4.2).

T2 is defined as a valid value for “x” in 28.3.1 (e.g., link\_status\_T2). T2 represents that the 100BASE-T2 PMA is the signal source.

## 28D.5 Extensions required for Clause 40 (1000BASE-T)

Clause 40 (1000BASE-T) makes special use of Auto-Negotiation and requires additional MII registers. This use is summarized below. Details are provided in 40.5.

- a) Auto-Negotiation is mandatory for 1000BASE-T. (40.5.1)
- b) 1000BASE-T requires an ordered exchange of Next Page messages. (40.5.1.2)
- c) 1000BASE-T parameters are configured based on information provided by the ordered exchange of NEXT Page messages.
- d) 1000BASE-T uses MASTER and SLAVE to define PHY operations and to facilitate the timing of transmit and receive operations. Auto-Negotiation is used to provide information used to configure MASTER-SLAVE status.(40.5.2)
- e) 1000BASE-T transmits and receives Next Pages for exchange of information related to MASTER-SLAVE operation. The information is specified in MII registers 9 and 10 (see 32.5.2 and 40.5.1.1), which are required in addition to registers 0-8 as defined in 28.2.4.
- f) 1000BASE-T adds new message codes to be transmitted during Auto-Negotiation. (40.5.1.3)
- g) 1000BASE-T adds 1000BASE-T full duplex and half duplex capabilities to the priority resolution table. (28B.3) and MII Extended Status Register (22.2.2.4)
- h) 1000BASE-T is defined as a valid value for “x” in 28.3.1 (e.g., link\_status\_1GigT.) 1GigT represents that the 1000BASE-T PMA is the signal source.
- i) 1000BASE-T supports Asymmetric Pause as defined in Annex 28B.

## **Annex 29A**

(informative)

### **DTE and repeater delay components**

#### **29A.1 DTE delay**

Round-trip DTE delay = MAC transmit start to MDI output  
+ MDI input to MDI output (worst case, nondeferred)  
+ MDI input to collision detect

NOTE 1—Refer to Clauses 23, 24, 25, and 26.

NOTE 2—Worst-case values are used for the one T4 and one TX/FX value shown in Table 29-3. (TX/FX values for MAC transmit start and MDI input to collision detect; T4 value for MDI input to MDI output.)

#### **29A.2 Repeater delay**

Repeater delay= SOP (start-of-packet propagation delay)  
+ SOJ (start-of-jam propagation delay)

NOTE—Refer to Clause 27.

## Annex 29B

(informative)

### Recommended topology documentation

It is strongly recommended that detailed records documenting the topology components of 100BASE-T networks be prepared and maintained to facilitate subsequent modification. Proper 100BASE-T topology design requires an accurate knowledge of link segment and hub parameters to ensure proper operation of single and multi-segment, single collision domain networks. Link segment documentation is site-specific and requires careful documentation. It is recommended that the information shown in Table 29B–1 be collected for each link segment and archived for future reference. Hub performance parameters may be obtained from manufacturer documentation.

**Table 29B–1 – Recommended link segment documentation**

	<b>Horizontal wiring (wiring closet, from punch-down block to end station wall plate)</b>	<b>MII cable(s)</b>	<b>Wiring closet patch cord</b>	<b>End station connecting cable</b>
Length				
Type (e.g., Category 3)				
Cable manufacturer				
Cable code/id (from manufacturer)				
Cable delay (in bit times per meter)				

## Annex 30A

(normative)

### GDMO specification for 802.3® managed object classes

This annex formally defines the protocol encodings for CMIP and ISO/IEC 15802-2: 1995 [ANSI/IEEE Std 802.1B™ and 802.1k™, 1995 Edition] for the IEEE 802.3® Managed Objects using the templates specified in ISO/IEC 10165-4: 1992. The application of a GDMO template compiler against 30A.1 to 30A.8 will produce the proper protocol encodings.

NOTE 1—The arcs (that is, object identifier values) defined in Annex 30A deprecate the arcs previously defined in Annexes H.1 (Layer Management), H.2 (Repeater Management), and H.3 (MAU Management). See IEEE Std 802.1F-1993™, Annex C.4.

NOTE 2—During the update for 1000 Mb/s operation differences between objects in the root of the registration arcs were harmonized. All instances of *{iso(1) std(0) iso8802(8802) csma(3)...* were changed to *{iso(1) member-body(2) us(840) 802dot3(10006)...* in order to harmonize with the rest of this GDMO specification. For maximum compatibility with previous implementations it is recommended that all implementations respond equally to requests for communication based on either registration arc.

Each attribute definition in this clause references directly by means of the WITH ATTRIBUTE SYNTAX construct or indirectly by means of the DERIVED FROM construct an ASN.1 type or subtype that defines the attribute's type and range. Those ASN.1 types and subtypes defined exclusively for CSMA/CD Management appear in a single ASN.1 module in 30B.1.

Counters for these protocol encodings are specified as either 32 or 64 bits wide. Thirty-two bit counters are used for the protocol encoding of counter attributes, providing the minimum rollover time is 58 min or more. Sixty-four bit counters are used for the protocol encoding of counter attributes that could roll over in less than 58 min with a 32-bit counter. Approximate counter rollover times are provided as notes below each counter BEHAVIOUR definition. 1000 Mb/s counters are 10 times faster than 100 Mb/s counters, similarly 100 Mb/s counters are 10 times faster than 10 Mb/s counters. For formal definition of the counter, refer to the BEHAVIOUR bCMCounter in 30B.1.

### 30A.1 DTE MAC entity managed object class

#### 30A.1.1 DTE MAC entity formal definition

<b>oMACEntity</b>	<b>MANAGED OBJECT CLASS</b>
DERIVED FROM	“CCITT Rec. X.721 (1992)   ISO/IEC 10165-2 : 1992”:top;
CHARACTERIZED BY	
pBasic	PACKAGE
ATTRIBUTES	aMACID GET;
ACTIONS	acInitializeMAC;
;	
;	
CONDITIONAL PACKAGES	
pMandatory	PACKAGE
ATTRIBUTES	aFramesTransmittedOK GET,
	aSingleCollisionFrames GET,

	aMultipleCollisionFrames	GET,
	aFramesReceivedOK	GET,
	aFrameCheckSequenceErrors	GET,
	aAlignmentErrors	GET,
	aMACCapabilities	GET,
	aDuplexStatus	GET-SET;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006)	
PRESENT IF	csmacdmgt(30) package(4) macMandatoryPkg(1)};	
	Conformance to DTE Management is desired.	
	Attributes aMACCapabilities and aDuplexStatus are	
	mandatory in systems that can operate in full duplex	
	mode and are recommended in systems that can only	
	operate in half duplex mode.;	
pRecommended	PACKAGE	
ATTRIBUTES	aOctetsTransmittedOK	GET,
	aFramesWithDeferredXmissions	GET,
	aLateCollisions	GET,
	aFramesAbortedDueToXSColls	GET,
	aFramesLostDueToIntMACXmitError	GET,
	aCarrierSenseErrors	GET,
	aOctetsReceivedOK	GET,
	aFramesLostDueToIntMACRcvError	GET,
	aPromiscuousStatus	GET-SET,
	aReadMulticastAddressList	GET;
ACTIONS	acAddGroupAddress,	
	acDeleteGroupAddress;	
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006)	
PRESENT IF	csmacdmgt(30) package(4)	
	macRecommendedPkg(2)};	
	The Recommended Package is implemented.;	
pOptional	PACKAGE	
ATTRIBUTES	aMulticastFramesXmittedOK	GET,
	aBroadcastFramesXmittedOK	GET,
	aMulticastFramesReceivedOK	GET,
	aBroadcastFramesReceivedOK	GET,
	aInRangeLengthErrors	GET,
	aOutOfRangeLengthField	GET,
	aFrameTooLongErrors	GET,
	aMACEnableStatus	GET-SET,
	aTransmitEnableStatus	GET-SET,
	aMulticastReceiveStatus	GET-SET,
	aReadWriteMACAddress	GET-SET;
ACTIONS	acExecuteSelfTest;	
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006)	
PRESENT IF	csmacdmgt(30) package(4) optionalPkg(3)};	
	The Optional Package and the Recommended Package	
	are implemented.;	
pArray	PACKAGE	
ATTRIBUTES	aCollisionFrames	GET;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006)	
PRESENT IF	csmacdmgt(30) package(4) arrayPkg(4)};	
	The Array Package and the Recommended Package	
	are implemented.;	
pExcessiveDeferral	PACKAGE	
ATTRIBUTES	aFramesWithExcessiveDeferral	GET;

REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) package(4) excessiveDeferralPkg(5)};
PRESENT IF	The ExcessiveDeferral Package and the Recommended Package are implemented.;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) managedObjectClass(3) macObjectClass(1)};

**nbMACName****NAME BINDING**

SUBORDINATE OBJECT CLASS	oMACEntity;
NAMED BY SUPERIOR OBJECT CLASS	“ISO/IEC 10165-2”:system;
WITH ATTRIBUTE	aMACID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) nameBinding(6) macName(1)};

**nbMACMonitor****NAME BINDING**

SUBORDINATE OBJECT CLASS	“IEEE802.1F”:ewmaMetricMonitor;
NAMED BY SUPERIOR OBJECT CLASS	“ISO/IEC 10165-2”:system;
WITH ATTRIBUTE	aScannerId;
CREATE	WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE	ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) nameBinding(6) macMonitor(2)};

**nbMAC-MACControl****NAME BINDING**

SUBORDINATE OBJECT CLASS	oMACEntity;
NAMED BY SUPERIOR OBJECT CLASS	oMACControlEntity;
WITH ATTRIBUTE	aMACID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) nameBinding(6) nbMAC-MACControl(16)};

**nbMAC-Aggregator****NAME BINDING**

SUBORDINATE OBJECT CLASS	oMACEntity;
NAMED BY SUPERIOR OBJECT CLASS	oAggregator;
WITH ATTRIBUTE	aMACID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) nameBinding(6) nbMAC-Aggregator(17)};

**30A.1.2 DTE MAC entity attributes****aMACID****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bMACID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)}



attribute(7) macID(1));

**bMACID**

DEFINED AS

**BEHAVIOUR**

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.1;

**aFramesTransmittedOK**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

**ATTRIBUTE**

aCMCounter;  
bFramesTransmittedOK;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) framesTransmittedOK(2)};

**bFramesTransmittedOK**

DEFINED AS

**BEHAVIOUR**

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.2;

The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

NOTE 2—This maps to framesSent (of the mandatory macPackage) in ISO/IEC 10742: 1994.;

**aSingleCollisionFrames**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

**ATTRIBUTE**

aCMCounter;  
bSingleCollisionFrames;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) singleCollisionFrames(3)};

**bSingleCollisionFrames**

DEFINED AS

**BEHAVIOUR**

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.3;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 103 h.;

**aMultipleCollisionFrames**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

**ATTRIBUTE**

aCMCounter;  
bMultipleCollisionFrames;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) multipleCollisionFrames(4)};

**bMultipleCollisionFrames**

DEFINED AS

**BEHAVIOUR**

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.4;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 125 h.;

**aFramesReceivedOK**

DERIVED FROM  
BEHAVIOUR

**ATTRIBUTE**

aCMCounter;  
bFramesReceivedOK;

REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) framesReceivedOK(5)};
<b>bFramesReceivedOK</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.5;
	The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;
	NOTE 2—This maps to framesReceived (of the mandatory macPackage) in ISO/IEC 10742: 1994.;
<b>aFrameCheckSequenceErrors</b>	<b>ATTRIBUTE</b>
DERIVED FROM BEHAVIOUR REGISTERED AS	aCMCounter; bFrameCheckSequenceErrors; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) frameCheckSequenceErrors(6)};
<b>bFrameCheckSequenceErrors</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.6;
	NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;
<b>aAlignmentErrors</b>	<b>ATTRIBUTE</b>
DERIVED FROM BEHAVIOUR REGISTERED AS	aCMCounter; bAlignmentErrors; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) alignmentErrors(7)};
<b>bAlignmentErrors</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.7;
	NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;
<b>aOctetsTransmittedOK</b>	<b>ATTRIBUTE</b>
DERIVED FROM BEHAVIOUR REGISTERED AS	aCMCounter; bOctetsTransmittedOK; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) octetsTransmittedOK(8)};
<b>bOctetsTransmittedOK</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.8;
	The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 min.

NOTE 2—This maps to octetsSent (of the mandatory macPackage) in ISO/IEC 10742: 1994.;

**aFramesWithDeferredXmissions****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bFramesWithDeferredXmissions;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) framesWithDeferredXmissions(9)};

**bFramesWithDeferredXmissions****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.9;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 103 h.;

**aLateCollisions****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bLateCollisions;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) lateCollisions(10)};

**bLateCollisions****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.10;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aFramesAbortedDueToXSColls****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bFramesAbortedDueToXSColls;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) framesAbortedDueToXSColls(11)};

**bFramesAbortedDueToXSColls****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.11;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 53 days.;

**aFramesLostDueToIntMACXmitError****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bFramesLostDueToIntMACXmitError;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) framesLostDueToIntMACXmitError(12)};

**bFramesLostDueToIntMACXmitError****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.12;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 16 h.;

**aCarrierSenseErrors****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bCarrierSenseErrors;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) carrierSenseErrors(13)};

**bCarrierSenseErrors****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.13;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aOctetsReceivedOK****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bOctetsReceivedOK;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) octetsReceivedOK(14)};

**bOctetsReceivedOK****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.14;

The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 min.

This maps to octetsReceived (of the mandatory macPackage) in ISO/IEC 10742: 1994.;

**aFramesLostDueToIntMACRcvError****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bFramesLostDueToIntMACRcvError;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) framesLostDueToIntMACRcvError(15)};

**bFramesLostDueToIntMACRcvError****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.15;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aPromiscuousStatus****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX  
BEHAVIOUR  
REGISTERED AS

IEEE802Dot3-MgmtAttributeModule.TrueFalse;  
bPromiscuousStatus;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) promiscuousStatus(16)};

<b>bPromiscuousStatus</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.16;
<b>aReadMulticastAddressList</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. MulticastAddressList
BEHAVIOUR	bReadMulticastAddressList;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) readMulticastAddressList(17)};
<b>bReadMulticastAddressList</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.17;
<b>aMulticastFramesXmittedOK</b>	<b>ATTRIBUTE</b>
DERIVED FROM	aCMCounter;
BEHAVIOUR	bMulticastFramesXmittedOK;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) multicastFramesXmittedOK(18)};
<b>bMulticastFramesXmittedOK</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.18;
	NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;
<b>aBroadcastFramesXmittedOK</b>	<b>ATTRIBUTE</b>
DERIVED FROM	aCMCounter;
BEHAVIOUR	bBroadcastFramesXmittedOK;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) broadcastFramesXmittedOK(19)};
<b>bBroadcastFramesXmittedOK</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.19;
	NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;
<b>aFramesWithExcessiveDeferral</b>	<b>ATTRIBUTE</b>
DERIVED FROM	aCMCounter;
BEHAVIOUR	bFramesWithExcessiveDeferral;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) framesWithExcessiveDeferral(20)};
<b>bFramesWithExcessiveDeferral</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.20;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 days.;

**aMulticastFramesReceivedOK****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bMulticastFramesReceivedOK;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) multicastFramesReceivedOK(21)};

**bMulticastFramesReceivedOK****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.21;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aBroadcastFramesReceivedOK****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bBroadcastFramesReceivedOK;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) broadcastFramesReceivedOK(22)};

**bBroadcastFramesReceivedOK****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.22;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aInRangeLengthErrors****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bInRangeLengthErrors;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) inRangeLengthErrors(23)};

**bInRangeLengthErrors****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.23;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aOutOfRangeLengthField****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bOutOfRangeLengthField;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) outOfRangeLengthField(24)};

**bOutOfRangeLengthField****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.24;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

<b>aFrameTooLongErrors</b>	<b>ATTRIBUTE</b>
DERIVED FROM BEHAVIOUR REGISTERED AS	aCMCounter; bFrameTooLongErrors; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) frameTooLongErrors(25)};
<b>bFrameTooLongErrors</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.25;
	NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 61 days.;
<b>aMACEnableStatus</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX BEHAVIOUR REGISTERED AS	IEEE802Dot3-MgmtAttributeModule.TrueFalse; bMACEnableStatus; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) mACEnableStatus(26)};
<b>bMACEnableStatus</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.26;
<b>aTransmitEnableStatus</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX BEHAVIOUR REGISTERED AS	IEEE802Dot3-MgmtAttributeModule.TrueFalse; bTransmitEnableStatus; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) transmitEnableStatus(27)};
<b>bTransmitEnableStatus</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.27;
<b>aMulticastReceiveStatus</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX BEHAVIOUR REGISTERED AS	IEEE802Dot3-MgmtAttributeModule.TrueFalse; bMulticastReceiveStatus; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) multicastReceiveStatus(28)};
<b>bMulticastReceiveStatus</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.1.28;
<b>aReadWriteMACAddress</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX BEHAVIOUR REGISTERED AS	IEEE802CommonDefinitions.MACAddress; bReadWriteMACAddress; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)}

attribute(7) modifyMACAddress(29));

**bReadWriteMACAddress****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.29;

NOTE—This maps to localMACAddress (of the mandatory macPackage) in ISO/IEC 10742: 1994.;

**aCollisionFrames****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX

IEEE802Dot3-MgmtAttributeModule.AttemptArray;

BEHAVIOUR

bCollisionFrames;

REGISTERED AS

{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) collisionFrames(30)};

**bCollisionFrames****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.30;

NOTE—The approximate minimum time for any single counter rollover for 10 Mb/s operation is 103 h.;

**aMACCapabilities****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX

IEEE802Dot3-

MgmtAttributeModule.MACCapabilitiesList;

MATCHES FOR

EQUALITY, ORDERING;

BEHAVIOUR

bMACCapabilities;

REGISTERED AS

{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aMACCapabilities(89)};

**bMACCapabilities****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.31;

**aDuplexStatus****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX

IEEE802Dot3-MgmtAttributeModule.DuplexValues;

MATCHES FOR

EQUALITY;

BEHAVIOUR

bDuplexStatus;

REGISTERED AS

{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aDuplexStatus(90)};

**bDuplexStatus****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.1.1.32;

**30A.1.3 DTE MAC entity actions****acInitializeMAC****ACTION**

BEHAVIOUR

bInitializeMAC;

MODE

CONFIRMED;



REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) action(9) initializeMAC(1)};
<b>bInitializeMAC</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.2.1;
<b>acAddGroupAddress</b>	<b>ACTION</b>
BEHAVIOUR	bAddGroupAddress;
MODE	CONFIRMED;
WITH INFORMATION SYNTAX	IEEE802CommonDefinitions.MACAddress;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) action(9) addGroupAddress(2)};
<b>bAddGroupAddress</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.2.2;
<b>acDeleteGroupAddress</b>	<b>ACTION</b>
BEHAVIOUR	bDeleteGroupAddress;
MODE	CONFIRMED;
WITH INFORMATION SYNTAX	IEEE802CommonDefinitions.MACAddress;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) action(9) deleteGroupAddress(3)};
<b>bDeleteGroupAddress</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.2.3;
<b>acExecuteSelfTest</b>	<b>ACTION</b>
BEHAVIOUR	bExecuteSelfTestMAC;
MODE	CONFIRMED;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) action(9) executeSelfTestMAC(4)};
<b>bExecuteSelfTestMAC</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.1.2.4;

## 30A.2 DTE physical entity managed object class

### 30A.2.1 DTE physical entity formal definition

oPHYEntity	MANAGED OBJECT CLASS
DERIVED FROM	“CCITT Rec. X.721 (1992)   ISO/IEC 10165-2 : 1992”:top;
CHARACTERIZED BY	
pBasic	PACKAGE
ATTRIBUTES	aPHYID GET,

```

aPHYType          GET,
aPHYTypeList      GET,
aMIIDetect        GET,
aPHYAdminState    GET;
;
;
CONDITIONAL PACKAGES
  pRecommended
    ATTRIBUTES
    REGISTERED AS
      PACKAGE
      aSQETestErrors          GET;
      {iso(1) member-body(2) us(840) ieee802dot3(10006)
      csmacdmgt(30) package(4)
      phyRecommendedPkg(6)};
    PRESENT IF
      The Recommended Package is implemented.;
  pMultiplePhy
    ACTIONS
    REGISTERED AS
      PACKAGE
      acPHYAdminControl;
      {iso(1) member-body(2) us(840) ieee802dot3(10006)
      csmacdmgt(30) package(4)
      phyMultiplePhyPkg(7)};
    PRESENT IF
      There is more than one PHY per MAC.;
  p100MbpsMonitor
    ATTRIBUTES
    REGISTERED AS
      PACKAGE
      aSymbolErrorDuringCarrier  GET;
      {iso(1) member-body(2) us(840) ieee802dot3(10006)
      csmacdmgt(30) package(4)
      phy100MbpsMonitor(8)};
    PRESENT IF
      The 100/1000 Mb/s Monitor capability is
      implemented.;
REGISTERED AS      {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)
managedObjectClass(3) phyObjectClass(2)};

nbPHYName                NAME BINDING

SUBORDINATE OBJECT CLASS    oPHYEntity;
NAMED BY SUPERIOR OBJECT CLASS
WITH ATTRIBUTE
REGISTERED AS              {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)
nameBinding(6) phyName(3)};

nbPHYMonitor            NAME BINDING

SUBORDINATE OBJECT CLASS    "IEEE802.1F":ewmaMetricMonitor;
NAMED BY SUPERIOR OBJECT CLASS
WITH ATTRIBUTE
  CREATE
  DELETE
REGISTERED AS              {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)
nameBinding(6) phyMonitor(4)};

```

**30A.2.2 DTE physical entity attributes**

<b>aPHYID</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bPHYID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) phyID(31)};
<b>bPHYID</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.2.1.1;
<b>aPHYType</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.PhyTypeValue;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bPHYType;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) phyType(32)};
<b>bPHYType</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.2.1.2;
<b>aPHYTypeList</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.PhyTypeList;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bPHYTypeList;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) phyTypeList(33)};
<b>bPHYTypeList</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.2.1.3;
<b>aSQETestErrors</b>	<b>ATTRIBUTE</b>
DERIVED FROM	aCMCounter;
BEHAVIOUR	bSQETestErrors;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) sqeTestErrors(34)};
<b>bSQETestErrors</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.3.2.1.4;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aSymbolErrorDuringCarrier****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bSymbolErrorDuringCarrier;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) symbolErrorDuringCarrier(35)};

**bSymbolErrorDuringCarrier****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.2.1.5;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aMIIDetect** **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX  
MATCHES FOR  
BEHAVIOUR  
REGISTERED AS

IEEE802Dot3-MgmtAttributeModule.MIIDetect;  
EQUALITY;  
bMIIDetect;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) mIIDetect(36)};

**bMIIDetect****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.2.1.6;

**aPHYAdminState****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX  
MATCHES FOR  
BEHAVIOUR  
REGISTERED AS

IEEE802Dot3-MgmtAttributeModule.  
PortAdminState;  
EQUALITY, ORDERING;  
bPHYAdminState;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) phyAdminState(37)};

**bPHYAdminState****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.2.1.7;

**30A.2.3 DTE physical entity actions****acPHYAdminControl****ACTION**

BEHAVIOUR  
MODE  
WITH INFORMATION SYNTAX

bPHYAdminControl;  
CONFIRMED;  
IEEE802Dot3-MgmtAttributeModule.  
PortAdminState;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
action(9) phyAdminControl(5)};

**bPHYAdminControl****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.3.2.2.1;

### 30A.3 DTE MAC control entity managed object class

#### 30A.3.1 DTE MAC control entity formal definition

```

oMACControlEntity                                MANAGED OBJECT CLASS

DERIVED FROM                                        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;
CHARACTERIZED BY
  pMandatory
  ATTRIBUTES
  ;
  ;
CONDITIONAL PACKAGES
  pRecommended
  ATTRIBUTES
  REGISTERED AS
  PRESENT IF
REGISTERED AS{iso(1) member-body(2) us(840) ieee802dot3(10006)
  csmacdmgt(30)
  managedObjectClass(3) macControlObjectClass(8)};

nbMACControl-System                               NAME BINDING

SUBORDINATE OBJECT CLASS                          oMACControlEntity;
NAMED BY SUPERIOR OBJECT CLASS                    "ISO/IEC 10165-2":system;
WITH ATTRIBUTE                                    aMACControlID;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) name-
Binding(6) nbMACControl-System(18)};

nbMACControl-Aggregator                           NAME BINDING

SUBORDINATE OBJECT CLASS                          oMACControlEntity;
NAMED BY SUPERIOR OBJECT CLASS                    oAggregator;
WITH ATTRIBUTE                                    aMACControlID;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) name-
Binding(6) nbMACControl-Aggregator(19)};

nbMACControlMonitor                               NAME BINDING

SUBORDINATE OBJECT CLASS "IEEE802.1F":ewmaMetricMonitor;
NAMED BY SUPERIOR OBJECT CLASS
  "ISO/IEC 10165-2":system;
WITH ATTRIBUTE EaScannerId;
  CREATE                                          WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE                                         ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS{iso(1) member-body(2) us(840) ieee802dot3(10006)
  csmacdmgt(30) nameBinding(6) macControlMonitor(15)};

```

**30A.3.2 DTE MAC Control entity attributes**

<b>aMACControlID</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR	EQUALITY;
BEHAVIOR	bMACControlID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aMACControlID(92)};
<b>bMACControlID</b>	<b>BEHAVIOR</b>
DEFINED AS	See "BEHAVIOR DEFINED AS" in 30.3.3.1;
<b>aMACControlFunctionsSupported</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. MACControlFunctionsList;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOR	bMACControlFunctionsSupported;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aMACControlFunctionsSupported(93)};
<b>bMACControlFunctionsSupported</b>	<b>BEHAVIOR</b>
DEFINED AS	See "BEHAVIOR DEFINED AS" in 30.3.3.2;
<b>aMACControlFramesTransmitted</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	aCMCounter;
BEHAVIOR	bMACControlFramesTransmitted;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aMACControlFramesTransmitted(94)};
<b>bMACControlFramesTransmitted</b>	<b>BEHAVIOR</b>
DEFINED AS	See "BEHAVIOR DEFINED AS" in 30.3.3.3;
<b>aMACControlFramesReceived</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	aCMCounter;
BEHAVIOR	bMACControlFramesReceived;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aMACControlFramesReceived(95)};
<b>bMACControlFramesReceived</b>	<b>BEHAVIOR</b>
DEFINED AS	See "BEHAVIOR DEFINED AS" in 30.3.3.4;
<b>aUnsupportedOpcodesReceived</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	aCMCounter;
BEHAVIOR	bUnsupportedOpcodesReceived;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aUnsupportedOpcodesReceived(96)};

**bUnsupportedOpcodesReceived****BEHAVIOR**

DEFINED AS See "BEHAVIOR DEFINED AS" in 30.3.3.5;

**30A.4 DTE MAC Control function entity managed object class****30A.4.1 DTE MAC Control function entity formal definition**

oMACControlFunctionEntityMANAGED OBJECT CLASS

DERIVED FROM"CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992":top;

CHARACTERIZED BY

pMandatory	PACKAGE	
ATTRIBUTES	aPAUSELinkDelayAllowance	GET-SET;
;		

;

CONDITIONAL PACKAGES

pRecommended	PACKAGE	
ATTRIBUTES	aPAUSEMACCtrlFramesTransmitted	GET,
	aPAUSEMACCtrlFramesReceived	GET;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006)	
	csmacdmgt(30)	
	package(4) macControlFunctionRecomendedPkg(17));	

PRESENT IF	The Recommended Package is implemented.;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006)
	csmacdmgt(30)
	managedObjectClass(3)macControlFunctionObjectClass(9));

**30A.4.2 DTE MAC Control function entity attributes****aPAUSELinkDelayAllowance****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX	LinkDelayAllowance;
BEHAVIOR	bPAUSELinkDelayAllowance;
REGISTERED AS	{iso(1) memberbody(2) us(840) ieee802dot3(10006)
	csmacdmgt(30) attribute(7) aPAUSELinkDelayAllowance(97));

**bPAUSELinkDelayAllowance****BEHAVIOR**

DEFINED AS See "BEHAVIOR DEFINED AS" in 30.3.4.1;

**aPAUSEMACCtrlFramesTransmitted****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX	aCMCounter;
BEHAVIOR	bPAUSEMACCtrlFramesTransmitted;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006)
	csmacdmgt(30) attribute(7) aPAUSEMACCtrlFramesTransmitted(98));

**bPAUSEMACCtrlFramesTransmitted****BEHAVIOR**

DEFINED AS See "BEHAVIOR DEFINED AS" in 30.3.4.2;

<b>aPAUSEMACCtrlFramesReceived</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	aCMCounter;
BEHAVIOR	bPAUSEMACCtrlFramesReceived;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aPAUSEMACCtrlFramesReceived(99)};

<b>bPAUSEMACCtrlFramesReceived</b>	<b>BEHAVIOR</b>
DEFINED AS	See "BEHAVIOR DEFINED AS" in 30.3.4.3;

## 30A.5 Repeater managed object class

### 30A.5.1 Repeater, formal definition

oRepeater	MANAGED OBJECT CLASS
DERIVED FROM	“CCITT Rec. X.721 (1992)   ISO/IEC 10165-2 1992”:top;
CHARACTERIZED BY	
pRepeaterBasicControl	PACKAGE
ATTRIBUTES	aRepeaterID GET, aRepeaterType GET, aRepeaterGroupCapacity GET, aGroupMap GET, aRepeaterHealthState GET, aRepeaterHealthText GET, aRepeaterHealthData GET;
ACTIONS	acResetRepeater, acExecuteNonDisruptiveSelfTest;
NOTIFICATIONS	nRepeaterHealth, nRepeaterReset, nGroupMapChange;
;	
;	
CONDITIONAL PACKAGES	
pRepeaterPerfMonitor	PACKAGE
ATTRIBUTES	aTransmitCollisions GET;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) package(4) repeaterPerfMonitorPkg(9)};
PRESENT IF	The Performance Monitor Capability is implemented.;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) managedObjectClass(3) repeaterObjectClass(3)};

<b>nbRepeaterName</b>	<b>NAME BINDING</b>
SUBORDINATE OBJECT CLASS	repeater;
NAMED BY SUPERIOR OBJECT CLASS	“ISO/IEC 10165-2”:system AND SUBCLASSES;
WITH ATTRIBUTE	aRepeaterID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) nameBinding(6) repeaterName(5)};



**nbRepeaterMonitor****NAME BINDING**

SUBORDINATE OBJECT CLASS "IEEE802.1F":oEWMAMetricMonitor;  
 NAMED BY SUPERIOR OBJECT CLASS "ISO/IEC 10165-2":system AND SUBCLASSES;  
 WITH ATTRIBUTE aScannerId;  
 CREATE WITH-AUTOMATIC-INSTANCE-NAMING;  
 DELETE ONLY-IF-NO-CONTAINED-OBJECTS;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) nameBinding(6) repeaterMonitor(6)};

**30A.5.2 Repeater attributes****aRepeaterID****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bRepeaterID;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) repeaterID(38)};

**bRepeaterID****BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.1.1.1.

**aRepeaterType****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.RepeaterType;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bRepeaterType;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) repeaterType (39)};

**bRepeaterType****BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.1.1.2.

**aRepeaterGroupCapacity****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bRepeaterGroupCapacity;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) repeaterGroupCapacity(40)};

**bRepeaterGroupCapacity****BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.1.1.3.

**aGroupMap****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.BitString;

MATCHES FOR BEHAVIOUR	EQUALITY; bGroupMap;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) groupMap(41)};
<b>bGroupMap</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.1.1.4.
<b>aRepeaterHealthState</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. RepeaterHealthState;
MATCHES FOR BEHAVIOUR	EQUALITY; bRepeaterHealthState;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) repeaterHealthState(42)};
<b>bRepeaterHealthState</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.1.1.5.
<b>aRepeaterHealthText</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. RepeaterHealthText;
MATCHES FOR BEHAVIOUR	EQUALITY; bRepeaterHealthText;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) repeaterHealthText(43)};
<b>bRepeaterHealthText</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.1.1.6.
<b>aRepeaterHealthData</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. RepeaterHealthData;
MATCHES FOR BEHAVIOUR	EQUALITY; bRepeaterHealthData;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) repeaterHealthData(44)};
<b>bRepeaterHealthData</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.1.1.7.
<b>aTransmitCollisions</b>	<b>ATTRIBUTE</b>
DERIVED FROM BEHAVIOUR	aCMCounter; bTransmitCollisions;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)

attribute(7) transmitCollisions (45));

**bTransmitCollisions****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.1.1.8.

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 16 h.

**30A.5.3 Repeater actions****acResetRepeater****ACTION**

BEHAVIOUR  
MODE  
REGISTERED AS

bResetRepeater;  
CONFIRMED;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
action(9) resetRepeater(6)};

**bResetRepeater****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.1.2.1.

**acExecuteNonDisruptiveSelfTest****ACTION**

BEHAVIOUR  
REGISTERED AS

bExecuteNonDisruptiveSelfTest;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
action(9) executeNonDisruptiveSelfTestAction(7)};

**bExecuteNonDisruptiveSelfTest****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.1.2.2.

**30A.5.4 Repeater notifications****nRepeaterHealth****NOTIFICATION**

BEHAVIOUR  
WITH INFORMATION SYNTAX

AND ATTRIBUTE IDS

;

REGISTERED AS

bRepeaterHealth;  
IEEE802Dot3-MgmtAttributeModule.  
RepeaterHealthInfo  
repeaterHealthState aRepeaterHealthState,  
repeaterHealthText aRepeaterHealthText,  
repeaterHealthData aRepeaterHealthData  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
notification(10) repeaterHealth(1)};

**bRepeaterHealth****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.1.3.1.

**nRepeaterReset****NOTIFICATION**

BEHAVIOUR  
WITH INFORMATION SYNTAX

bRepeaterReset;  
IEEE802Dot3-MgmtAttributeModule.

AND ATTRIBUTE IDS  
 RepeaterHealthInfo  
 repeaterHealthState aRepeaterHealthState,  
 repeaterHealthText aRepeaterHealthText,  
 repeaterHealthData aRepeaterHealthData  
 ;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 notification(10) repeaterReset(2)};

**bRepeaterReset****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.1.3.2.

**nGroupMapChange****NOTIFICATION**

BEHAVIOUR bGroupMapChange;  
 WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.BitString;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 notification(10) groupMapChange(3)};

**bGroupMapChange****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.1.3.3.

**30A.6 Group managed object class****30A.6.1 Group, formal definition**

oGroup MANAGED OBJECT CLASS  
 DERIVED FROM “CCITT Rec. X.721 (1992) | ISO/IEC 10165-2  
 1992”:top;  
 CHARACTERIZED BY  
 pGroupBasicControl PACKAGE  
 ATTRIBUTES aGroupID GET,  
 aGroupPortCapacity GET,  
 aPortMap GET;  
 NOTIFICATIONS nPortMapChange;  
 ;  
 ;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 managedObjectClass(3) groupObjectClass(4)};

**nbGroupName****NAME BINDING**

SUBORDINATE OBJECT CLASS oGroup;  
 NAMED BY SUPERIOR OBJECT CLASS  
 oRepeater AND SUBCLASSES;  
 WITH ATTRIBUTE aGroupID;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 nameBinding(6) groupName(7)};

**30A.6.2 Group attributes****aGroupID****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bGroupID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) groupID(46)};

**bGroupID****BEHAVIOUR**

DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.2.1.1;
------------	---

**aGroupPortCapacity****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bGroupPortCapacity;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) groupPortCapacity(47)};

**bGroupPortCapacity****BEHAVIOUR**

DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.2.1.2;
------------	---

**aPortMap****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.BitString;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bPortMap;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) portMap(48)};

**bPortMap****BEHAVIOUR**

DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.2.1.3;
------------	---

**30A.6.3 Group notifications****nPortMapChange****NOTIFICATION**

BEHAVIOUR	bPortMapChange;
WITH INFORMATION SYNTAX	IEEE802Dot3-MgmtAttributeModule.BitString;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) notification(10) portMapChange(4)};

**bPortMapChange****BEHAVIOUR**

DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.2.2.1;
------------	---



	PRESENT IF	portBurst(18}); The 1000 Mb/s Burst Monitor capability is implemented;
	REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) managedObjectClass(3) repeaterPortObjectClass(5)};
nbPortName		NAME BINDING
	SUBORDINATE OBJECT CLASS	oRepeaterPort;
	NAMED BY SUPERIOR OBJECT CLASS	oGroup AND SUBCLASSES;
	WITH ATTRIBUTE	aPortID;
	REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) nameBinding(6) portName(8)};

### 30A.7.2 Port attributes

#### aPortID

#### ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.OneOfName;
BEHAVIOUR	bPortID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) portID(49)};

#### bPortID

#### BEHAVIOUR

DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.3.1.1;
------------	---

#### aPortAdminState

#### ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. PortAdminState;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bPortAdminState;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) portAdminState(50)};

#### bPortAdminState

#### BEHAVIOUR

DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.3.1.2;
------------	---

#### aAutoPartitionState

#### ATTRIBUTE

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoPartitionState;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bAutoPartition;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) autoPartitionState(51)};

#### bAutoPartition

#### BEHAVIOUR

DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.3.1.3;
------------	---

**aReadableFrames****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bReadableFrames;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) readableFrames(52)};

**bReadableFrames****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.4;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aReadableOctets****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bReadableOctets;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) readableOctets(53)};

**bReadableOctets****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.5;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 min.;

**aFrameCheckSequenceErrors****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bFCSErrors;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) frameCheckSequenceErrors(54)};

**bFCSErrors****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.6;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aAlignmentErrors****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bAlignmentErrors;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) alignmentErrors(55)};

**bAlignmentErrors****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.7;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;



**aFramesTooLong**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

**ATTRIBUTE**

aCMCounter;  
bFramesTooLong;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) framesTooLong(56)};

**bFramesTooLong**

DEFINED AS

**BEHAVIOUR**

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.8;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 61 days.;

**aShortEvents**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

**ATTRIBUTE**

aCMCounter;  
bShortEvents;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) shortEvents(57)};

**bShortEvents**

DEFINED AS

**BEHAVIOUR**

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.9;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 16 hours;

**aRunts**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

**ATTRIBUTE**

aCMCounter;  
bRunts;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) runts(58)};

**bRunts**

DEFINED AS

**BEHAVIOUR**

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.10;

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 16 h.;

**aCollisions**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

**ATTRIBUTE**

aCMCounter;  
bCollisions;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) collisions(59)};

**bCollisions**

DEFINED AS

**BEHAVIOUR**

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.11;

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 16 h.;

<b>aLateEvents</b>	<b>ATTRIBUTE</b>
DERIVED FROM BEHAVIOUR REGISTERED AS	aCMCounter; bLateEvents; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) lateEvents(60)};
<b>bLateEvents</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.3.1.12;
	NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 81 h.;
<b>aVeryLongEvents</b>	<b>ATTRIBUTE</b>
DERIVED FROM BEHAVIOUR REGISTERED AS	aCMCounter; bVeryLongEvents; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) veryLongEvents(61)};
<b>bVeryLongEvents</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.3.1.13;
	NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 198 days.;
<b>aDataRateMismatches</b>	<b>ATTRIBUTE</b>
DERIVED FROM BEHAVIOUR REGISTERED AS	aCMCounter; bDataRateMismatches; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) dataRateMismatches(62)};
<b>bDataRateMismatches</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.3.1.14;
<b>aAutoPartitions</b>	<b>ATTRIBUTE</b>
DERIVED FROM BEHAVIOUR REGISTERED AS	aCMCounter; bAutoPartitions; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) autoPartitions(63)};
<b>bAutoPartitions</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.4.3.1.15;
<b>aIsolates</b>	<b>ATTRIBUTE</b>
DERIVED FROM BEHAVIOUR REGISTERED AS	aCMCounter; bIsolates; {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)}

attribute(7) isolates(64));

**bIsolates****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.16;

**aSymbolErrorDuringPacket****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bSymbolErrorDuringPacket;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) symbolErrorDuringPacket(65)};

**bSymbolErrorDuringPacket****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.17;

**aLastSourceAddress****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX  
MATCHES FOR  
BEHAVIOUR  
REGISTERED AS

IEEE802CommonDefinitions.MACAddress;  
EQUALITY;  
bLastSourceAddress;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) lastSourceAddress(66)};

**bLastSourceAddress****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.18;

**aSourceAddressChanges****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bSourceAddressChanges;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) sourceAddressChanges(67)};

**bSourceAddressChanges****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.19;

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 81 h.;

**aBursts****ATTRIBUTE**

DERIVED FROM  
BEHAVIOUR  
REGISTERED AS

aCMCounter;  
bBursts;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) bursts(100)};

**bBursts****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.4.3.1.20;

**30A.7.3 Port actions****acPortAdminControl****ACTION**

BEHAVIOUR  
WITH INFORMATION SYNTAX

bPortAdminControl;  
IEEE802Dot3-MgmtAttributeModule.  
PortAdminState;

REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
action(9) portAdminControl(8)};

**bPortAdminControl****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.3.2.1;

**30A.8 MAU managed object class****30A.8.1 MAU, formal definition****oMAU****MANAGED OBJECT CLASS**

DERIVED FROM

“CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :  
1992”:top;

CHARACTERIZED BY

pMAUBasic  
ATTRIBUTES

PACKAGE  
aMAUID GET,  
aMAUType GET-SET,  
aMAUTypeList GET,  
aMediaAvailable GET,  
aJabber GET,  
aMAUAdminState GET;  
nJabber;

NOTIFICATIONS

;

CONDITIONAL PACKAGES

pMAUControl  
ACTIONS

PACKAGE  
acResetMAU,  
acMAUAdminControl;  
{iso(1) member-body(2) us(840) ieee802dot3(10006)  
csmacdmgt(30)  
package(4) mauControlPkg(13)};  
The pMAUControl package is implemented.;

REGISTERED AS

PRESENT IF

pMediaLossTracking  
ATTRIBUTES  
REGISTERED AS

PACKAGE  
aLoseMediaCounter GET;  
{iso(1) member-body(2) us(840) ieee802dot3(10006)  
csmacdmgt(30)  
package(4) mediaLossTrackingPkg(14)};  
MAU TypeValue = AUI or if the  
pMediaLossTracking package is implemented.;

PRESENT IF

pBroadbandDTEMAU  
ATTRIBUTES

PACKAGE  
aBbMAUXmitRcvSplitType GET,

REGISTERED AS	aBroadbandFrequencies	GET;
	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) package(4) broadbandMAUPkg(15)};	
PRESENT IF	The MAU is of type 10BROAD36.;	
p100MbpsMonitor	PACKAGE	
ATTRIBUTES	aFalseCarriers	GET;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) package(4) mau100MbpsMonitor(16)};	
PRESENT IF	The MAU is capable of 100 Mb/s operation.;	
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) managedObjectClass(3) mauObjectClass(6)};	

**nbMAU-repeaterName****NAME BINDING**

SUBORDINATE OBJECT CLASS	oMAU;
NAMED BY SUPERIOR OBJECT CLASS	--(of oRepeaterPort) oRepeaterPort AND SUBCLASSES; --{1.2.840.10006.30.3.5}
WITH ATTRIBUTE	aMAUID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) nameBinding(6) mau-repeaterName(9)};

**nbMAU-dteName****NAME BINDING**

SUBORDINATE OBJECT CLASS	oMAU;
NAMED BY SUPERIOR OBJECT CLASS	--(of oPHYEntity) oPHYEntity AND SUBCLASSES --{1.2.840.10006.30.3.2};
WITH ATTRIBUTE	aMAUID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) nameBinding(6) mau-dteName(10)};

**30A.8.2 MAU attributes****aMAUID****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bMAUID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) mauID(68)};

**bMAUID****BEHAVIOUR**

DEFINED AS	See "BEHAVIOUR DEFINED AS" in 30.5.1.1.1;
------------	---

<b>aMAUType</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.TypeValue;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bMAUType;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) mauType(69)};
<b>bMAUType</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.5.1.1.2;
<b>aMAUTypeList</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.TypeList;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bMAUTypeList;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) mauTypeList(70)};
<b>bMAUTypeList</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.5.1.1.3;
<b>aMediaAvailable</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.
	MediaAvailState;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bMediaAvailable;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) mauMediaAvailable(71)};
<b>bMediaAvailable</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.5.1.1.4;
<b>aLoseMediaCounter</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.aCMCounter;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bLoseMediaCounter;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) mauLoseMediaCounter(72)};
<b>bLoseMediaCounter</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.5.1.1.5;
<b>aJabber</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.Jabber;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bJabberAttribute;

REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) jabber(73)};
<b>bJabberAttribute</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.5.1.1.6;
<b>aMAUAdminState</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.AdminState;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bMAUAdminState;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) mauAdminState(74)};
<b>bMAUAdminState</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.5.1.1.7;
<b>aBbMAUXmitRcvSplitType</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.
MATCHES FOR	BbandXmitRcvSplitType;
BEHAVIOUR	EQUALITY;
REGISTERED AS	bBbMAUXmitRcvSplitType;
	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) bBandSplitType(75)};
<b>bBbMAUXmitRcvSplitType</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.5.1.1.8;
<b>aBroadbandFrequencies</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.
MATCHES FOR	BbandFrequency;
BEHAVIOUR	EQUALITY;
REGISTERED AS	bBroadbandFrequencies;
	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) bBandFrequencies(76)};
<b>bBroadbandFrequencies</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.5.1.1.9;
<b>aFalseCarriers</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.aCMCounter;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bFalseCarriers;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) falseCarriers(77)};

**bFalseCarriers****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.1.10;

**aIdleErrorCount****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.RegisterEight;  
 BEHAVIOUR bIdleErrorCount;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) idleErrorCount(91)};

**bIdleErrorCount****BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.11;

**30A.8.3 MAU actions****acResetMAU****ACTION**

BEHAVIOUR bResetMAU;  
 MODE CONFIRMED;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 action(9) resetMAU(9)};

**bResetMAU****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.2.1;

**acMAUAdminControl****ACTION**

BEHAVIOUR bMAUAdminControl;  
 WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.AdminState;  
 MODE CONFIRMED;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 action(9) mauAdminCtrl(10)};

**bMAUAdminControl****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.2.2;

**30A.8.4 MAU notifications****nJabber****NOTIFICATION**

BEHAVIOUR bJabberNotification;  
 WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.Jabber;  
 ;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 notification(10) jabber(5)};

**bJabberNotification****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.3.1;



## 30A.9 AutoNegotiation managed object class

### 30A.9.1 AutoNegotiation, formal definition

**oAutoNegotiation** **MANAGED OBJECT CLASS**

DERIVED FROM “CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992”:top;

CHARACTERIZED BY

**pAutoNeg**

**ATTRIBUTES**

**PACKAGE**

**aAutoNegID** GET,

**aAutoNegAdminState** GET,

**aAutoNegRemoteSignaling** GET,

**aAutoNegAutoConfig** GET-SET,

**aAutoNegLocalTechnologyAbility** GET,

**aAutoNegAdvertisedTechnologyAbility** GET-SET,

**aAutoNegReceivedTechnologyAbility** GET,

**aAutoNegLocalSelectorAbility** GET,

**aAutoNegAdvertisedSelectorAbility** GET-SET,

**aAutoNegReceivedSelectorAbility** GET;

**ACTIONS**

**acAutoNegRestartAutoConfig**,

**acAutoNegAdminControl**;

        ;

    ;

REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) managedObjectClass(3) autoNegObjectClass(7)};

**nbAutoNeg-mauName** **NAME BINDING**

SUBORDINATE OBJECT CLASS oMAU;

NAMED BY SUPERIOR OBJECT CLASS --(of oMAU)

oMAU AND SUBCLASSES;

--{1.2.840.10006.30.3.6}

WITH ATTRIBUTE aMAUID;

REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) nameBinding(6) autoNeg-mauName(11)};

### 30A.9.2 Auto-Negotiation attributes

**aAutoNegID** **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;

MATCHES FOR EQUALITY;

BEHAVIOUR bAutoNegID;

REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) autoNegID(78)};

**bAutoNegID** **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.6.1.1.1;

<b>aAutoNegAdminState</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegAdminState;
MATCHES FOR BEHAVIOUR	EQUALITY; bAutoNegAdminState;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) autoNegAdminState(79)};
<b>bAutoNegAdminState</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.6.1.1.2;
<b>aAutoNegRemoteSignaling</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegRemoteSignalingDetect;
MATCHES FOR BEHAVIOUR	EQUALITY; bAutoNegRemoteSignaling;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) autoNegRemoteSignaling(80)};
<b>bAutoNegRemoteSignaling</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.6.1.1.3;
<b>aAutoNegAutoConfig</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegAutoConfig;
MATCHES FOR BEHAVIOUR	EQUALITY; bAutoNegAutoConfig;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) autoNegAutoConfig(81)};
<b>bAutoNegAutoConfig</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.6.1.1.4;
<b>aAutoNegLocalTechnologyAbility</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegTechnologyList;
MATCHES FOR BEHAVIOUR	EQUALITY, ORDERING; bAutoNegLocalTechnologyAbility;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) autoNegLocalTechnologyAbility(82)};
<b>bAutoNegLocalTechnologyAbility</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.6.1.1.5;

<b>aAutoNegAdvertisedTechnologyAbility</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegTechnologyList;
MATCHES FOR BEHAVIOUR	EQUALITY, ORDERING; bAutoNegAdvertisedTechnologyAbility;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) autoNegAdvertisedTechnologyAbility(83)};
<b>bAutoNegAdvertisedTechnologyAbility</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.6.1.1.6;
<b>aAutoNegReceivedTechnologyAbility</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegTechnologyList;
MATCHES FOR BEHAVIOUR	EQUALITY, ORDERING; bAutoNegReceivedTechnologyAbility;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) autoNegReceivedTechnologyAbility(84)};
<b>bAutoNegReceivedTechnologyAbility</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.6.1.1.7;
<b>aAutoNegLocalSelectorAbility</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegSelectorList;
MATCHES FOR BEHAVIOUR	EQUALITY, ORDERING; bAutoNegLocalSelectorAbility;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) autoNegLocalSelectorAbility(85)};
<b>bAutoNegLocalSelectorAbility</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.6.1.1.8;
<b>aAutoNegAdvertisedSelectorAbility</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule. AutoNegSelectorList;
MATCHES FOR BEHAVIOUR	EQUALITY, ORDERING; bAutoNegAdvertisedSelectorAbility;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) autoNegAdvertisedSelectorAbility(86)};
<b>bAutoNegAdvertisedSelectorAbility</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.6.1.1.9;

**aAutoNegReceivedSelectorAbility****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX

IEEE802Dot3-MgmtAttributeModule.

MATCHES FOR  
BEHAVIOURAutoNegSelectorList;  
EQUALITY, ORDERING;

REGISTERED AS

bAutoNegReceivedSelectorAbility;  
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) autoNegReceivedSelectorAbility(87)};**bAutoNegReceivedSelectorAbility BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.6.1.1.10;

**30A.9.3 AutoNegotiation actions****acAutoNegRestartAutoConfig****ACTION**BEHAVIOUR  
MODEbAutoNegRestartAutoConfig;  
CONFIRMED;

REGISTERED AS

{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
action(9) autoNegRestartAutoConfig(11)};**bAutoNegRestartAutoConfig****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.6.1.2.1;

**acAutoNegAdminControl****ACTION**BEHAVIOUR  
WITH INFORMATION SYNTAXbAutoNegAdminControl;  
IEEE802Dot3-MgmtAttributeModule.

MODE

AutoNegAdminState;  
CONFIRMED;

REGISTERED AS

{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
action(9) autoNegAdminCtrl(12)};**bAutoNegAdminControl****BEHAVIOUR**

DEFINED AS

See “BEHAVIOUR DEFINED AS” in 30.6.1.2.2;

**30A.10 ResourceTypeID managed object class****30A.10.1 ResourceTypeID, formal definition**

- Implementation of this managed object in accordance with the definition contained in IEEE Std 802.1F-1993<sup>™</sup> is a conformance requirement of this standard.
- NOTE—A single instance of the Resource Type ID managed object exists within the oMACEntity managed object class, a single instance of the Resource Type ID managed object exists within the oRepeater managed object class, and a single instance of the Resource Type ID managed object exists within the oMAU managed object class conditional on the presence of an MII.
- The managed object itself is contained in IEEE Std 802.1F-1993<sup>™</sup>, therefore only name bindings appear in this standard;

**nbResourceTypeID-mac****NAME BINDING**

SUBORDINATE OBJECT CLASS      “IEEE802.1F”:oResourceTypeID;  
 NAMED BY SUPERIOR OBJECT CLASS  
    oMACEntity;  
 WITH ATTRIBUTE                    “IEEE802.1F”:aResourceTypeIDName;  
 REGISTERED AS                    {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
    nameBinding(6) resourceTypeID-mac(12)};

**nbResourceTypeID-repeater****NAME BINDING**

SUBORDINATE OBJECT CLASS      “IEEE802.1F”:oResourceTypeID;  
 NAMED BY SUPERIOR OBJECT CLASS  
    oRepeater AND SUBCLASSES;  
 WITH ATTRIBUTE                    “IEEE802.1F”:aResourceTypeIDName;  
 REGISTERED AS                    {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
    nameBinding(6) resourceTypeID-repeater(13)};

**nbResourceTypeID-mau****NAME BINDING**

SUBORDINATE OBJECT CLASS      “IEEE802.1F”:oResourceTypeID;  
 NAMED BY SUPERIOR OBJECT CLASS  
    oMAU AND SUBCLASSES;  
 WITH ATTRIBUTE                    “IEEE802.1F”:aResourceTypeIDName;  
 REGISTERED AS                    {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
    nameBinding(6) resourceTypeID-mau(14)};

**30A.11 Aggregator managed object class****30A.11.1 Aggregator, formal definition****oAggregator****MANAGED OBJECT CLASS**

DERIVED FROM                      “CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992”:top;  
 CHARACTERIZED BY  
    pAggregatorBasic      PACKAGE  
    ATTRIBUTES            aAggID                    GET;  
 ;  
 ;  
 CONDITIONAL PACKAGES  
    pAggregatorMandatory    PACKAGE  
    ATTRIBUTES            aAggDescription        GET,  
                               aAggName                GET-REPLACE,  
                               aAggActorSystemID     GET-REPLACE,  
                               aAggActorSystemPriority GET-REPLACE,  
                               aAggAggregateOrIndividual GET,  
                               aAggActorAdminKey     GET-REPLACE,  
                               aAggActorOperKey       GET,  
                               aAggMACAddress        GET,  
                               aAggPartnerSystemID    GET,  
                               aAggPartnerSystemPriority GET,  
                               aAggPartnerOperKey     GET,  
                               aAggAdminState         GET-REPLACE,

```

aAggOperState GET,
aAggTimeOfLastOperChange GET,
aAggDataRate GET,
aAggFramesTxOK GET,
aAggFramesRxOK GET,
aAggLinkUpDownNotificationEnable GET-REPLACE,
aAggCollectorMaxDelay GET-REPLACE;
NOTIFICATIONS
nAggLinkUpNotification,
nAggLinkDownNotification;
REGISTERED AS
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacd-
mgt(30) package(4) pAggregatorMandatory(19)};
PRESENT IF Conformance to Link Aggregation management is desired;
pAggregatorRecommendedPACKAGE
ATTRIBUTES aAggOctetsTxOK GET,
aAggOctetsRxOK GET,
aAggFramesDiscardedOnTx GET,
aAggFramesDiscardedOnRx GET,
aAggFramesWithTxErrors GET,
aAggFramesWithRxErrors GET,
aAggUnknownProtocolFrames GET,
aAggPortList GET;
REGISTERED AS
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacd-
mgt(30) package(4) pAggregatorRecommended(20)};
PRESENT IF The recommended package is implemented;
pAggregatorOptional PACKAGE
ATTRIBUTES aAggMulticastFramesTxOK GET,
aAggMulticastFramesRxOK GET,
aAggBroadcastFramesTxOK GET,
aAggBroadcastFramesRxOK GET;
REGISTERED AS
{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacd-
mgt(30) package(4) pAggregatorOptional(21)};
PRESENT IF The optional package is implemented;
;
REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) manage-
dObjectClass(3) oAggregator(10)};

```

**nbAggregatorName****NAME BINDING**

```

SUBORDINATE OBJECT CLASS oAggregator;
NAMED BY SUPERIOR OBJECT CLASS "ISO/IEC 10165-2":system;
WITH ATTRIBUTE aAggID
REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) name-
Binding(6) nbAggregatorName(20)};

```

**30A.11.2 Aggregator attributes****aAggID****ATTRIBUTE**

```

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AgID;

```

MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggID;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggID(101)};

**bAggID** **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.1;

**aAggDescription** **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.DescriptionString;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggDescription;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggDescription(102)};

**bAggDescription** **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.2;

**aAggName** **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.DescriptionString;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggName;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggName(103)};

**bAggName** **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.3;

**aAggActorSystemID** **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MACAddress;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggActorSystemID;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggActorSystemID(104)};

**bAggActorSystemID** **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.4;

**aAggActorSystemPriority** **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PriorityValue;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggActorSystemPriority;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggActorSystemPriority(105)};

**bAggActorSystemPriority****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.5;

**aAggAggregateOrIndividual****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AggOrInd;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggAggregateOrIndividual;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggAggregateOrIndividual(106)};

**bAggAggregateOrIndividual****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.6;

**aAggActorAdminKey****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.KeyValue;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggActorAdminKey;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggActorAdminKey(107)};

**bAggActorAdminKey****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.7;

**aAggActorOperKey****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.KeyValue;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggActorOperKey;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggActorOperKey(108)};

**bAggActorOperKey****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.8;

**aAggMACAddress****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MACAddress;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggMACAddress;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggMACAddress(109)};

**bAggMACAddress****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.9;



<b>aAggPartnerSystemID</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.MACAddress;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bAggPartnerSystemID;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPartnerSystemID(110)};
<b>bAggPartnerSystemID</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.7.1.1.10;
<b>aAggPartnerSystemPriority</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.PriorityValue;
MATCHES FOR	EQUALITY, ORDERING;
BEHAVIOUR	bAggPartnerSystemPriority;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPartnerSystemPriority(111)};
<b>bAggPartnerSystemPriority</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.7.1.1.11;
<b>aAggPartnerOperKey</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.KeyValue;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bAggPartnerOperKey;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPartnerOperKey(112)};
<b>bAggPartnerOperKey</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.7.1.1.12;
<b>aAggAdminState</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.AggState;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bAggAdminState;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggAdminState(113)};
<b>bAggAdminState</b>	<b>BEHAVIOUR</b>
DEFINED AS	See “BEHAVIOUR DEFINED AS” in 30.7.1.1.13;
<b>aAggOperState</b>	<b>ATTRIBUTE</b>
WITH ATTRIBUTE SYNTAX	IEEE802Dot3-MgmtAttributeModule.AggState;
MATCHES FOR	EQUALITY;
BEHAVIOUR	bAggOperState ;

REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) aAggOperState(114)};

**bAggOperState** **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.14;

**aAggTimeOfLastOperChange** **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer32;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR bAggTimeOfLastOperChange;  
REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) aAggTimeOfLastOperChange(115)};

**bAggTimeOfLastOperChange** **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.15;

**aAggDataRate** **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AggDataRate;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR bAggDataRate;  
REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) aAggDataRate(116)};

**bAggDataRate** **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.16;

**aAggOctetsTxOK** **ATTRIBUTE**

DERIVED FROM aCMCounter;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR bAggOctetsTxOK;  
REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) aAggOctetsTxOK(117)};

**bAggOctetsTxOK** **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.17.

NOTE—This counter has a maximum increment rate of 1 230 000 counts per second at 10 Mb/s.;

**aAggOctetsRxOK** **ATTRIBUTE**

DERIVED FROM aCMCounter;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR bAggOctetsRxOK;  
REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) aAggOctetsRxOK(118)};

**bAggOctetsRxOK****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.18.

NOTE—This counter has a maximum increment rate of 1 230 000 counts per second at 10 Mb/s.;

**aAggFramesTxOK****ATTRIBUTE**

DERIVED FROM aCMCounter;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR bAggFramesTxOK;  
REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) aAggFramesTxOK(119)};

**bAggFramesTxOK****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.19.

NOTE—This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s.;

**aAggFramesRxOK****ATTRIBUTE**

DERIVED FROM aCMCounter;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR bAggFramesRxOK;  
REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) aAggFramesRxOK(120)};

**bAggFramesRxOK****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.20.

NOTE—This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s.;

**aAggMulticastFramesTxOK****ATTRIBUTE**

DERIVED FROM aCMCounter;  
MATCHES FOR EQUALITY, ORDERING;  
BEHAVIOUR bAggMulticastFramesTxOK;  
REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
attribute(7) aAggMulticastFramesTxOK(121)};

**bAggMulticastFramesTxOK****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.21.

NOTE—This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s.;

**aAggMulticastFramesRxOK****ATTRIBUTE**

DERIVED FROM aCMCounter;

MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggMulticastFramesRxOK;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggMulticastFramesRxOK(122)};

**bAggMulticastFramesRxOK BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.22.

NOTE—This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s.;

**aAggBroadcastFramesTxOK ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggBroadcastFramesTxOK;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggBroadcastFramesTxOK(123)};

**bAggBroadcastFramesTxOK BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.23.

NOTE—This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s.;

**aAggBroadcastFramesRxOK ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggBroadcastFramesRxOK;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggBroadcastFramesRxOK(124)};

**bAggBroadcastFramesRxOK BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.24.

NOTE—This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s.;

**aAggFramesDiscardedOnTx ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggFramesDiscardedOnTx;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggFramesDiscardedOnTx(125)};

**bAggFramesDiscardedOnTx BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.25.

NOTE—This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s.;

**aAggFramesDiscardedOnRx****ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggFramesDiscardedOnRx;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggFramesDiscardedOnRx(126)};

**bAggFramesDiscardedOnRx****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.26.

NOTE—This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s.;

**aAggFramesWithTxErrors****ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggFramesWithTxErrors;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggFramesWithTxErrors(127)};

**bAggFramesWithTxErrors****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.27.

NOTE—This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s.;

**aAggFramesWithRxErrors****ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggFramesWithRxErrors;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggFramesWithRxErrors(128)};

**bAggFramesWithRxErrors****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.1.1.28.

NOTE—This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s.;

**aAggUnknownProtocolFrames****ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggUnknownProtocolFrames;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggUnknownProtocolFrames(129)};

**bAggUnknownProtocolFrames**      **BEHAVIOUR**

DEFINED AS      See “BEHAVIOUR DEFINED AS” in 30.7.1.1.29.

NOTE—This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s.;

**aAggLinkUpDownNotificationEnable**      **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX      IEEE802Dot3-MgmtAttributeModule.NotificationEnable;  
 MATCHES FOR      EQUALITY;  
 BEHAVIOUR      bAggLinkUpDownNotificationEnable;  
 REGISTERED AS      {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
                          attribute(7) aAggLinkUpDownNotificationEnable(130)};

**bAggLinkUpDownNotificationEnable**      **BEHAVIOUR**

DEFINED AS      See “BEHAVIOUR DEFINED AS” in 30.7.1.1.31;

**aAggPortList**      **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX      IEEE802Dot3-MgmtAttributeModule.AggPortList;  
 MATCHES FOR      EQUALITY;  
 BEHAVIOUR      bAggPortList;  
 REGISTERED AS      {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
                          attribute(7) aAggPortList(131)};

**bAggPortList**      **BEHAVIOUR**

DEFINED AS      See “BEHAVIOUR DEFINED AS” in 30.7.1.1.30;

**aAggCollectorMaxDelay**      **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX      IEEE802Dot3-MgmtAttributeModule.CollectorMaxDelay;  
 MATCHES FOR      EQUALITY;  
 BEHAVIOUR      bAggCollectorMaxDelay;  
 REGISTERED AS      {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
                          attribute(7) aAggCollectorMaxDelay(132)};

**bAggCollectorMaxDelay**      **BEHAVIOUR**

DEFINED AS      See “BEHAVIOUR DEFINED AS” in 30.7.1.1.32;

**30A.11.3 Aggregator notifications****nAggLinkUpNotification**      **NOTIFICATION**

BEHAVIOUR      bAggLinkUpNotification;  
 WITH INFORMATION SYNTAX      IEEE802Dot3-MgmtAttributeModule.AggID;  
 REGISTERED AS      {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) notifica-  
                          tion(10) nAggLinkUpNotification(6)};



REGISTERED AS  
   {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacd-  
 mgt(30) package(4) pAggregationPortMandatory(22)};  
 PRESENT IF   Conformance to Link Aggregation management is desired;  
 ;  
 REGISTERED AS   {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) manage-  
 dObjectClass(3) oAggregationPort(11)};

**nbAggregationPort****NAME BINDING**

SUBORDINATE OBJECT CLASS           oAggregationPort;  
 NAMED BY SUPERIOR OBJECT CLASS   oAggregator;  
 WITH ATTRIBUTE                       aAggPortID;  
 REGISTERED AS   {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) name-  
 Binding(6) nbAggregationPortName(21)};

**30A.12.2 Aggregation Port attributes****aAggPortID****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.AggPortID;  
 MATCHES FOR                         EQUALITY;  
 BEHAVIOUR                            bAggPortID;  
 REGISTERED AS   {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortID(133)};

**bAggPortID****BEHAVIOUR**

DEFINED AS       See “BEHAVIOUR DEFINED AS” in 30.7.2.1.1;

**aAggPortActorSystemPriority****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.PriorityValue;  
 MATCHES FOR                         EQUALITY, ORDERING;  
 BEHAVIOUR                            bAggPortActorSystemPriority;  
 REGISTERED AS   {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortActorSystemPriority(134)};

**bAggPortActorSystemPriority****BEHAVIOUR**

DEFINED AS       See “BEHAVIOUR DEFINED AS” in 30.7.2.1.2;

**aAggPortActorSystemID****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.MACAddress;  
 MATCHES FOR                         EQUALITY, ORDERING;  
 BEHAVIOUR                            bAggPortActorSystemID;  
 REGISTERED AS   {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortActorSystemID(135)};

**bAggPortActorSystemID****BEHAVIOUR**

DEFINED AS       See “BEHAVIOUR DEFINED AS” in 30.7.2.1.3;



**aAggPortActorAdminKey****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.KeyValue;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortActorAdminKey;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortActorAdminKey(136)};

**bAggPortActorAdminKey****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.4;

**aAggPortActorOperKey****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.KeyValue;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortActorOperKey;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortActorOperKey(137)};

**bAggPortActorOperKey****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.5;

**aAggPortPartnerAdminSystemPriority** **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PriorityValue;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortPartnerAdminSystemPriority;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortPartnerAdminSystemPriority(138)};

**bAggPortPartnerAdminSystemPriority** **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.6;

**aAggPortPartnerOperSystemPriority** **ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PriorityValue;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortPartnerOperSystemPriority;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortPartnerOperSystemPriority(139)};

**bAggPortPartnerOperSystemPriority** **BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.7;

**aAggPortPartnerAdminSystemID****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MACAddress;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortPartnerAdminSystemID;

REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPortPartnerAdminSystemID(140)};

**bAggPortPartnerAdminSystemID BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.8;

**aAggPortPartnerOperSystemID ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MACAddress;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR baAggPortPartnerOperSystemID;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPortPartnerOperSystemID(141)};

**bAggPortPartnerOperSystemID BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.9;

**aAggPortPartnerAdminKey ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Key Value;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortPartnerAdminKey;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPortPartnerAdminKey(142)};

**bAggPortPartnerAdminKey BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.10;

**aAggPortPartnerOperKey ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Key Value;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortPartnerOperKey;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPortPartnerOperKey(143)};

**bAggPortPartnerOperKey BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.11;

**aAggPortSelectedAggID ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AggID;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortSelectedAggID;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPortSelectedAggID(144)};

**bAggPortSelectedAggID BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.12;

**aAggPortAttachedAggID****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AggID;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortAttachedAggID;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortAttachedAggID(145)};

**bAggPortAttachedAggID****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.13;

**aAggPortActorPort****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PortNumber;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortActorPort;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortActorPort(146)};

**bAggPortActorPort****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.14;

**aAggPortActorPortPriority****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PriorityValue;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortActorPortPriority;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortActorPortPriority(147)};

**bAggPortActorPortPriority****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.15;

**aAggPortPartnerAdminPort****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PortNumber;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortPartnerAdminPort;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortPartnerAdminPort(148)};

**bAggPortPartnerAdminPort****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.16;

**aAggPortPartnerOperPort****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PortNumber;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortPartnerOperPort;

REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPortPartnerOperPort(149)};

**bAggPortPartnerOperPort BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.17;

**aAggPortPartnerAdminPortPriority ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PriorityValue;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortPartnerAdminPortPriority;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPortPartnerAdminPortPriority(150)};

**bAggPortPartnerAdminPortPriority BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.18;

**aAggPortPartnerOperPortPriority ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PriorityValue;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortPartnerOperPortPriority;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPortPartnerOperPortPriority(151)};

**bAggPortPartnerOperPortPriority BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.19;

**aAggPortActorAdminState ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AggPortState;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortActorAdminState;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPortActorAdminState(152)};

**bAggPortActorAdminState BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.20;

**aAggPortActorOperState ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AggPortState;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortActorOperState;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) aAggPortActorOperState(153)};

**bAggPortActorOperState BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.21;

**aAggPortPartnerAdminState****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AggPortState;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortPartnerAdminState;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortPartnerAdminState(154)};

**bAggPortPartnerAdminState****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.22;

**aAggPortPartnerOperState****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AggPortState;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortPartnerOperState;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortPartnerOperState(155)};

**bAggPortPartnerOperState****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.23;

**aAggPortAggregateOrIndividual****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AggOrInd;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortAggregateOrIndividual;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortAggregateOrIndividual(156)};

**bAggPortAggregateOrIndividual****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.2.1.24;

**30A.13 Aggregation Port Statistics managed object class****30A.13.1 Aggregation Port Statistics, formal definition****oAggPortStats****MANAGED OBJECT CLASS**

DERIVED FROM “CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992”:top;  
 CONDITIONAL PACKAGES

pAggPortStats

ATTRIBUTES

PACKAGE

aAggPortStatsID GET,

aAggPortStatsLACPDUrx GET,

aAggPortStatsMarkerPDUrx GET,

aAggPortStatsMarkerResponsePDUrx GET,

aAggPortStatsUnknownRx GET,

aAggPortStatsIllegalRx GET,

aAggPortStatsLACPDUtx GET,

aAggPortStatsMarkerPDUsTx GET;  
 aAggPortStatsMarkerResponsePDUsTx GET;  
 REGISTERED AS  
   {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacd-  
 mgt(30) package(4) pAggPortStats(23)};  
 PRESENT IF The Aggregation Port Statistics package is supported;  
 ;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) manage-  
 dObjectClass(3) oAggPortStats(12)};

**nbAggPortStats****NAME BINDING**

SUBORDINATE OBJECT CLASS oAggPortStats;  
 NAMED BY SUPERIOR OBJECT CLASS oAggregationPort;  
 WITH ATTRIBUTE aAggPortStatsID;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) name-  
 Binding(6) nbAggPortStats(22)};

**30A.13.2 Aggregation Port Statistics attributes****aAggPortStatsID****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AggregationPortID;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortStatsID;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortStatsID(157)};

**bAggPortStatsID****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.3.1.1;

**aAggPortStatsLACPDUUsRx****ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortStatsLACPDUUsRx;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortStatsLACPDUUsRx(158)};

**bAggPortStatsLACPDUUsRx****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.3.1.2.

NOTE—This counter has a maximum increment rate of 5 counts per second at 10 Mb/s.;

**aAggPortStatsMarkerPDUsRx****ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortStatsMarkerPDUsRx;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortStatsMarkerPDUsRx(159)};

**bAggPortStatsMarkerPDUsRx BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.3.1.3.

NOTE—This counter has a maximum increment rate of 5 counts per second at 10 Mb/s.;

**aAggPortStatsMarkerResponsePDUsRx ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortStatsMarkerResponsePDUsRx;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortStatsMarkerResponsePDUsRx(160)};

**bAggPortStatsMarkerResponsePDUsRx BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.3.1.4.

NOTE—This counter has a maximum increment rate of 5 counts per second at 10 Mb/s.;

**aAggPortStatsUnknownRx ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortStatsUnknownRx;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortStatsUnknownRx(161)};

**bAggPortStatsUnknownRx BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.3.1.5.

NOTE—This counter has a maximum increment rate of 50 counts per second at 10 Mb/s.;

**aAggPortStatsIllegalRx ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortStatsIllegalRx;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortStatsIllegalRx(162)};

**bAggPortStatsIllegalRx BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.3.1.6.

NOTE—This counter has a maximum increment rate of 50 counts per second at 10 Mb/s.;

**aAggPortStatsLACPDUstx ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortStatsLACPDUstx;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortStatsLACPDUstx(163)};

**bAggPortStatsLACPDUstx BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.3.1.7.

NOTE—This counter has a maximum increment rate of 5 counts per second at 10 Mb/s.;

**aAggPortStatsMarkerPDUstx ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortStatsMarkerPDUstx;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortStatsMarkerPDUstx(164)};

**bAggPortStatsMarkerPDUstx BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.3.1.8.

NOTE—This counter has a maximum increment rate of 5 counts per second at 10 Mb/s.;

**aAggPortStatsMarkerResponsePDUstx ATTRIBUTE**

DERIVED FROM aCMCounter;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortStatsMarkerResponsePDUstx;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortStatsMarkerResponsePDUstx(165)};

**bAggPortStatsMarkerResponsePDUstx BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.3.1.9.

NOTE—This counter has a maximum increment rate of 5 counts per second at 10 Mb/s.;

**30A.14 Aggregation Port Debug Information managed object class****30A.14.1 Aggregation Port Debug Information, formal definition****oAggPortDebugInformation MANAGED OBJECT CLASS**

DERIVED FROM “CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992”:top;

CONDITIONAL PACKAGES

pLACPDebug

ATTRIBUTES

PACKAGE

aAggPortDebugInformationID	GET,
aAggPortDebugRxState	GET,
aAggPortDebugLastRxTime	GET,
aAggPortDebugMuxState	GET,
aAggPortDebugMuxReason	GET,
aAggPortDebugActorChurnState	GET,
aAggPortDebugPartnerChurnState	GET,
aAggPortDebugActorChurnCount	GET,
aAggPortDebugPartnerChurnCount	GET,
aAggPortDebugActorSyncTransitionCount	GET,
aAggPortDebugPartnerSyncTransitionCount	GET,



aAggPortDebugActorChangeCount GET;  
 aAggPortDebugPartnerChangeCount GET;  
 REGISTERED AS  
     {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacd-  
 mgt(30) package(4) pAggPortDebugInformation(24)};  
 PRESENT IF The Aggregation Port Debug Information package is sup-  
 ported;  
 ;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) manage-  
 dObjectClass(3) oAggPortDebugInformation(13)};

**nbAggPortDebugInformation****NAME BINDING**

SUBORDINATE OBJECT CLASS oAggPortDebugInformation;  
 NAMED BY SUPERIOR OBJECT CLASS oAggregationPort;  
 WITH ATTRIBUTE aAggPortDebugInformationID  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) name-  
 Binding(6) nbAggPortDebugInformation(23)};

**30A.14.2 Aggregation Port Debug Information attributes****aAggPortDebugInformationID****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AggregationPortID;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortDebugInformationID;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortDebugInformationID(166)};

**bAggPortDebugInformationID****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.4.1.1;

**aAggPortDebugRxState****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.RxState;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortDebugRxState;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortDebugRxState(167)};

**bAggPortDebugRxState****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.4.1.2;

**aAggPortDebugLastRxTime****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Integer32;  
 MATCHES FOR EQUALITY, ORDERING;  
 BEHAVIOUR bAggPortDebugLastRxTime;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortDebugLastRxTime(168)};

**bAggPortDebugLastRxTime****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.4.1.3;

**aAggPortDebugMuxState****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MuxState;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortDebugMuxState;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortDebugMuxState(169)};

**bAggPortDebugMuxState****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.4.1.4;

**aAggPortDebugMuxReason****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.DescriptionString;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortDebugMuxReason;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortDebugMuxReason(170)};

**bAggPortDebugMuxReason****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.4.1.5;

**aAggPortDebugActorChurnState****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.ChurnState;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortDebugActorChurnState;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortDebugActorChurnState(171)};

**bAggPortDebugActorChurnState****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.4.1.6;

**aAggPortDebugPartnerChurnState****ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.ChurnState;  
 MATCHES FOR EQUALITY;  
 BEHAVIOUR bAggPortDebugPartnerChurnState;  
 REGISTERED AS {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
 attribute(7) aAggPortDebugPartnerChurnState(172)};

**bAggPortDebugPartnerChurnState****BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.7.4.1.7;

**aAggPortDebugActorChurnCount      ATTRIBUTE**

DERIVED FROM                      aCMCounter;  
 MATCHES FOR                      EQUALITY, ORDERING;  
 BEHAVIOUR                        bAggPortDebugActorChurnCount;  
 REGISTERED AS    {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
    attribute(7) aAggPortDebugActorChurnCount(173)};

**bAggPortDebugActorChurnCount      BEHAVIOUR**

DEFINED AS            See “BEHAVIOUR DEFINED AS” in 30.7.4.1.8.

NOTE—This counter has a maximum increment rate of 5 counts per second.;

**aAggPortDebugPartnerChurnCount      ATTRIBUTE**

DERIVED FROM                      aCMCounter;  
 MATCHES FOR                      EQUALITY, ORDERING;  
 BEHAVIOUR                        bAggPortDebugPartnerChurnCount;  
 REGISTERED AS    {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
    attribute(7) aAggPortDebugPartnerChurnCount(174)};

**bAggPortDebugPartnerChurnCount      BEHAVIOUR**

DEFINED AS            See “BEHAVIOUR DEFINED AS” in 30.7.4.1.9.

NOTE—This counter has a maximum increment rate of 5 counts per second.;

**aAggPortDebugActorSyncTransitionCount      ATTRIBUTE**

DERIVED FROM                      aCMCounter;  
 MATCHES FOR                      EQUALITY, ORDERING;  
 BEHAVIOUR                        bAggPortDebugActorSyncTransitionCount;  
 REGISTERED AS    {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
    attribute(7) aAggPortDebugActorSyncTransitionCount(175)};

**bAggPortDebugActorSyncTransitionCount      BEHAVIOUR**

DEFINED AS            See “BEHAVIOUR DEFINED AS” in 30.7.4.1.10.

NOTE—This counter has a maximum increment rate of 5 counts per second.;

**aAggPortDebugPartnerSyncTransitionCount      ATTRIBUTE**

DERIVED FROM                      aCMCounter;  
 MATCHES FOR                      EQUALITY, ORDERING;  
 BEHAVIOUR                        bAggPortDebugPartnerSyncTransitionCount;  
 REGISTERED AS    {iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30)  
    attribute(7) aAggPortDebugPartnerSyncTransitionCount(176)};

**bAggPortDebugPartnerSyncTransitionCount      BEHAVIOUR**

DEFINED AS            See “BEHAVIOUR DEFINED AS” in 30.7.4.1.11.



## Annex 30B

(normative)

### GDMO and ASN.1 definitions for management

#### 30B.1 Common attributes template

This template defines generic facilities that are used in the standard.

##### aCMCounter ATTRIBUTE

DERIVED FROM	“ISO/IEC 10165-5”:genericWrappingCounter;
BEHAVIOUR	bCMCounter;
REGISTERED AS	{iso(1) member-body(2) us(840) ieee802dot3(10006) csmacdmgt(30) attribute(7) cmCounter(88)};

##### bCMCounter BEHAVIOUR

DEFINED AS	Wraps at one of two sizes. Size is conditional. Wraps at 32 bits, that is this counter reaches its maximum value at $2^{32}-1$ (i.e., approximately $4.294 \times 10^9$ ) and then rolls over to zero on the next increment, if maximum increment rate from zero causes a rollover in 58 min or more. Wraps at 64 bits, that is this counter reaches its maximum value at $2^{64}-1$ (i.e., approximately $1.844 \dots \times 10^{19}$ ) and then rolls over to zero on the next increment, if maximum increment rate from zero would cause a 32 bit counter to roll over in less than 58 min. The counter that this is derived from initializes to zero. Initialization to zero is not a requirement of this standard;
------------	--

#### 30B.2 ASN.1 module for CSMA/CD managed objects

This ASN.1 module defines the ASN.1 types and subtypes that are referred to immediately after the WITH ATTRIBUTE SYNTAX construct in this clause’s uses of the attribute template defined in ISO/IEC 10165-4: 1992, Guidelines for the definition of managed objects (GDMO).

```
IEEE802Dot3-MgmtAttributeModule {iso(1) member-body(2) us(840) ieee802dot3(10006) global(1)
asn1Module(2) commonDefinitions(0) version(2)} DEFINITIONS IMPLICIT TAGS ::= BEGIN
```

```
EXPORTS--everything
```

```
IMPORTS--implicitly imports ISO 8824: 1990
```

```
MACAddress
  FROM IEEE802CommonDefinitions
  {iso(1) member-body(2) us(840) ieee802dot1partF(10011)
  asn1Module(2) commonDefinitions(0) version1(0)};

AdminState ::= ENUMERATED {
  other (1), --undefined
```

```

unknown          (2),    --initializing, true state not yet known
operational      (3),    --powered and connected
standby          (4),    --inactive but on
shutdown         (5),    --similar to power down
}

```

AggDataRate ::= INTEGER (0..2<sup>32</sup>-1)--The data rate of an Aggregation

AggID ::= INTEGER (0..2<sup>32</sup>-1)

AggOrInd ::= BOOLEAN

AggPortID ::= INTEGER (0..2<sup>32</sup>-1)

AggPortList ::= SEQUENCE OF AggPortID

AggPortState ::= BIT STRING (SIZE (1..8))

```

AggState ::= ENUMERATED {
  up          (0),    --operational
  down       (1),    --disabled
}

```

AttemptArray ::= SEQUENCE OF aCMCounter--array [1..attempt limit - 1]

```

AutoNegAdminState ::= ENUMERATED {
  disabled (1),
  enabled  (2)
}

```

```

AutoNegAutoConfig ::= ENUMERATED {
  other          (1),
  configuring    (2),
  complete       (3),
  disabled       (4),
  parallel detect fail (5)
}

```

```

AutoNegRemoteSignalingDetect ::= ENUMERATED {
  detected (1),
  notdetected (2)
}

```

```

AutoNegSelector ::= ENUMERATED {
  other (1),    --undefined
  ethernet (2), --802.3
  isoethernet (3), --802.9
  tokenring (4) --802.5
}

```

AutoNegSelectorList ::= SEQUENCE OF AutoNegSelector

```

AutoNegTechnology ::= ENUMERATED {
    global                (0),    --reserved for future use.
    other                 (1),    --undefined
    unknown              (2),    --initializing, true ability not yet known.
    10BASE-T             (14),   --10BASE-T as defined in Clause 14
    10BASE-TFD           (142),  --Full duplex 10BASE-T as defined in Clauses 14 and 31
    100BASE-T4           (23),   --100BASE-T4 as defined in Clause 23
    100BASE-TX           (25),   --100BASE-TX as defined in Clause 25
    100BASE-TXFD         (252),  --Full duplex 100BASE-TX as defined in Clauses 25 and 31
    FDX PAUSE            (312),  --PAUSE operation for full duplex links as defined in
    Annex 31B
    FDX APAUSE          (313),  --Asymmetric PAUSE operation for full duplex links as defined
    in Clause 37 and Annex 31B
    FDX SPAUSE          (314),  --Symmetric PAUSE operation for full duplex links as defined
    in Clause 37 and Annex 31B
    FDX BPAUSE          (315),  --Asymmetric and Symmetric PAUSE operation for full duplex
    links as defined in Clause 37 and Annex 31B
    100BASE-T2           (32),   --100BASE-T2 as defined in Clause 32
    100BASE-T2FD         (322),  --Full duplex 100BASE-T2 as defined in Clauses 31 and 32
    1000BASE-X           (36),   --1000BASE-X as defined in Clause 36
    1000BASE-XFD        (362),  --Full duplex 1000BASE-X as defined in Clause 36
    1000BASE-T           (40),   --1000BASE-T UTP PHY as defined in Clause 36
    1000BASE-TFD        (402),  --Full duplex 1000BASE-T UTP PHY to be defined in
    Clause 36
    Rem Fault1           (37),   --Remote fault bit 1 (RF1) as specified in Clause 37
    Rem Fault2           (372),  --Remote fault bit 2 (RF1) as specified in Clause 37
    isoethernet          (8029)  --802.9 ISLAN-16T
}

```

AutoNegTechnologyList ::= SEQUENCE OF AutoNegTechnology

```

AutoPartitionState ::= ENUMERATED {
    autoPartitioned      (1),
    notAutoPartitioned  (2)
}

```

```

BbandFrequency ::= SEQUENCE {
    xmitCarrierFrequency [1]  INTEGER , --Frequency in MHz times 4 (250 kHz resolution)
    translationFrequency  [2] INTEGER  --Frequency in MHz times 4 (250 kHz resolution)
}

```

```

BbandXmitRcvSplitType ::= ENUMERATED {
    other                (1),    --undefined
    single               (2),    --single-cable system
    dual                 (3),    --dual-cable system, offset normally zero
}

```

BitString ::= BIT STRING (SIZE (1..1024))

```

ChurnState ::= ENUMERATED {
    noChurn              (0),    --NO_ACTOR/PARTNER_CHURN
    --or ACTOR/PARTNER_CHURN_MONITOR
    churn                (1),    --ACTOR/PARTNER_CHURN
}

```

CollectorMaxDelay ::= INTEGER --16 bit value, tens of microseconds  
--(max = 0.65535 seconds)

DescriptionString ::= PrintableString (SIZE 0..255))

```
DuplexValues ::= ENUMERATED {
    half duplex      (1),  --capable of operating in half duplex mode
    full duplex     (2),  --capable of operating in full duplex mode
    unknown         (3),  --unknown duplex capability
}
```

```
Integer32 ::= INTEGER (0..232-1)  --32 bit value
```

```
Jabber ::= SEQUENCE {
    jabberFlag      [1]    JabberFlag,
    jabberCounter   [2]    JabberCounter
}
```

```
JabberFlag ::= ENUMERATED {
    other          (1),  --undefined
    unknown       (2),  --initializing, true state not yet known
    normal        (3),  --state is true or normal
    fault         (4),  --state is false, fault or abnormal
}
```

```
JabberCounter ::= INTEGER (0..232-1)
```

```
KeyValue ::= INTEGER (0..216-1)  --16 bit value; range 0-65535
```

```
LACPActivity ::= ENUMERATED {
    active        (0),  --Port is Active LACP
    passive       (1),  --Port is Passive LACP
}
```

```
LACPTimeout ::= ENUMERATED {
    short         (0),  --Timeouts are Short
    long          (1),  --Timeouts are Long
}
```

```
LinkDelayAllowance ::= INTEGER (0..232-1)
```

```
MACControlFunctions ::= ENUMERATED {
    PAUSE         (312)  --PAUSE command implemented
}
```

```
MACControlFunctionsList ::= SEQUENCE OF MACControlFunctions
```

```
MACCapabilitiesList ::= SEQUENCE OF DuplexValues
```

```
MauTypeList ::= SEQUENCE OF TypeValue
```

```
MediaAvailState ::= ENUMERATED {
    other          (1),  --undefined
    unknown       (2),  --initializing, true state not yet known
    available     (3),  --link or light normal, loopback normal
    not available (4),  --link loss or low light, no loopback
}
```



```

remote fault          (5), --remote fault with no detail
invalid signal       (6), --invalid signal, applies only to 10BASE-FB
remote jabber        (7), --remote fault, reason known to be jabber
remote link loss     (8), --remote fault, reason known to be far-end link loss
remote test          (9), --remote fault, reason known to be test
offline              (10), --offline, applies only to Clause 37 Auto-Negotiation
auto neg error       (11)  --Auto-Negotiation error, applies only to Clause 37
                        Auto-Negotiation
}

```

```

MIIDetect ::= ENUMERATED {
  unknown          (1),
  presentNothingConnected(2),
  presentConnected (3),
  absent           (4)
}

```

MulticastAddressList ::= SEQUENCE OF MACAddress

```

MuxState ::= ENUMERATED {
  detached          (0), --DETACHED
  waiting           (1), --WAITING
  attached          (2), --ATTACHED
  collecting         (3), --COLLECTING
  distributing      (4), --DISTRIBUTING
  collecting_distributing (5) --COLLECTING_DISTRIBUTING
}

```

```

NotificationEnable ::= ENUMERATED {
  enabled (0), --Notifications enabled
  disabled (1) --Notifications disabled
}

```

OneOfName ::= INTEGER (1..1024)

PhyTypeList ::= SEQUENCE OF PhyTypeValue

```

PhyTypeValue ::= ENUMERATED {
  other          (1), --undefined
  unknown        (2), --initializing, true state or type not yet known
  none           (3), --MII present and nothing connected
  10 Mb/s        (7), --Clause 7 10 Mb/s Manchester
  100BASE-T4     (23), --Clause 23 100 Mb/s 8B/6T
  100BASE-X      (24), --Clause 24 100 Mb/s 4B/5B
  100BASE-T2     (32), --Clause 32 100 Mb/s PAM5x5
  1000BASE-X     (36), --Clause 36 1000 Mb/s 8B/10B
  1000BASE-T     (40), --Clause 40 1000 Mb/s 4D-PAM5
}

```

```

PortAdminState ::= ENUMERATED {
  disabled (1),
  enabled  (2)
}

```

PortNumber ::= INTEGER (0..2<sup>16</sup>-1)

PriorityValue ::= INTEGER (0..2<sup>16</sup>-1) --16 bits

RegisterEight ::= INTEGER (0..255)

RepeaterHealthData ::= OCTET STRING (SIZE (0..255))

```
RepeaterHealthInfo ::= SEQUENCE {
    repeaterHealthState [1] RepeaterHealthState,
    repeaterHealthText  [2] RepeaterHealthText OPTIONAL,
    repeaterHealthData  [3] RepeaterHealthData OPTIONAL
}
```

```
RepeaterHealthState ::= ENUMERATED {
    other          (1), --undefined or unknown
    ok             (2), --no known failures
    repeaterFailure (3), --known to have a repeater-related failure
    groupFailure   (4), --known to have a group-related failure
    portFailure    (5), --known to have a port-related failure
    generalFailure (6)  --has a failure condition, unspecified type
}
```

```
RepeaterType ::= ENUMERATED {
    other          (1), --See 30.2.5:
    unknown        (2), --initializing, true state or type not yet known
    10 Mb/s        (9), --Clause 9 10 Mb/s Baseband repeater
    100 Mb/sClassI (271), --Clause 27 class I 100 Mb/s Baseband repeater
    100 Mb/sClassII (272), --Clause 27 class II 100 Mb/s Baseband repeater
    1000 Mb/s      (41), --Clause 41 1000 Mb/s Baseband repeater
    802.9a         (99)  --Integrated services repeater
}
```

RepeaterHealthText ::= PrintableString (SIZE (0..255))

```
RxState ::= ENUMERATED {
    current          (0),
    expired          (1),
    defaulted        (2),
    initialize       (3),
    lacpDisabled     (4),
    portDisabled     (5)
}
```

TrueFalse ::= BOOLEAN

TypeList ::= SEQUENCE OF TypeValue

```
TypeValue ::= ENUMERATED {
    global          (0), --undefined
    other           (1), --undefined
    unknown        (2), --initializing, true state not yet known
    AUI             (7), --no internal MAU, view from AUI
    10BASE5         (8), --Thick coax MAU as specified in Clause 8
    FOIRL          (9), --FOIRL MAU as specified in 9.9
    10BASE2        (10), --Thin coax MAU as specified in Clause 10
}
```

10BROAD36	(11),	--Broadband DTE MAU as specified in Clause 11
10BASE-T	(14),	--UTP MAU as specified in Clause 14, duplex mode unknown
10BASE-THD	(141),	--UTP MAU as specified in Clause 14, half duplex mode
10BASE-TFD	(142),	--UTP MAU as specified in Clause 14, full duplex mode
10BASE-FP	(16),	--Passive fiber MAU as specified in Clause 16
10BASE-FB	(17),	--Synchronous fiber MAU as specified in Clause 17
10BASE-FL	(18),	--Asynchronous fiber MAU as specified in Clause 18, duplex mode unknown
10BASE-FLHD	(181),	--Asynchronous fiber MAU as specified in Clause 18, half duplex mode
10BASE-FLFD	(182),	--Asynchronous fiber MAU as specified in Clause 18, full duplex mode
100BASE-T4	(23),	--Four-pair Category 3 UTP as specified in Clause 23
100BASE-TX	(25),	--Two-pair Category 5 UTP as specified in Clause 25, duplex mode unknown
100BASE-TXHD	(251),	--Two-pair Category 5 UTP as specified in Clause 25, half duplex mode
100BASE-TXFD	(252),	--Two-pair Category 5 UTP as specified in Clause 25, full duplex mode
100BASE-FX	(26),	--X fiber over PMD as specified in Clause 26, duplex mode unknown
100BASE-FXHD	(261),	--X fiber over PMD as specified in Clause 26, half duplex mode
100BASE-FXFD	(262),	--X fiber over PMD as specified in Clause 26, full duplex mode
100BASE-T2	(32),	--Two-pair Category 3 UTP as specified in Clause 32, duplex mode unknown
100BASE-T2HD	(321),	--Two-pair Category 3 UTP as specified in Clause 32, half duplex mode
100BASE-T2FD	(322),	--Two-pair Category 3 UTP as specified in Clause 32, full duplex mode
1000BASE-X	(36),	--X PCS/PMA as specified in Clause 36 over unknown PMD, duplex mode unknown
1000BASE-XHD	(361),	--X PCS/PMA as specified in Clause 36 over unknown PMD, half duplex mode
1000BASE-XFD	(362),	--X PCS/PMA as specified in Clause 36 over unknown PMD, full duplex mode
1000BASE-LX	(381),	--X fiber over long-wavelength laser PMD as specified in Clause 38, duplex mode unknown
1000BASE-LXHD	(382),	--X fiber over long-wavelength laser PMD as specified in Clause 38, half duplex mode
1000BASE-LXFD	(383),	--X fiber over long-wavelength laser PMD as specified in Clause 38, full duplex mode
1000BASE-SX	(384),	--X fiber over short-wavelength laser PMD as specified in Clause 38, duplex mode unknown
1000BASE-SXHD	(385),	--X fiber over short-wavelength laser PMD as specified in Clause 38, half duplex mode
1000BASE-SXFD	(386),	--X fiber over short-wavelength laser PMD as specified in Clause 38, full duplex mode
1000BASE-CX	(39),	--X copper over 150-Ohm balanced cable PMD as specified in Clause 39, duplex mode unknown
1000BASE-CXHD	(391),	--X copper over 150-Ohm balanced cable PMD as specified in Clause 39, half duplex mode
1000BASE-CXFD	(392),	--X copper over 150-Ohm balanced cable PMD as specified in Clause 39, full duplex mode
1000BASE-T	(40),	--Four-pair Category 5 UTP PHY as specified in Clause 40, duplex mode unknown
1000BASE-THD	(401),	--Four-pair Category 5 UTP PHY as specified in Clause 40, half duplex mode
1000BASE-TFD	(402),	--Four-pair Category 5 UTP PHY as specified in Clause 40, full duplex mode
802.9a	(99)	--Integrated services MAU as specified in IEEE Std 802.9 ISLAN-16T

}

END

## Annex 30C

(normative)

### SNMP MIB definitions for Link Aggregation

#### 30C.1 Introduction

This annex defines a portion of the Management Information Base (MIB) for use with network management protocols in TCP/IP based internets. In particular it defines objects for managing the operation of the Link Aggregation sublayer, based on the specification of Link Aggregation contained in Clause 43. This annex includes an MIB module that is SNMPv2 SMI compliant.

#### 30C.2 The SNMP Management Framework

The SNMP Management Framework presently consists of five major components

- a) An overall architecture, described in RFC 2271.
- b) Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIV1 and is described in RFC 1155, RFC 1212, and RFC 1215. The second version, called SMIV2, is described in RFC 1902, RFC 1903, and RFC 1904.
- c) Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and is described in RFC 1157. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and is described in RFC 1901 and RFC 1906. The third version of the message protocol is called SNMPv3 and is described in RFC 1906, RFC 2272, and RFC 2274.
- d) Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in RFC 1157. A second set of protocol operations and associated PDU formats is described in RFC 1905.
- e) A set of fundamental applications described in RFC 2273 and the view-based access control mechanism described in RFC 2275.

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This annex specifies an MIB module that is compliant to the SMIV2. An MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine-readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine-readable information is not considered to change the semantics of the MIB.

#### 30C.3 Security considerations

There are a number of management objects defined in this MIB that have a MAX-ACCESS clause of readwrite and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment can have a negative effect on network operations.

SNMPv1 by itself is not a secure environment. Even if the network itself is secure (e.g., by using IPSec), there is no control as to who on the secure network is allowed to access (read/change/create/delete) the objects in this MIB.

It is recommended that the implementors consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2274 and the View-based Access Control Model RFC 2275 is recommended. It then becomes a user responsibility to ensure that the SNMP entity giving access to an instance of this MIB is properly configured to give access only to those principals (users) that have legitimate rights to access(read/change/create/delete) them, as appropriate.

## 30C.4 Structure of the MIB

A single MIB module is defined in this annex. Objects in the MIB are arranged into groups. Each group is organized as a set of related objects. The overall structure and assignment of objects to their groups is shown in the following subclauses.

### 30C.4.1 Relationship to the managed objects defined in Clause 30

This subclause contains cross-references to the objects defined in Clause 30. Table 30C–1 contains cross references for the MIB objects defined in this annex. Table 30C–2 contains definitions of ifTable elements, as defined in RFC 2233, for an Aggregator. These table elements are cross referenced to the corresponding definitions in Clause 30.

**Table 30C–1 — Managed object cross reference table**

Definition in Clause 30	MIB object
30.7.1.1.1 aAggID	ifIndex value (see RFC 2233)
30.7.1.1.4 aAggActorSystemID	dot3adAggActorSystemID
30.7.1.1.5 aAggActorSystemPriority	dot3adAggActorSystemPriority
30.7.1.1.6 aAggAggregateOrIndividual	dot3adAggAggregateOrIndividual
30.7.1.1.7 aAggActorAdminKey	dot3adAggActorAdminKey
30.7.1.1.8 aAggActorOperKey	dot3adAggActorOperKey
30.7.1.1.10 aAggPartnerSystemID	dot3adAggPartnerSystemID
30.7.1.1.11 aAggPartnerSystemPriority	dot3adAggPartnerSystemPriority
30.7.1.1.12 aAggPartnerOperKey	dot3adAggPartnerOperKey
30.7.1.1.30 aAggPortList	dot3adAggPortListTable
30.7.1.1.32 aAggCollectorMaxDelay	dot3adAggCollectorMaxDelay
30.7.2.1.1 aAggPortID	ifIndex value (see RFC 2233)
30.7.2.1.2 aAggPortActorSystemPriority	dot3adAggPortActorSystemPriority

**Table 30C–1 — Managed object cross reference table (continued)**

Definition in Clause 30	MIB object
30.7.2.1.3 aAggPortActorSystemID	dot3adAggPortActorSystemID
30.7.2.1.4 aAggPortActorAdminKey	dot3adAggPortActorAdminKey
30.7.2.1.5 aAggPortActorOperKey	dot3adAggPortActorOperKey
30.7.2.1.6 aAggPortPartnerAdminSystemPriority	dot3adAggPortPartnerAdminSystemPriority
30.7.2.1.7 aAggPortPartnerOperSystemPriority	dot3adAggPortPartnerOperSystemPriority
30.7.2.1.8 aAggPortPartnerAdminSystemID	dot3adAggPortPartnerAdminSystemID
30.7.2.1.9 aAggPortPartnerOperSystemID	dot3adAggPortPartnerOperSystemID
30.7.2.1.10 aAggPortPartnerAdminKey	dot3adAggPortPartnerAdminKey
30.7.2.1.11 aAggPortPartnerOperKey	dot3adAggPortPartnerOperKey
30.7.2.1.12 aAggPortSelectedAggID	dot3adAggPortSelectedAggID
30.7.2.1.13 aAggPortAttachedAggID	dot3adAggPortAttachedAggID
30.7.2.1.14 aAggPortActorPort	dot3adAggPortActorPort
30.7.2.1.15 aAggPortActorPortPriority	dot3adAggPortActorPortPriority
30.7.2.1.16 aAggPortPartnerAdminPort	dot3adAggPortPartnerAdminPort
30.7.2.1.17 aAggPortPartnerOperPort	dot3adAggPortPartnerOperPort
30.7.2.1.18 aAggPortPartnerAdminPortPriority	dot3adAggPortPartnerAdminPortPriority
30.7.2.1.19 aAggPortPartnerOperPortPriority	dot3adAggPortPartnerOperPortPriority
30.7.2.1.20 aAggPortActorAdminState	dot3adAggPortActorAdminState
30.7.2.1.21 aAggPortActorOperState	dot3adAggPortActorOperState
30.7.2.1.22 aAggPortPartnerAdminState	dot3adAggPortPartnerAdminState
30.7.2.1.23 aAggPortPartnerOperState	dot3adAggPortPartnerOperState
30.7.2.1.24 aAggPortAggregateOrIndividual	dot3adAggPortAggregateOrIndividual
30.7.3.1.1 aAggPortStatsID	ifIndex value (see RFC 2233) of the port
30.7.3.1.2 aAggPortStatsLACPDU Rx	dot3adAggPortStatsLACPDU Rx
30.7.3.1.3 aAggPortStatsMarkerPDU Rx	dot3adAggPortStatsMarkerPDU Rx
30.7.3.1.4 aAggPortStatsMarkerResponsePDU Rx	dot3adAggPortStatsMarkerResponsePDU Rx
30.7.3.1.5 aAggPortStatsUnknownRx	dot3adAggPortStatsUnknownRx

**Table 30C–1 — Managed object cross reference table (continued)**

Definition in Clause 30	MIB object
30.7.3.1.6 aAggPortStatsIllegalRx	dot3adAggPortStatsIllegalRx
30.7.3.1.7 aAggPortStatsLACPDUstx	dot3adAggPortStatsLACPDUstx
30.7.3.1.8 aAggPortStatsMarkerPDUstx	dot3adAggPortStatsMarkerPDUstx
30.7.3.1.9 aAggPortStatsMarkerResponsePDUstx	dot3adAggPortStatsMarkerResponsePDUstx
30.7.4.1.1 aAggPortDebugInformationID	ifIndex value (see RFC 2233) of the port
30.7.4.1.2 aAggPortDebugRxState	dot3adAggPortDebugRxState
30.7.4.1.3 aAggPortDebugLastRxTime	dot3adAggPortDebugLastRxTime
30.7.4.1.4 aAggPortDebugMuxState	dot3adAggPortDebugMuxState
30.7.4.1.5 aAggPortDebugMuxReason	dot3adAggPortDebugMuxReason
30.7.4.1.6 aAggPortDebugActorChurnState	dot3adAggPortDebugActorChurnState
30.7.4.1.7 aAggPortDebugPartnerChurnState	dot3adAggPortDebugPartnerChurnState
30.7.4.1.8 aAggPortDebugActorChurnCount	dot3adAggPortDebugActorChurnCount
30.7.4.1.9 aAggPortDebugPartnerChurnCount	dot3adAggPortDebugPartnerChurnCount
30.7.4.1.10 aAggPortDebugActorSyncTransitionCount	dot3adAggPortDebugActorSyncTransitionCount
30.7.4.1.11 aAggPortDebugPartnerSyncTransitionCount	dot3adAggPortDebugPartnerSyncTransitionCount
30.7.4.1.12 aAggPortDebugActorChangeCount	dot3adAggPortDebugActorChangeCount
30.7.4.1.13 aAggPortDebugPartnerChangeCount	dot3adAggPortDebugPartnerChangeCount

### 30C.4.2 The Aggregator Group

This group of objects, in combination with the ifTable entry for an Aggregator and the Aggregator Port List, provides the functionality of the Aggregator managed object class (30.7.1). The Aggregator Group provides the control elements necessary to configure an Aggregator, plus the statistical information necessary to monitor the behaviour of an Aggregator.

### 30C.4.3 The Aggregator Port List Group

This group of objects implements the functionality defined for the Aggregator Port List attribute (30.7.1.1.30).

**Table 30C–2—ifTable element definitions for an Aggregator**

Object	Definition
ifIndex	A unique integer value is allocated to each Aggregator by the local System. Interpreted as defined in RFC 2233.
ifDescr	Interpreted as defined in RFC 2233 and as further refined in the definition of aAggDescription (30.7.1.1.2).
ifType	ieee8023adLag(161) <sup>a</sup> .
ifMTU	The largest MAC Client SDU that can be carried by this Aggregator—1500 octets.
ifSpeed	The data rate of the Aggregation as defined for aAggDataRate (30.7.1.1.16).
ifPhysAddress	The individual MAC Address of the Aggregator as defined for aAggMACAddress (30.7.1.1.9).
ifAdminStatus	The administrative state of the Aggregator as defined for aAggAdminState (30.7.1.1.13).
ifOperStatus	The operational state of the Aggregator as defined for aAggOperState (30.7.1.1.14).
ifLastChange	Interpreted as defined in RFC 2233; see also the definition of aAggTimeOfLastOperChange (30.7.1.1.15).
ifInOctets	The total number of user data octets received by the aggregation, as defined for aAggOctetsRxOK (30.7.1.1.18).
ifInUcastPkts	The total number of unicast user data frames received by the aggregation. This value is calculated as the value of aAggFramesRxOK (30.7.1.1.20), less the values of aAggMulticastFramesRxOK (30.7.1.1.22) and aAggBroadcastFramesRxOK (30.7.1.1.24).
ifInNUcastPkts	Deprecated in RFC 2233.
ifInDiscards	The number of frames discarded on reception, as defined for aAggFramesDiscardedOnRx (30.7.1.1.26).
ifInErrors	The number of frames with reception errors, as defined for aAggFramesWithRxErrors (30.7.1.1.28).
ifInUnknownProtos	The number of unknown protocol frames discarded on reception, as defined for aAggUnknownProtocolFrames (30.7.1.1.29).
ifOutOctets	The total number of user data octets transmitted by the aggregation, as defined for aAggOctetsTxOK (30.7.1.1.17).
ifOutUcastPkts	The total number of unicast user data frames transmitted by the aggregation. This value is calculated as the value of aAggFramesTxOK (30.7.1.1.19), less the values of aAggMulticastFramesTxOK (30.7.1.1.21) and aAggBroadcastFramesTxOK (30.7.1.1.23).
ifOutNUcastPkts	Deprecated in RFC 2233.



**Table 30C–2—ifTable element definitions for an Aggregator (continued)**

Object	Definition
ifOutDiscards	The number of frames discarded on transmission, as defined for aAggFramesDiscardedOnTx (30.7.1.1.25).
ifOutErrors	The number of frames discarded due to transmission errors, as defined for aAggFramesWithTxErrors (30.7.1.1.27).
ifOutQLen	Deprecated in RFC 2233. Set to zero if present.
ifSpecific	Deprecated in RFC 2233. Set to { 0.0 } if present.
ifLinkUpDownTrapEnable	See the definition of aAggLinkUpDownNotificationEnable (30.7.1.1.31).
ifConnectorPresent	“FALSE.”
ifHighSpeed	Set to zero.
ifName	The locally assigned textual name of the Aggregator, as defined for aAggName (30.7.1.1.3). Interpreted as defined in RFC 2233.
linkUp TRAP	See the definition of nAggLinkUpNotification (30.7.1.2.1).
linkDown TRAP	See the definition of nAggLinkDownNotification (30.7.1.2.2).

<sup>a</sup>Values of ifType are assigned by the Internet Assigned Numbers Authority (IANA). A directory of number assignments is maintained on their website, at URL: <http://www.iana.org/numbers.html>. The currently assigned ifType values can be found in the SMI Numbers (Network Management Parameters) section of that directory.

### 30C.4.4 The Aggregation Port Group

This group of objects provides the functionality of the Aggregation Port managed object class (30.7.2). The Aggregation Port Group provides the control elements necessary to configure a Port for Link Aggregation, plus the statistical information necessary to monitor the behavior of the port.

### 30C.4.5 The Aggregation Port Statistics Group

This group of objects provides the functionality of the Aggregation Port Statistics managed object class (30.7.3). The Aggregation Port Statistics Group provides additional statistical information related to LACP and Marker protocol activity on the port.

### 30C.4.6 The Aggregation Port Debug Group

This group of objects provides the functionality of the Aggregation Port Debug managed object class (30.7.4). The Aggregation Port Debug Group provides additional information related to the operation of LACP on the port; this information is primarily aimed at debugging the operation of the protocol and detecting fault conditions.

## 30C.5 Relationship to other MIBs

It is assumed that a system implementing this MIB will also implement (at least) the “system” group defined in MIB-II defined in RFC 1213 and the “interfaces” group defined in RFC 2233.

### 30C.5.1 Relationship to the Interfaces MIB

RFC 2233, the Interface MIB Evolution, requires that any MIB that is an adjunct of the Interface MIB, clarify specific areas within the Interface MIB. These areas were intentionally left vague in RFC 2233 to avoid over constraining the MIB, thereby precluding management of certain media types.

Section 3.3 of RFC 2233 enumerates several areas that a media-specific MIB must clarify. Each of these areas is addressed in 30C.5.2 and 30C.5.3. The implementor is referred to RFC 2233 in order to understand the general intent of these areas.

In RFC 2233, the “interfaces” group is defined as being mandatory for all systems and contains information on an entity’s interfaces, where each interface is thought of as being attached to a *subnetwork*. (Note that this term is not to be confused with *subnet*, which refers to an addressing partitioning scheme used in the Internet suite of protocols.) The term *segment* is sometimes used to refer to such a subnetwork.

Implicit in this MIB is the notion of Aggregators and Aggregation ports. Each of these Aggregators and Aggregation ports is associated with one interface of the “interfaces” group (one row in the ifTable) and each port is associated with a different interface.

Each Aggregator and Aggregation port is uniquely identified by an interface number (ifIndex). The ifIndex value assigned to a given Aggregation port is the same as the ifIndex value assigned to the MAC interface with which that Aggregation port is associated.

### 30C.5.2 Layering model

This annex assumes the interpretation of the Interfaces Group to be in accordance with RFC 2233, which states that the ifTable contains information on the managed resource’s interfaces and that each sublayer below the internetwork layer of a network interface is considered an interface.

This annex recommends that, within an entity, aggregations that are instantiated as an entry in dot3adAggTable are also represented by an entry in ifTable.

Where an entity contains Link Aggregation entities that transmit and receive traffic to/from an aggregation, these should be represented in the ifTable as interfaces of type ieee8023adLag(161).

### 30C.5.3 ifStackTable

If the ifStackTable defined in RFC1573 is implemented, then

- a) The relationship shown in the table has the property that an Aggregation is a higher interface relative to an Aggregation Port.
- b) This relationship is read-only.

NOTE—The restriction stated here is intended to enforce a strict hierarchical relationship between Aggregations and Aggregation Ports, and to prevent those relationships from being modified. The read-only restriction does not apply to any other relationships that may be expressed in the ifStackTable.

### 30C.5.4 ifRcvAddressTable

The ifRcvAddressTable contains all MAC Addresses, unicast, multicast, and broadcast, for which an interface can receive packets and forward them up to a higher layer entity for local consumption. An Aggregator has at least one such address.

### 30C.6 Definitions for Link Aggregation MIB

In the MIB definition<sup>21</sup> below, should there be any discrepancy between the DESCRIPTION text and the BEHAVIOUR DEFINED AS in the corresponding definition in Clause 30, the definition in Clause 30 shall take precedence.

#### NOTES

1—The ASCII for 30C.6 is available from <http://www.ieee802.org/3/publication/index.html>.<sup>22</sup>

```

IEEE8023-LAG-MIB DEFINITIONS ::= BEGIN

-----
-- IEEE 802.3ad MIB
-----

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE, Counter32, TimeTicks, BITS
        FROM SNMPv2-SMI
    DisplayString, MacAddress, TEXTUAL-CONVENTION, TruthValue
        FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    InterfaceIndex
        FROM IF-MIB
    PortList
        FROM Q-BRIDGE-MIB
    ;

lagMIB MODULE-IDENTITY
    LAST-UPDATED "9911220000Z"
    ORGANIZATION "IEEE 802.3ad Working Group"
    CONTACT-INFO
        " stds-802-3-trunking@majordomo.ieee.org"
    DESCRIPTION
        "The Link Aggregation module for managing IEEE 802.3ad."
    ::= { iso(1) member-body(2) us(840) ieee802dot3(10006) snmpmibs(300) linkagg(43)
        }

lagMIBObjects OBJECT IDENTIFIER ::= { lagMIB 1 }

-----

```

<sup>21</sup>MIB definitions are available at <http://www.ietf.org/>.

<sup>22</sup>Copyright release for SNMP MIB: Users of this standard may freely reproduce the SNMP MIB in this annex so it can be used for its intended purpose.

```
-- Textual Conventions
```

```
-----

LacpKey ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The Actor or Partner Key value."
    SYNTAX      INTEGER (0..65535)

LacpState ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The Actor and Partner State values from the LACPDU."
    SYNTAX      BITS {
        lacpActivity(0),
        lacpTimeout(1),
        aggregation(2),
        synchronization(3),
        collecting(4),
        distributing(5),
        defaulted(6),
        expired(7)
    }

ChurnState ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The state of the Churn Detection machine."
    SYNTAX      INTEGER {
        noChurn(1),
        churn(2),
        churnMonitor(3)
    }

-----

-- groups in the LAG MIB
-----

dot3adAgg OBJECT IDENTIFIER ::= { lagMIBObjects 1 }
dot3adAggPort OBJECT IDENTIFIER ::= { lagMIBObjects 2 }

-----

-- The Tables Last Changed Object
-----

dot3adTablesLastChanged OBJECT-TYPE
    SYNTAX      TimeTicks
    MAX-ACCESS  read-only
```

```

STATUS      current
DESCRIPTION
    "This object indicates the time of the
    most recent change to the dot3adAggTable,
    dot3adAggPortListTable, or
    dot3adAggPortTable."
 ::= { lagMIBObjects 3 }

-----
-- The Aggregator Configuration Table
-----

dot3adAggTable OBJECT-TYPE
SYNTAX      SEQUENCE OF Dot3adAggEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "A table that contains information about every
    Aggregator that is associated with this System."
REFERENCE
    "IEEE 802.3 Subclause 30.7.1"
 ::= { dot3adAgg 1 }

dot3adAggEntry OBJECT-TYPE
SYNTAX      Dot3adAggEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "A list of the Aggregator parameters. This is indexed
    by the ifIndex of the Aggregator."
INDEX { dot3adAggIndex }
 ::= { dot3adAggTable 1 }

Dot3adAggEntry ::=
SEQUENCE {
    dot3adAggIndex
        InterfaceIndex,
    dot3adAggMACAddress
        MacAddress,
    dot3adAggActorSystemPriority
        INTEGER,
    dot3adAggActorSystemID
        MacAddress,
    dot3adAggAggregateOrIndividual
        TruthValue,
    dot3adAggActorAdminKey
        LACPKey,
    dot3adAggActorOperKey
        LACPKey,
    dot3adAggPartnerSystemID
        MacAddress,
    dot3adAggPartnerSystemPriority
        INTEGER,
    dot3adAggPartnerOperKey
        LACPKey,
    dot3adAggCollectorMaxDelay

```

```

        INTEGER
    }

```

dot3adAggIndex OBJECT-TYPE

SYNTAX InterfaceIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The unique identifier allocated to this Aggregator by the local System. This attribute identifies an Aggregator instance among the subordinate managed objects of the containing object. This value is read-only."

REFERENCE

"IEEE 802.3 Subclause 30.7.1.1.1"

::= { dot3adAggEntry 1 }

dot3adAggMACAddress OBJECT-TYPE

SYNTAX MacAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A 6-octet read-only value carrying the individual MAC address assigned to the Aggregator."

REFERENCE

"IEEE 802.3 Subclause 30.7.1.1.9"

::= { dot3adAggEntry 2 }

dot3adAggActorSystemPriority OBJECT-TYPE

SYNTAX INTEGER (0..65535)

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"A 2-octet read-write value indicating the priority value associated with the Actor's System ID."

REFERENCE

"IEEE 802.3 Subclause 30.7.1.1.5"

::= { dot3adAggEntry 3 }

dot3adAggActorSystemID OBJECT-TYPE

SYNTAX MacAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A 6-octet read-write MAC address value used as a unique identifier for the System that contains this Aggregator. NOTE-From the perspective of the Link Aggregation mechanisms described in Clause 43, only a single combination of Actor's System ID and System Priority are considered, and no distinction is made between the values of these parameters for an Aggregator and the port(s) that are associated with it; i.e., the protocol is described in terms of the operation of aggregation within a single System. However, the managed objects provided for the Aggregator and the port both allow management of these parameters. The result of this is to permit a single piece of equipment to be configured by

management to contain more than one System from the point of view of the operation of Link Aggregation. This may be of particular use in the configuration of equipment that has limited aggregation capability (see 43.6)."

## REFERENCE

"IEEE 802.3 Subclause 30.7.1.1.4"  
 ::= { dot3adAggEntry 4 }

## dot3adAggAggregateOrIndividual OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"A read-only Boolean value indicating whether the Aggregator represents an Aggregate ('TRUE') or an Individual link ('FALSE')."

## REFERENCE

"IEEE 802.3 Subclause 30.7.1.1.6"  
 ::= { dot3adAggEntry 5 }

## dot3adAggActorAdminKey OBJECT-TYPE

SYNTAX LACPKey

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"The current administrative value of the Key for the Aggregator. The administrative Key value may differ from the operational Key value for the reasons discussed in 43.6.2. This is a 16-bit, read-write value. The meaning of particular Key values is of local significance."

## REFERENCE

"IEEE 802.3 Subclause 30.7.1.1.7"  
 ::= { dot3adAggEntry 6 }

## dot3adAggActorOperKey OBJECT-TYPE

SYNTAX LACPKey

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The current operational value of the Key for the Aggregator. The administrative Key value may differ from the operational Key value for the reasons discussed in 43.6.2. This is a 16-bit read-only value. The meaning of particular Key values is of local significance."

## REFERENCE

"IEEE 802.3 Subclause 30.7.1.1.8"  
 ::= { dot3adAggEntry 7 }

## dot3adAggPartnerSystemID OBJECT-TYPE

SYNTAX MacAddress

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"A 6-octet read-only MAC address value consisting of the unique identifier for the current protocol Partner of

this Aggregator. A value of zero indicates that there is no known Partner. If the aggregation is manually configured, this System ID value will be a value assigned by the local System."

## REFERENCE

"IEEE 802.3 Subclause 30.7.1.1.10"

::= { dot3adAggEntry 8 }

## dot3adAggPartnerSystemPriority OBJECT-TYPE

SYNTAX INTEGER (0..65535)

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"A 2-octet read-only value that indicates the priority value associated with the Partner's System ID. If the aggregation is manually configured, this System Priority value will be a value assigned by the local System."

## REFERENCE

"IEEE 802.3 Subclause 30.7.1.1.11"

::= { dot3adAggEntry 9 }

## dot3adAggPartnerOperKey OBJECT-TYPE

SYNTAX LcpKey

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The current operational value of the Key for the Aggregator's current protocol Partner. This is a 16-bit read-only value. If the aggregation is manually configured, this Key value will be a value assigned by the local System."

## REFERENCE

"IEEE 802.3 Subclause 30.7.1.1.12"

::= { dot3adAggEntry 10 }

## dot3adAggCollectorMaxDelay OBJECT-TYPE

SYNTAX INTEGER (0..65535)

MAX-ACCESS read-write

STATUS current

## DESCRIPTION

"The value of this 16-bit read-write attribute defines the maximum delay, in tens of microseconds, that may be imposed by the Frame Collector between receiving a frame from an Aggregator Parser, and either delivering the frame to its MAC Client or discarding the frame (see 43.2.3.1.1)."

## REFERENCE

"IEEE 802.3 Subclause 30.7.1.1.32"

::= { dot3adAggEntry 11 }

-----  
 -- The Aggregation Port List Table  
 -----

## dot3adAggPortListTable OBJECT-TYPE



```

SYNTAX      SEQUENCE OF Dot3adAggPortListEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "A table that contains a list of all the ports
    associated with each Aggregator."
REFERENCE
    "IEEE 802.3 Subclause 30.7.1.1.30"
 ::= { dot3adAgg 2 }

```

## dot3adAggPortListEntry OBJECT-TYPE

```

SYNTAX      Dot3adAggPortListEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "A list of the ports associated with a given Aggregator.
    This is indexed by the ifIndex of the Aggregator."
INDEX { dot3adAggIndex }
 ::= { dot3adAggPortListTable 1 }

```

## Dot3adAggPortListEntry ::=

```

SEQUENCE {
    dot3adAggPortListPorts
        PortList
}

```

## dot3adAggPortListPorts OBJECT-TYPE

```

SYNTAX      PortList
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The complete set of ports currently associated with
    this Aggregator. Each bit set in this list represents
    an Actor Port member of this Link Aggregation."
REFERENCE
    "IEEE 802.3 Subclause 30.7.1.1.30"
 ::= { dot3adAggPortListEntry 1 }

```

```

-----
-- The Aggregation Port Table
-----

```

## dot3adAggPortTable OBJECT-TYPE

```

SYNTAX      SEQUENCE OF Dot3adAggPortEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "A table that contains Link Aggregation Control
    configuration information about every
    Aggregation Port associated with this device.
    A row appears in this table for each physical port."
REFERENCE
    "IEEE 802.3 Subclause 30.7.2"
 ::= { dot3adAggPort 1 }

```

```

dot3adAggPortEntry OBJECT-TYPE
    SYNTAX      Dot3adAggPortEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A list of Link Aggregation Control configuration
        parameters for each Aggregation Port on this device."
    INDEX       { dot3adAggPortIndex }
    ::= { dot3adAggPortTable 1 }

```

```

Dot3adAggPortEntry ::=
    SEQUENCE {
        dot3adAggPortIndex
            InterfaceIndex,
        dot3adAggPortActorSystemPriority
            INTEGER,
        dot3adAggPortActorSystemID
            MacAddress,
        dot3adAggPortActorAdminKey
            LACPKey,
        dot3adAggPortActorOperKey
            LACPKey,
        dot3adAggPortPartnerAdminSystemPriority
            INTEGER,
        dot3adAggPortPartnerOperSystemPriority
            INTEGER,
        dot3adAggPortPartnerAdminSystemID
            MacAddress,
        dot3adAggPortPartnerOperSystemID
            MacAddress,
        dot3adAggPortPartnerAdminKey
            LACPKey,
        dot3adAggPortPartnerOperKey
            LACPKey,
        dot3adAggPortSelectedAggID
            InterfaceIndex,
        dot3adAggPortAttachedAggID
            InterfaceIndex,
        dot3adAggPortActorPort
            INTEGER,
        dot3adAggPortActorPortPriority
            INTEGER,
        dot3adAggPortPartnerAdminPort
            INTEGER,
        dot3adAggPortPartnerOperPort
            INTEGER,
        dot3adAggPortPartnerAdminPortPriority
            INTEGER,
        dot3adAggPortPartnerOperPortPriority
            INTEGER,
        dot3adAggPortActorAdminState
            LACPState,
        dot3adAggPortActorOperState
            LACPState,
        dot3adAggPortPartnerAdminState
            LACPState,

```

```

dot3adAggPortPartnerOperState
    LACPState,
dot3adAggPortAggregateOrIndividual
    TruthValue
}

```

## dot3adAggPortIndex OBJECT-TYPE

```

SYNTAX      InterfaceIndex
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "The ifIndex of the port"
REFERENCE
    "IEEE 802.3 Subclause 30.7.2.1.1"
::= { dot3adAggPortEntry 1 }

```

## dot3adAggPortActorSystemPriority OBJECT-TYPE

```

SYNTAX      INTEGER (0..65535)
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "A 2-octet read-write value used to define the priority
    value associated with the Actor's System ID."
REFERENCE
    "IEEE 802.3 Subclause 30.7.2.1.2"
::= { dot3adAggPortEntry 2 }

```

## dot3adAggPortActorSystemID OBJECT-TYPE

```

SYNTAX      MacAddress
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A 6-octet read-only MAC address value that defines
    the value of the System ID for the System that contains this
    Aggregation Port."
REFERENCE
    "IEEE 802.3 Subclause 30.7.2.1.3"
::= { dot3adAggPortEntry 3 }

```

## dot3adAggPortActorAdminKey OBJECT-TYPE

```

SYNTAX      LACPKey
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The current administrative value of the Key for the
    Aggregation Port. This is a 16-bit read-write value.
    The meaning of particular Key values is of local significance."
REFERENCE
    "IEEE 802.3 Subclause 30.7.2.1.4"
::= { dot3adAggPortEntry 4 }

```

## dot3adAggPortActorOperKey OBJECT-TYPE

```

SYNTAX      LACPKey
MAX-ACCESS  read-write

```

STATUS current  
 DESCRIPTION  
 "The current operational value of the Key for the Aggregation Port. This is a 16-bit read-only value. The meaning of particular Key values is of local significance."  
 REFERENCE  
 "IEEE 802.3 Subclause 30.7.2.1.5"  
 ::= { dot3adAggPortEntry 5 }

## dot3adAggPortPartnerAdminSystemPriority OBJECT-TYPE

SYNTAX INTEGER (0..65535)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "A 2-octet read-write value used to define the administrative value of priority associated with the Partner's System ID. The assigned value is used, along with the value of aAggPortPartnerAdminSystemID, aAggPortPartnerAdminKey, aAggPortPartnerAdminPort, and aAggPortPartnerAdminPortPriority, in order to achieve manually configured aggregation."  
 REFERENCE  
 "IEEE 802.3 Subclause 30.7.2.1.7"  
 ::= { dot3adAggPortEntry 6 }

## dot3adAggPortPartnerOperSystemPriority OBJECT-TYPE

SYNTAX INTEGER (0..65535)  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "A 2-octet read-only value indicating the operational value of priority associated with the Partner's System ID. The value of this attribute may contain the manually configured value carried in aAggPortPartnerAdminSystemPriority if there is no protocol Partner."  
 REFERENCE  
 "IEEE 802.3 Subclause 30.7.2.1.7"  
 ::= { dot3adAggPortEntry 7 }

## dot3adAggPortPartnerAdminSystemID OBJECT-TYPE

SYNTAX MacAddress  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "A 6-octet read-write MacAddress value representing the administrative value of the Aggregation Port's protocol Partner's System ID. The assigned value is used, along with the value of aAggPortPartnerAdminSystemPriority, aAggPortPartnerAdminKey, aAggPortPartnerAdminPort, and aAggPortPartnerAdminPortPriority, in order to achieve manually configured aggregation."  
 REFERENCE  
 "IEEE 802.3 Subclause 30.7.2.1.8"  
 ::= { dot3adAggPortEntry 8 }

## dot3adAggPortPartnerOperSystemID OBJECT-TYPE

```

SYNTAX      MacAddress
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A 6-octet read-only MACAddress value representing
    the current value of the Aggregation Port's protocol Partner's
    System ID. A value of zero indicates that there is no known
    protocol Partner. The value of this attribute may contain the
    manually configured value carried in
    aAggPortPartnerAdminSystemID if there is no protocol Partner."
REFERENCE
    "IEEE 802.3 Subclause 30.7.2.1.9"
::= { dot3adAggPortEntry 9 }

```

## dot3adAggPortPartnerAdminKey OBJECT-TYPE

```

SYNTAX      LACPKey
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The current administrative value of the Key for the
    protocol Partner. This is a 16-bit read-write value.
    The assigned value is used, along with the value of
    aAggPortPartnerAdminSystemPriority, aAggPortPartnerAdminSystemID,
    aAggPortPartnerAdminPort, and aAggPortPartnerAdminPortPriority,
    in order to achieve manually configured aggregation.
REFERENCE
    "IEEE 802.3 Subclause 30.7.2.1.10"
::= { dot3adAggPortEntry 10 }

```

## dot3adAggPortPartnerOperKey OBJECT-TYPE

```

SYNTAX      LACPKey
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The current operational value of the Key for the
    protocol Partner. The value of this attribute may contain
    the manually configured value carried in
    aAggPortPartnerAdminKey if there is no protocol Partner.
    This is a 16-bit read-only value."
REFERENCE
    "IEEE 802.3 Subclause 30.7.2.1.11"
::= { dot3adAggPortEntry 11 }

```

## dot3adAggPortSelectedAggID OBJECT-TYPE

```

SYNTAX      InterfaceIndex
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The identifier value of the Aggregator that this Aggregation
    Port has currently selected. Zero indicates that the Aggregation
    Port has not selected an Aggregator, either because it is in the
    process of detaching from an Aggregator or because there is no
    suitable Aggregator available for it to select. This value is read-only."
REFERENCE
    "IEEE 802.3 Subclause 30.7.2.1.12"
::= { dot3adAggPortEntry 12 }

```

```

dot3adAggPortAttachedAggID OBJECT-TYPE
    SYNTAX      InterfaceIndex
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The identifier value of the Aggregator that this Aggregation
        Port is currently attached to. Zero indicates that the Aggregation
        Port is not currently attached to an Aggregator. This value is read-only."
    REFERENCE
        "IEEE 802.3 Subclause 30.7.2.1.13"
    ::= { dot3adAggPortEntry 13 }

```

```

dot3adAggPortActorPort OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The port number locally assigned to the Aggregation Port.
        The port number is communicated in LACPDUs as the
        Actor_Port. This value is read-only."
    REFERENCE
        "IEEE 802.3 Subclause 30.7.2.1.14"
    ::= { dot3adAggPortEntry 14 }

```

```

dot3adAggPortActorPortPriority OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The priority value assigned to this Aggregation Port.
        This 16-bit value is read-write."
    REFERENCE
        "IEEE 802.3 Subclause 30.7.2.1.15"
    ::= { dot3adAggPortEntry 15 }

```

```

dot3adAggPortPartnerAdminPort OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)
    MAX-ACCESS  read-write
    STATUS      current
    DESCRIPTION
        "The current administrative value of the port number
        for the protocol Partner. This is a 16-bit read-write value.
        The assigned value is used, along with the value of
        aAggPortPartnerAdminSystemPriority,
        aAggPortPartnerAdminSystemID, aAggPortPartnerAdminKey,
        and aAggPortPartnerAdminPortPriority,
        in order to achieve manually configured aggregation."
    REFERENCE
        "IEEE 802.3 Subclause 30.7.2.1.16"
    ::= { dot3adAggPortEntry 16 }

```

```

dot3adAggPortPartnerOperPort OBJECT-TYPE
    SYNTAX      INTEGER (0..65535)

```

MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The operational port number assigned to this Aggregation Port by the Aggregation Port's protocol Partner. The value of this attribute may contain the manually configured value carried in aAggPortPartnerAdminPort if there is no protocol Partner. This 16-bit value is read-only."

## REFERENCE

"IEEE 802.3 Subclause 30.7.2.1.17"

::= { dot3adAggPortEntry 17 }

## dot3adAggPortPartnerAdminPortPriority OBJECT-TYPE

SYNTAX INTEGER (0..65535)  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "The current administrative value of the port priority for the protocol Partner. This is a 16-bit read-write value. The assigned value is used, along with the value of aAggPortPartnerAdminSystemPriority, aAggPortPartnerAdminSystemID, aAggPortPartnerAdminKey, and aAggPortPartnerAdminPort, in order to achieve manually configured aggregation."

## REFERENCE

"IEEE 802.3 Subclause 30.7.2.1.18"

::= { dot3adAggPortEntry 18 }

## dot3adAggPortPartnerOperPortPriority OBJECT-TYPE

SYNTAX INTEGER (0..65535)  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The priority value assigned to this Aggregation Port by the Partner. The value of this attribute may contain the manually configured value carried in aAggPortPartnerAdminPortPriority if there is no protocol Partner. This 16-bit value is read-only."

## REFERENCE

"IEEE 802.3 Subclause 30.7.2.1.19"

::= { dot3adAggPortEntry 19 }

## dot3adAggPortActorAdminState OBJECT-TYPE

SYNTAX LacpState  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 "A string of 8 bits, corresponding to the administrative values of Actor\_State (43.4.2) as transmitted by the Actor in LACPDUs. The first bit corresponds to bit 0 of Actor\_State (LACP\_Activity), the second bit corresponds to bit 1 (LACP\_Timeout), the third bit corresponds to bit 2 (Aggregation), the fourth bit corresponds to bit 3 (Synchronization), the fifth bit corresponds to bit 4 (Collecting), the sixth bit corresponds to bit 5 (Distributing), the seventh bit corresponds to bit 6 (Defaulted), and the eighth bit corresponds to bit 7 (Expired). These values allow administrative control over the values of LACP\_Activity,

LACP\_Timeout and Aggregation. This attribute value is read-write.”  
 REFERENCE  
 “IEEE 802.3 Subclause 30.7.2.1.20”  
 ::= { dot3adAggPortEntry 20 }

dot3adAggPortActorOperState OBJECT-TYPE

SYNTAX LACPState  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 “A string of 8 bits, corresponding to the current operational values of Actor\_State as transmitted by the Actor in LACPDUs. The bit allocations are as defined in 30.7.2.1.20. This attribute value is read-only.”  
 REFERENCE  
 “IEEE 802.3 Subclause 30.7.2.1.21”  
 ::= { dot3adAggPortEntry 21 }

dot3adAggPortPartnerAdminState OBJECT-TYPE

SYNTAX LACPState  
 MAX-ACCESS read-write  
 STATUS current  
 DESCRIPTION  
 “A string of 8 bits, corresponding to the current administrative value of Actor\_State for the protocol Partner. The bit allocations are as defined in 30.7.2.1.20. This attribute value is read-write. The assigned value is used in order to achieve manually configured aggregation.”  
 REFERENCE  
 “IEEE 802.3 Subclause 30.7.2.1.22”  
 ::= { dot3adAggPortEntry 22 }

dot3adAggPortPartnerOperState OBJECT-TYPE

SYNTAX LACPState  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 “A string of 8 bits, corresponding to the current values of Actor\_State in the most recently received LACPDU transmitted by the protocol Partner. The bit allocations are as defined in 30.7.2.1.20. In the absence of an active protocol Partner, this value may reflect the manually configured value aAggPortPartnerAdminState. This attribute value is read-only.”  
 REFERENCE  
 “IEEE 802.3 Subclause 30.7.2.1.23”  
 ::= { dot3adAggPortEntry 23 }

dot3adAggPortAggregateOrIndividual OBJECT-TYPE

SYNTAX TruthValue  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 “A read-only Boolean value indicating whether the Aggregation Port is able to Aggregate ('TRUE') or is only able to operate as an Individual link ('FALSE').”



## REFERENCE

“IEEE 802.3 Subclause 30.7.1.1.24”  
 ::= { dot3adAggPortEntry 24 }

-----  
 -- LACP Statistics Table  
 -----

## dot3adAggPortStatsTable OBJECT-TYPE

SYNTAX SEQUENCE OF Dot3adAggPortStatsEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

“A table that contains Link Aggregation information  
 about every port that is associated with this device.  
 A row appears in this table for each physical port.”

## REFERENCE

“IEEE 802.3 Subclause 30.7.3”  
 ::= { dot3adAggPort 2 }

## dot3adAggPortStatsEntry OBJECT-TYPE

SYNTAX Dot3adAggPortStatsEntry

MAX-ACCESS not-accessible

STATUS current

## DESCRIPTION

“A list of Link Aggregation Control Protocol statistics  
 for each port on this device.”

INDEX { dot3adAggPortIndex }

::= { dot3adAggPortStatsTable 1 }

## Dot3adAggPortStatsEntry ::=

SEQUENCE {

dot3adAggPortStatsLACPDUsRx  
 Counter32,  
 dot3adAggPortStatsMarkerPDUsRx  
 Counter32,  
 dot3adAggPortStatsMarkerResponsePDUsRx  
 Counter32,  
 dot3adAggPortStatsUnknownRx  
 Counter32,  
 dot3adAggPortStatsIllegalRx  
 Counter32,  
 dot3adAggPortStatsLACPDUsTx  
 Counter32,  
 dot3adAggPortStatsMarkerPDUsTx  
 Counter32,  
 dot3adAggPortStatsMarkerResponsePDUsTx  
 Counter32

}

## dot3adAggPortStatsLACPDUsRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of valid LACPDUs received on this Aggregation Port. This value is read-only."

## REFERENCE

"IEEE 802.3 Subclause 30.7.3.1.2"

::= { dot3adAggPortStatsEntry 1 }

## dot3adAggPortStatsMarkerPDUsRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of valid Marker PDUs received on this Aggregation Port. This value is read-only."

## REFERENCE

"IEEE 802.3 Subclause 30.7.3.1.3"

::= { dot3adAggPortStatsEntry 2 }

## dot3adAggPortStatsMarkerResponsePDUsRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of valid Marker Response PDUs received on this Aggregation Port. This value is read-only."

## REFERENCE

"IEEE 802.3 Subclause 30.7.3.1.4"

::= { dot3adAggPortStatsEntry 3 }

## dot3adAggPortStatsUnknownRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of frames received that either:  
 - carry the Slow Protocols Ethernet Type value (43B.4),  
 but contain an unknown PDU, or:  
 - are addressed to the Slow Protocols group MAC  
 Address (43B.4), but do not carry the Slow Protocols Ethernet Type.  
 This value is read-only."

## REFERENCE

"IEEE 802.3 Subclause 30.7.3.1.5"

::= { dot3adAggPortStatsEntry 4 }

## dot3adAggPortStatsIllegalRx OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

## DESCRIPTION

"The number of frames received that carry the Slow Protocols Ethernet Type value (43B.4), but contain a badly formed PDU or an illegal value of Protocol Subtype (43B.4).  
 This value is read-only."

## REFERENCE

"IEEE 802.3 Subclause 30.7.3.1.6"

```
::= { dot3adAggPortStatsEntry 5 }
```

```
dot3adAggPortStatsLACPDUsTx OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
    "The number of LACPDUs transmitted on this
    Aggregation Port. This value is read-only."
```

```
REFERENCE
```

```
    "IEEE 802.3 Subclause 30.7.3.1.7"
```

```
::= { dot3adAggPortStatsEntry 6 }
```

```
dot3adAggPortStatsMarkerPDUsTx OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
    "The number of Marker PDUs transmitted on this
    Aggregation Port. This value is read-only."
```

```
REFERENCE
```

```
    "IEEE 802.3 Subclause 30.7.3.1.8"
```

```
::= { dot3adAggPortStatsEntry 7 }
```

```
dot3adAggPortStatsMarkerResponsePDUsTx OBJECT-TYPE
```

```
SYNTAX Counter32
```

```
MAX-ACCESS read-only
```

```
STATUS current
```

```
DESCRIPTION
```

```
    "The number of Marker Response PDUs transmitted
    on this Aggregation Port. This value is read-only."
```

```
REFERENCE
```

```
    "IEEE 802.3 Subclause 30.7.3.1.9"
```

```
::= { dot3adAggPortStatsEntry 8 }
```

```
-----
-- LACP Debug Table
-----
```

```
dot3adAggPortDebugTable OBJECT-TYPE
```

```
SYNTAX SEQUENCE OF Dot3adAggPortDebugEntry
```

```
MAX-ACCESS not-accessible
```

```
STATUS current
```

```
DESCRIPTION
```

```
    "A table that contains Link Aggregation debug
    information about every port that is associated with
    this device. A row appears in this table for each
    physical port."
```

```
REFERENCE
```

```
    "IEEE 802.3 Subclause 30.7.4"
```

```
::= { dot3adAggPort 3 }
```

```
dot3adAggPortDebugEntry OBJECT-TYPE
```

```
SYNTAX Dot3adAggPortDebugEntry
```

```
MAX-ACCESS not-accessible
```

```

STATUS      current
DESCRIPTION
    "A list of the debug parameters for a port."
INDEX { dot3adAggPortIndex }
 ::= { dot3adAggPortDebugTable 1 }

```

```

Dot3adAggPortDebugEntry ::=
SEQUENCE {
    dot3adAggPortDebugRxState
        INTEGER,
    dot3adAggPortDebugLastRxTime
        TimeTicks,
    dot3adAggPortDebugMuxState
        INTEGER,
    dot3adAggPortDebugMuxReason
        DisplayString,
    dot3adAggPortDebugActorChurnState
        ChurnState,
    dot3adAggPortDebugPartnerChurnState
        ChurnState,
    dot3adAggPortDebugActorChurnCount
        Counter32,
    dot3adAggPortDebugPartnerChurnCount
        Counter32,
    dot3adAggPortDebugActorSyncTransitionCount
        Counter32,
    dot3adAggPortDebugPartnerSyncTransitionCount
        Counter32,
    dot3adAggPortDebugActorChangeCount
        Counter32,
    dot3adAggPortDebugPartnerChangeCount
        Counter32
}

```

dot3adAggPortDebugRxState OBJECT-TYPE

```

SYNTAX      INTEGER {
                current(1),
                expired(2),
                defaulted(3),
                initialize(4),
                lacpDisabled(5),
                portDisabled(6)
            }

```

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This attribute holds the value 'current' if the Receive state machine for the Aggregation Port is in the CURRENT state, 'expired' if the Receive state machine is in the EXPIRED state, 'defaulted' if the Receive state machine is in the DEFAULTED state, 'initialize' if the Receive state machine is in the INITIALIZE state, 'lacpDisabled' if the Receive state machine is in the LACP\_DISABLED state, or 'portDisabled' if the Receive state machine is in the PORT\_DISABLED state. This value is read-only."

REFERENCE

```

    "IEEE 802.3 Subclause 30.7.4.1.2"
    ::= { dot3adAggPortDebugEntry 1 }

```

dot3adAggPortDebugLastRxTime OBJECT-TYPE

```

SYNTAX      TimeTicks
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The value of aTimeSinceSystemReset (F.2.1) when
    the last LACPDU was received by this Aggregation Port.
    This value is read-only."
REFERENCE
    "IEEE 802.3 Subclause 30.7.4.1.3"
    ::= { dot3adAggPortDebugEntry 2 }

```

dot3adAggPortDebugMuxState OBJECT-TYPE

```

SYNTAX      INTEGER {
                detached(1),
                waiting(2),
                attached(3),
                collecting(4),
                distributing(5),
                collecting_distributing(6)
            }
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "This attribute holds the value 'detached' if the Mux
    state machine (43.4.14) for the Aggregation Port is
    in the DETACHED state, 'waiting' if the Mux state machine
    is in the WAITING state, 'attached' if the Mux state
    machine for the Aggregation Port is in the ATTACHED
    state, 'collecting' if the Mux state machine for the
    Aggregation Port is in the COLLECTING state, 'distributing'
    if the Mux state machine for the Aggregation Port is
    in the DISTRIBUTING state, and 'collecting_distributing'
    if the Mux state machine for the Aggregation Port is in
    the COLLECTING_DISTRIBUTING state.
    This value is read-only."
REFERENCE
    "IEEE 802.3 Subclause 30.7.4.1.4"
    ::= { dot3adAggPortDebugEntry 3 }

```

dot3adAggPortDebugMuxReason OBJECT-TYPE

```

SYNTAX      DisplayString
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "A human-readable text string indicating the reason
    for the most recent change of Mux machine state.
    This value is read-only."
REFERENCE
    "IEEE 802.3 Subclause 30.7.4.1.5"
    ::= { dot3adAggPortDebugEntry 4 }

```

dot3adAggPortDebugActorChurnState OBJECT-TYPE  
 SYNTAX ChurnState  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The state of the Actor Churn Detection machine (43.4.17) for the Aggregation Port. A value of 'noChurn' indicates that the state machine is in either the NO\_ACTOR\_CHURN or the ACTOR\_CHURN\_MONITOR state, and 'churn' indicates that the state machine is in the ACTOR\_CHURN state. This value is read-only."  
 REFERENCE  
 "IEEE 802.3 Subclause 30.7.4.1.6"  
 ::= { dot3adAggPortDebugEntry 5 }

dot3adAggPortDebugPartnerChurnState OBJECT-TYPE  
 SYNTAX ChurnState  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "The state of the Partner Churn Detection machine (43.4.17) for the Aggregation Port. A value of 'noChurn' indicates that the state machine is in either the NO\_PARTNER\_CHURN or the PARTNER\_CHURN\_MONITOR state, and 'churn' indicates that the state machine is in the PARTNER\_CHURN state. This value is read-only."  
 REFERENCE  
 "IEEE 802.3 Subclause 30.7.4.1.7"  
 ::= { dot3adAggPortDebugEntry 6 }

dot3adAggPortDebugActorChurnCount OBJECT-TYPE  
 SYNTAX Counter32  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "Count of the number of times the Actor Churn state machine has entered the ACTOR\_CHURN state. This value is read-only."  
 REFERENCE  
 "IEEE 802.3 Subclause 30.7.4.1.8"  
 ::= { dot3adAggPortDebugEntry 7 }

dot3adAggPortDebugPartnerChurnCount OBJECT-TYPE  
 SYNTAX Counter32  
 MAX-ACCESS read-only  
 STATUS current  
 DESCRIPTION  
 "Count of the number of times the Partner Churn state machine has entered the PARTNER\_CHURN state. This value is read-only."  
 REFERENCE  
 "IEEE 802.3 Subclause 30.7.4.1.9"  
 ::= { dot3adAggPortDebugEntry 8 }

```

dot3adAggPortDebugActorSyncTransitionCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Count of the number of times the Actor's Mux state
        machine (43.4.15) has entered the IN_SYNC state.
        This value is read-only."
    REFERENCE
        "IEEE 802.3 Subclause 30.7.4.1.10"
    ::= { dot3adAggPortDebugEntry 9 }

dot3adAggPortDebugPartnerSyncTransitionCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Count of the number of times the Partner's Mux
        state machine (43.4.15) has entered the IN_SYNC state.
        This value is read-only."
    REFERENCE
        "IEEE 802.3 Subclause 30.7.4.1.11"
    ::= { dot3adAggPortDebugEntry 10 }

dot3adAggPortDebugActorChangeCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Count of the number of times the Actor's perception of
        the LAG ID for this Aggregation Port has changed.
        This value is read-only."
    REFERENCE
        "IEEE 802.3 Subclause 30.7.4.1.12"
    ::= { dot3adAggPortDebugEntry 11 }

dot3adAggPortDebugPartnerChangeCount OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Count of the number of times the Partner's perception of
        the LAG ID (see 43.3.6.1) for this Aggregation Port has changed.
        This value is read-only."
    REFERENCE
        "IEEE 802.3 Subclause 30.7.4.1.13"
    ::= { dot3adAggPortDebugEntry 12 }

-----
-- IEEE 802.3ad MIB - Conformance Information
-----

dot3adAggConformance OBJECT IDENTIFIER ::= { lagMIB 2 }

dot3adAggGroups OBJECT IDENTIFIER ::= { dot3adAggConformance 1 }

```

```

dot3adAggCompliances OBJECT IDENTIFIER
  ::= { dot3adAggConformance 2 }

-----
-- units of conformance
-----

dot3adAggGroup OBJECT-GROUP
  OBJECTS {
    dot3adAggActorSystemID,
    dot3adAggActorSystemPriority,
    dot3adAggAggregateOrIndividual,
    dot3adAggActorAdminKey,
    dot3adAggMACAddress,
    dot3adAggActorOperKey,
    dot3adAggPartnerSystemID,
    dot3adAggPartnerSystemPriority,
    dot3adAggPartnerOperKey,
    dot3adAggCollectorMaxDelay
  }
  STATUS      current
  DESCRIPTION
    "A collection of objects providing information about an
    aggregation."
  ::= { dot3adAggGroups 1 }

dot3adAggPortListGroup OBJECT-GROUP
  OBJECTS {
    dot3adAggPortListPorts
  }
  STATUS      current
  DESCRIPTION
    "A collection of objects providing information about every
    port in an aggregation."
  ::= { dot3adAggGroups 2 }

dot3adAggPortGroup OBJECT-GROUP
  OBJECTS {
    dot3adAggPortActorSystemPriority,
    dot3adAggPortActorSystemID,
    dot3adAggPortActorAdminKey,
    dot3adAggPortActorOperKey,
    dot3adAggPortPartnerAdminSystemPriority,
    dot3adAggPortPartnerOperSystemPriority,
    dot3adAggPortPartnerAdminSystemID,
    dot3adAggPortPartnerOperSystemID,
    dot3adAggPortPartnerAdminKey,
    dot3adAggPortPartnerOperKey,
    dot3adAggPortSelectedAggID,
    dot3adAggPortAttachedAggID,
    dot3adAggPortActorPort,
    dot3adAggPortActorPortPriority,
    dot3adAggPortPartnerAdminPort,
    dot3adAggPortPartnerOperPort,
    dot3adAggPortPartnerAdminPortPriority,

```



```

        dot3adAggPortPartnerOperPortPriority,
        dot3adAggPortActorAdminState,
        dot3adAggPortActorOperState,
        dot3adAggPortPartnerAdminState,
        dot3adAggPortPartnerOperState,
        dot3adAggPortAggregateOrIndividual
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information about every
        port in an aggregation."
    ::= { dot3adAggGroups 3 }

dot3adAggPortStatsGroup OBJECT-GROUP
    OBJECTS {
        dot3adAggPortStatsLACPDUsRx,
        dot3adAggPortStatsMarkerPDUsRx,
        dot3adAggPortStatsMarkerResponsePDUsRx,
        dot3adAggPortStatsUnknownRx,
        dot3adAggPortStatsIllegalRx,
        dot3adAggPortStatsLACPDUsTx,
        dot3adAggPortStatsMarkerPDUsTx,
        dot3adAggPortStatsMarkerResponsePDUsTx
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing information about every
        port in an aggregation."
    ::= { dot3adAggGroups 4 }

dot3adAggPortDebugGroup OBJECT-GROUP
    OBJECTS {
        dot3adAggPortDebugRxState,
        dot3adAggPortDebugLastRxTime,
        dot3adAggPortDebugMuxState,
        dot3adAggPortDebugMuxReason,
        dot3adAggPortDebugActorChurnState,
        dot3adAggPortDebugPartnerChurnState,
        dot3adAggPortDebugActorChurnCount,
        dot3adAggPortDebugPartnerChurnCount,
        dot3adAggPortDebugActorSyncTransitionCount,
        dot3adAggPortDebugPartnerSyncTransitionCount,
        dot3adAggPortDebugActorChangeCount,
        dot3adAggPortDebugPartnerChangeCount
    }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing debug information about
        every aggregated port."
    ::= { dot3adAggGroups 5 }

dot3adTablesLastChangedGroup OBJECT-GROUP
    OBJECTS {
        dot3adTablesLastChanged
    }
    STATUS      current
    DESCRIPTION

```

```
    "A collection of objects providing information about the time
    of changes to the configuration of aggregations and their ports."
 ::= { dot3adAggGroup 6 }
```

```
-----
-- compliance statements
-----
```

```
dot3adAggCompliance MODULE-COMPLIANCE
  STATUS current
  DESCRIPTION
    "The compliance statement for device support of
    Link Aggregation."
```

```
MODULE
  MANDATORY-GROUPS {
    dot3adAggGroup,
    dot3adAggPortGroup,
    dot3adTablesLastChangedGroup
  }
```

```
GROUP dot3adAggPortListGroup
DESCRIPTION
  "This group is optional."
```

```
GROUP dot3adAggPortStatsGroup
DESCRIPTION
  "This group is optional."
```

```
GROUP dot3adAggPortDebugGroup
DESCRIPTION
  "This group is optional."
```

```
 ::= { dot3adAggCompliances 1 }
```

```
END
```

## Annex 31A

(normative)

### MAC Control opcode assignments

Table 31A–1 shows the currently defined opcode values and interpretations:

**Table 31A–1 – MAC Control opcodes**

Opcode (Hexadecimal)	MAC Control function	Specified in annex	Value/comment
00-00	Reserved		
00-01	PAUSE	31B	Requests that the recipient stop transmitting non-control frames for a period of time indicated by the parameters of this function.
00-02 through FF-FF	Reserved		

Opcodes are transmitted high-order octet first. Within each octet, bits are transmitted least-significant bit first. Reserved opcodes shall not be used by MAC Control sublayer entities.

Table 31A–2 shows the elements and semantics of the `indication_operand_list` for `MA_CONTROL.indication` primitives for each currently defined opcode value in Table 31A–1:

**Table 31A–2 – MAC Control indications**

PAUSE (opcode 0x0001)		
indication_operand_list element	Value	Interpretation
pause_status	paused	Indicates that the PAUSE function is inhibiting transmission of data frames by the MAC client.
	not_paused	Indicates that the PAUSE function is not inhibiting transmission of data frames by the MAC client.

## Annex 31B

(normative)

### MAC Control PAUSE operation

#### 31B.1 PAUSE description

The PAUSE operation is used to inhibit transmission of data frames for a specified period of time. A MAC Control client wishing to inhibit transmission of data frames from another station on the network generates a MA\_CONTROL.request primitive specifying:

- a) The globally assigned 48-bit multicast address 01-80-C2-00-00-01,
- b) The PAUSE opcode,
- c) A request\_operand indicating the length of time for which it wishes to inhibit data frame transmission. (See 31B.2.)

The PAUSE operation cannot be used to inhibit transmission of MAC Control frames.

PAUSE frames shall only be sent by DTEs configured to the full duplex mode of operation.

The globally assigned 48-bit multicast address 01-80-C2-00-00-01 has been reserved for use in MAC Control PAUSE frames for inhibiting transmission of data frames from a DTE in a full duplex mode IEEE 802.3® LAN. IEEE 802.1D®-conformant bridges will not forward frames sent to this multicast destination address, regardless of the state of the bridge's ports, or whether or not the bridge implements the MAC Control sublayer. To allow generic full duplex flow control, stations implementing the PAUSE operation shall instruct the MAC (e.g., through layer management) to enable reception of frames with destination address equal to this multicast address.

NOTE—By definition, an IEEE 802.3® LAN operating in full duplex mode comprises exactly two stations, thus there is no ambiguity regarding the destination DTE's identity. The use of a well-known multicast address relieves the MAC Control sublayer and its client from having to know, and maintain knowledge of the individual 48-bit address of the other DTE in a full duplex environment

#### 31B.2 Parameter semantics

The PAUSE opcode takes the following request\_operand:

**pause\_time**

A 2-octet, unsigned integer containing the length of time for which the receiving station is requested to inhibit data frame transmission. The field is transmitted most-significant octet first, and least-significant octet second. The pause\_time is measured in units of pause\_quanta, equal to 512 bit times of the particular implementation (See 4.4). The range of possible pause\_time is 0 to 65535 pause\_quanta.

## 31B.3 Detailed specification of PAUSE operation

### 31B.3.1 Transmit operation

Upon receipt of a MA\_CONTROL.request primitive containing the PAUSE opcode from a MAC Control client, the MAC Control sublayer calls the MAC sublayer TransmitFrame function with the following parameters:

- a) The destinationParam is set equal to the destination\_address parameter of the MA\_DATA.request primitive. This is currently restricted to the value specified in 31B.1.
- b) The sourceParam is set equal to the 48-bit individual address of the station.
- c) The lengthOrTypeParam is set to the reserved 802.3\_MAC\_Control value specified in 31.4.1.3.
- d) The dataParam is set equal to the concatenation of the PAUSE opcode encoding (see Annex 31A), the pause\_time request\_operand specified in the MA\_CONTROL.request primitive (see 2.3.3.2), and a field containing zeroes of the length specified in 31.4.1.6.

Upon receipt of a data transmission request from the MAC Control client through the MA\_DATA.request primitive, if the transmission of data frames has not been inhibited due to reception of a valid MAC Control frame specifying the PAUSE operation and a non-zero pause\_time, the MAC Control sublayer calls the MAC sublayer TransmitFrame function with the following parameters:

- a) The destinationParam is set equal to the destination\_address parameter of the MA\_CONTROL.request primitive.
- b) The sourceParam is set equal to the 48-bit individual address of the station.
- c) The lengthOrTypeParam and dataParam are set from the m\_sdu field of the MA\_DATA.request primitive.

### 31B.3.2 Transmit state diagram for PAUSE operation

It is not required that an implementation be able to transmit PAUSE frames. MAC Control sublayer entities that transmit PAUSE frames shall implement the Transmit State machine specified in this subclause.

#### 31B.3.2.1 Constants

pause\_command

The 2-octet encoding of the PAUSE command, as specified in Annex 31A.

reserved\_multicast\_address

The 48-bit address specified in 31B.1 (a).

802.3\_MAC\_Control

The 16 bit type field assignment for 802.3 MAC Control specified in 31.4.1.3.

phys\_Address

A 48-bit value, equal to the unicast address of the station implementing the MAC Control sublayer.

#### 31B.3.2.2 Variables

transmitEnabled

A boolean set by Network Management to indicate that the station is permitted to transmit on the network.

Values: true; Transmitter is enabled by management

false; Transmitter is disabled by management

transmission\_in\_progress

A boolean used to indicate to the Receive State Machine that a TransmitFrame function call is pending.

Values: true; transmission is in progress  
false; transmission is not in progress

n\_quanta\_tx

An integer indicating the number of pause\_quanta that the transmitter of a PAUSE frame is requesting that the receiver pause.

pause\_timer\_Done

A boolean set by the MAC Control PAUSE Operation Receive State Machine to indicate the expiration of a pause\_timer initiated by reception of a MAC Control frame with a PAUSE opcode.

Values: true; The pause\_timer has expired.  
false; The pause\_timer has not expired.

### 31B.3.2.3 Functions

TransmitFrame

The MAC Sublayer primitive called to transmit a frame with the specified parameters.

### 31B.3.2.4 Timers

pause\_timer

The timer governing the inhibition of transmission of data frames from a MAC Client by the PAUSE function.

### 31B.3.2.5 Messages

MA\_CONTROL.request

The service primitive used by a client to request a MAC Control sublayer function with the specified request\_operands.

MA\_DATA.request

The service primitive used by a client to request MAC data transmission with the specified parameters.

### 31B.3.2.6 Transmit state diagram for PAUSE operation

Figure 31B–1 depicts the Transmit State Diagram for a MAC Control sublayer entity implementing the PAUSE operation.

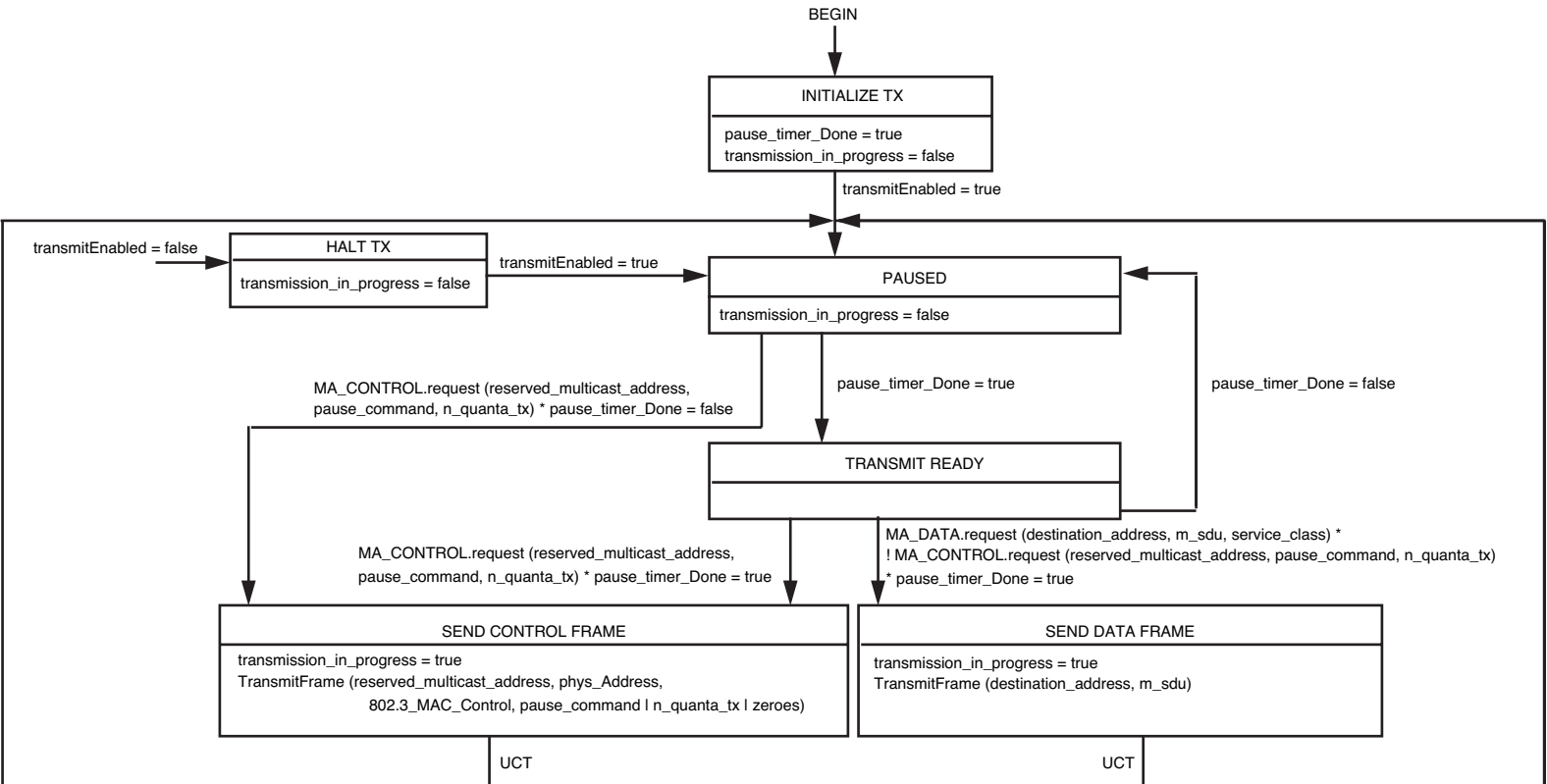


Figure 31B-1 — PAUSE Operation Transmit state diagram

### 31B.3.3 Receive operation

The opcode-independent MAC Control sublayer Receive State Machine accepts and parses valid frames received from the MAC sublayer. MAC Control sublayer entities that implement the PAUSE operation shall implement the Receive State machine specified in this subclause. The functions specified in this subclause are performed upon receipt of a valid Control frame containing the PAUSE opcode, and define the function called by the INITIATE MAC CONTROL FUNCTION state of Figure 31–4. (See 31.5.3.)

Upon receipt of a valid MAC Control frame with the opcode indicating PAUSE and the destination address indicating either: (1) the reserved multicast address specified in 31B.1, or (2) the unique physical address associated with this station, the MAC Control sublayer starts a timer for the length of time specified by the `pause_time request_operand` in the Control frame (see 31B.2). The value of the variable `pause_timer_Done` is set to false upon the setting of the `pause_timer` to a non-zero value by the MAC Control PAUSE Operation Receive State Machine. The value of the variable `pause_timer_Done` is set to true when the timer value reaches zero (i.e., the timer expires). If the received PAUSE operation indicates a `pause_timer` value of zero, the value of `pause_timer_Done` is set to true immediately.

The receipt of a valid MAC Control frame with the opcode indicating PAUSE and the destination address indicating the reserved multicast address specified in 31B.1 or the unique physical address associated with this station will cause the `pause_timer` to be set according to the `pause_time request_operand` in the newly received frame, regardless of the current setting of the `pause_timer`, i.e., new PAUSE operations override earlier PAUSE operations.

### 31B.3.4 Receive state diagram for PAUSE operation

#### 31B.3.4.1 Constants

`pause_quantum`

The unit of measurement for pause time specified in 31B.2

`pause_command`

The 2-octet encoding of the PAUSE command, as specified in Annex 31A.

`reserved_multicast_address`

The 48-bit address specified in 31B.1 (a).

`phys_Address`

A 48-bit value, equal to the unicast address of the station implementing the MAC Control sublayer.

#### 31B.3.4.2 Variables

`n_quanta_rx`

An integer used to extract the number of pause quanta to set the `pause_timer` from the `dataParam` of the received frame.

`opcode`

The opcode parsed from the received frame.

`DA`

The destination address from the `ReceiveFrame` function.

`transmission_in_progress`

A boolean used by the Transmit State Machine to indicate that a `TransmitFrame` function call is pending.

Values: true; transmission is in progress

false; transmission is not in progress



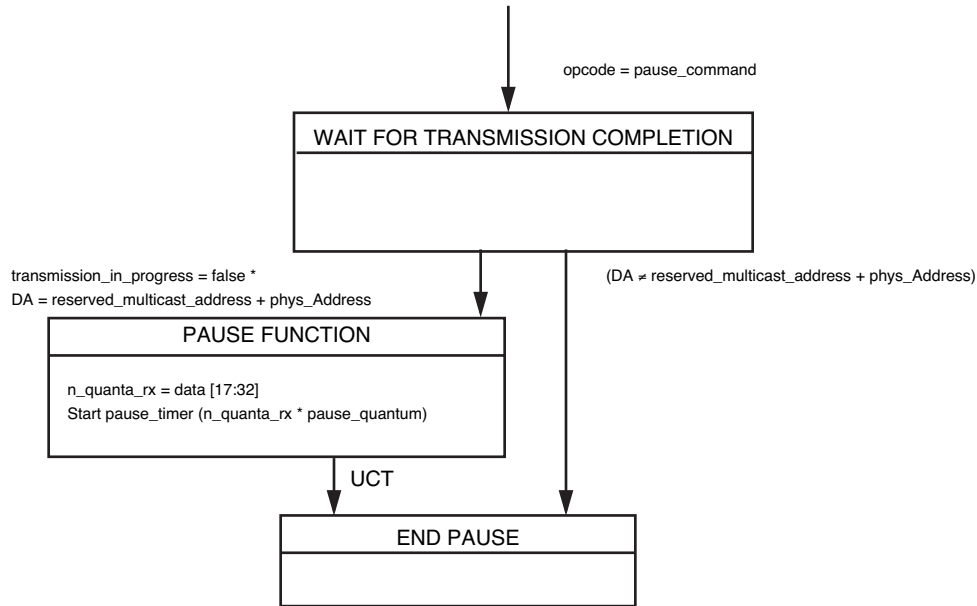
### 31B.3.4.3 Timers

`pause_timer`

The timer governing the inhibition of transmission of data frames.

### 31B.3.4.4 Receive state diagram (INITIATE\_MAC CONTROL FUNCTION) for PAUSE operation

Figure 31B–2 depicts the INITIATE MAC CONTROL FUNCTION for the PAUSE operation. (See 31.5.3.)



**Figure 31B–2— PAUSE Operation Receive state diagram**

### 31B.3.5 Status indication operation

MAC Control sublayer entities that implement the PAUSE operation shall implement the Indication State machine specified in this subclause.

The PAUSE function sets the `pause_status` variable to one of two values: `paused` or `not_paused`, and indicates this to its client through the `MA_CONTROL.indication` primitive.

### 31B.3.6 Indication state diagram for pause operation

#### 31B.3.6.1 Constants

`pause_command`

The 2-octet encoding of the PAUSE command, as specified in Annex 31A.

#### 31B.3.6.2 Variables

`pause_status`

Used to indicate the state of the MAC Control sublayer for the PAUSE operation. It takes one of two values: `paused` or `not_paused`.

`pause_timer_Done`

A boolean set by the MAC Control PAUSE Operation Receive State Machine to indicate the expiration of a `pause_timer` initiated by reception of a MAC Control frame with a PAUSE opcode.

Values: true; The pause\_timer has expired.  
false; The pause\_timer has not expired.

### 31B.3.6.3 Messages

MA\_CONTROL.indication

The service primitive used indicate the state of the MAC Control sublayer to its client.

### 31B.3.6.4 Indication state diagram for PAUSE operation

Figure 31B–3 depicts the Indication State Diagram for a MAC Control sublayer implementing the PAUSE operation.

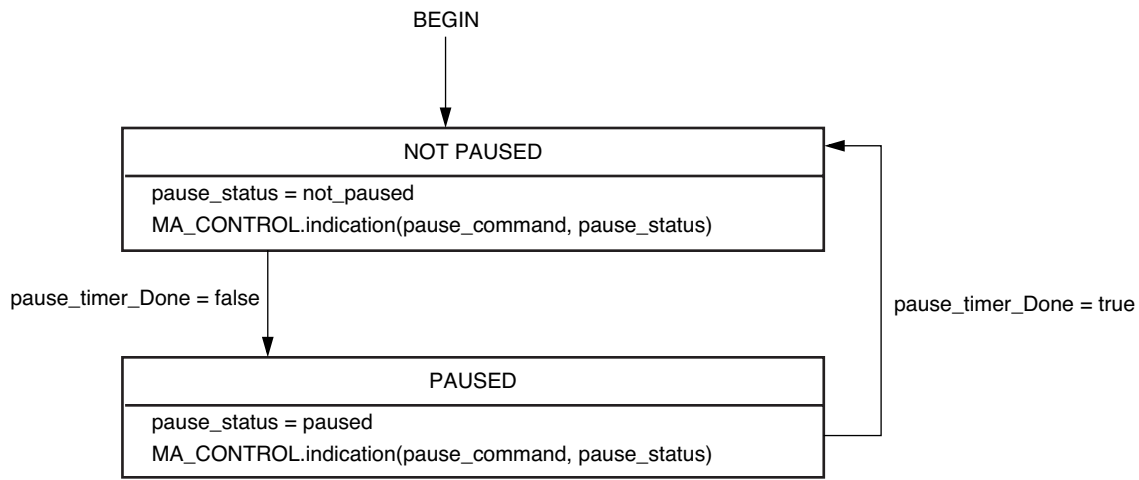


Figure 31B–3—PAUSE Operation Indication state diagram

### 31B.3.7 Timing considerations for PAUSE operation

In a full duplex mode DTE, it is possible to receive PAUSE frames asynchronously with respect to the transmission of Data frames. For effective flow control, it is necessary to place an upper bound on the length of time that a DTE can transmit Data frames after receiving a valid PAUSE frame with a non-zero pause\_time request\_operand.

Reception of a PAUSE frame shall not affect the transmission of a frame that has been submitted by the MAC Control sublayer to the underlying MAC (i.e., the TransmitFrame function is synchronous, and is never interrupted).

At operating speeds of 100 Mb/s or less, a station that implements an exposed MII, shall not begin to transmit a (new) frame (assertion of TX\_EN at the MII, see 22.2.2.3) more than pause\_quantum bit times after the reception of a valid PAUSE frame (de-assertion of RX\_DV at the MII, see 22.2.2.6) that contains a non-zero value of pause\_time. Stations that do not implement an exposed MII, shall measure this time at the MDI, with the timing specification increased to (pause\_quantum + 64) bit times.

At operating speeds above 100 Mb/s, a station shall not begin to transmit a (new) frame more than two pause\_quantum bit times after the reception of a valid PAUSE frame that contains a non-zero value of pause\_time, as measured at the MDI.

In addition to DTE and MAC Control delays, system designers should take into account the delay of the link segment when designing devices that implement the PAUSE operation (see Clause 29).

## 31B.4 Protocol Conformance Statement (PICS) proforma for MAC Control PAUSE operation<sup>23</sup>

### 31B.4.1 Introduction

The supplier of a protocol implementation that is claimed to conform to Annex 31B, MAC Control PAUSE operation, shall complete the following PICS proforma in addition to the PICS of Clause 31.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in Clause 21.

### 31B.4.2 Identification

#### 31B.4.2.1 Implementation identification

Supplier	
Contact point for queries about the PICS	
Implementation Name(s) and Version(s)	
Other information necessary for full identification— e.g., name(s) and version(s) for machines and/or operating systems; System Names(s)	
<p>NOTE 1—Only the first three items are required for all implementations, other information may be completed as appropriate in meeting the requirements for the identification.</p> <p>NOTE 2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model).</p>	

#### 31B.4.2.2 Protocol summary

Identification of protocol specification	IEEE Std 802.3-2002 <sup>®</sup> , Annex 31B, MAC Control PAUSE operation
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	
<p>Have any Exception items been required? No <input type="checkbox"/> Yes <input type="checkbox"/></p> <p>(See Clause 21; The answer Yes means that the implementation does not conform to IEEE Std 802.3-2002<sup>®</sup>)</p>	

Date of Statement	
-------------------	--

<sup>23</sup>Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

**31B.4.3 Major capabilities/options**

Item	Feature	Subclause	Value/Comment	Status	Support
*PST	Support for transmit of PAUSE frames	31B.3.2	N/A	O	Yes [ ] No [ ]
*MIIa	At operating speeds of 100 Mb/s or less, MII connection exists and is accessible for test.	31B.3.7	N/A	O	Yes [ ] No [ ]
*MIIb	At operating speeds of 100 Mb/s or less, MII connection does not exist or is not accessible for test.	31B.3.7	N/A	O	Yes [ ] No [ ]
*MIIc	At operating speeds above 100 Mb/s	31B.3.7	N/A	O	Yes [ ] No [ ]

**31B.4.4 PAUSE command requirements**

Item	Feature	Subclause	Value/comment	Status	Support
PCR1	Duplex mode of DTE	31B.1	Full duplex	M	Yes [ ]
PCR2	Reception of frames with destination address equal to the multicast address 01-80-C2-00-00-01	31B.1	Enabled	M	Yes [ ]

**31B.4.5 PAUSE command state diagram requirements**

Item	Feature	Subclause	Value/comment	Status	Support
PSD1	Transmit state diagram for PAUSE operation	31B.3.2	Meets requirements of Figure 31B-1	PST: M	N/A [ ] M:Yes [ ]
PSD2	Receive state diagram for PAUSE operation	31B.3.4	Meets requirements of Figure 31B-2	M	Yes [ ]
PSD3	Indication state diagram for PAUSE operation	31B.3.6	Meets requirements of Figure 31B-3	M	Yes [ ]

**31B.4.6 PAUSE command MAC timing considerations**

Item	Feature	Subclause	Value/Comment	Status	Support
TIM1	Effect of PAUSE frame on a frame already submitted to underlying MAC	31B.3.7	Has no effect	M	Yes [ ]
	Delay from receiving valid PAUSE command, with non-zero value for pause_time, to cessation of transmission	31B.3.7	Measured as described		
TIM2	Measurement point for station with MII		Delay at MDI $\leq$ pause_quantum bits	MIIa: M	N/A [ ] M: Yes [ ]
TIM3	Measurement point for station without MII		Delay at MDI $\leq$ (pause_quantum + 64) bits	MIIb: M	N/A [ ] M: Yes [ ]
TIM4	Measurement point for station at greater than 100 Mb/s		Delay at MDI $\leq$ (2 $\times$ pause-quantum) bits	MIIc: M	N/A [ ] M: Yes [ ]

## Annex 32A

(informative)

### Use of cabling systems with nominal differential characteristic impedance of 120 $\Omega$ or 150 $\Omega$

The 100BASE-T2 standard specifies only the use of 100  $\Omega$  link segments for conformance. Since ISO/IEC 11801: 1995 also recognizes 120  $\Omega$  and 150  $\Omega$  cabling, this informative annex specifies the conditions for using cabling systems with nominal characteristic impedances of 120  $\Omega$  and 150  $\Omega$  by 100BASE-T2 conformant stations.

The use of cabling with a characteristic impedance outside the range specified in 32.7.2.2 will generally increase the mismatching effects in the link components if directly coupled to the PHY without a impedance transforming balun and results in the following consequences:

- a) Increased echo due primarily to poorer hybrid performance
- b) Increased cabling attenuation roughness due to increased reflections
- c) Increased transmitter launch amplitude
- d) Possible non-linearities in transmitter

The effect of increased echo and reflection is more than compensated by use of Category 4 or higher 120  $\Omega$  cabling or 150  $\Omega$  cabling consistent with ISO/IEC 11801 specifications. The increased transmitter launch level will not be an additional noise or EMC problem on these quality cabling. Manufacturers intending to support a direct connection of 120  $\Omega$  or 150  $\Omega$  cabling to the 100BASE-T2 PHY must ensure that the transmitters can tolerate the impedance without additional non-linearity.

If baluns are used instead of a direct connection, care must be taken to ensure that the corner frequencies of the baluns do not substantially interfere with the 100BASE-T2 signals. In practice, a lower corner frequency less than 100 kHz and an upper corner frequency greater than 30 MHz should be adequate for use with Category 4 or higher 120  $\Omega$  cabling or 150  $\Omega$  cabling consistent with ISO/IEC 11801 specifications.

**CAUTION**—Users of the present annex are further advised to check with the manufacturer of the particular 100BASE-T2 devices they intend to use with a 120  $\Omega$  or 150  $\Omega$  link whether those devices can operate correctly on cabling with  $Z_c$  as high as 120  $\Omega \pm 15 \Omega$  and 150  $\Omega \pm 15 \Omega$ .