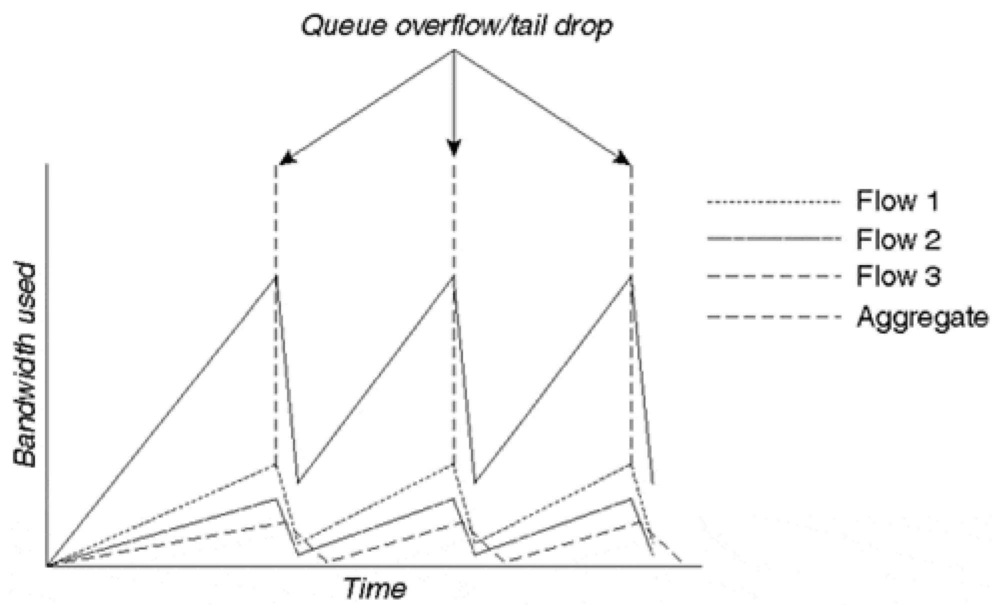
Weighted Random Early Detection

Random early detection (RED) is a congestion avoidance algorithm that is most useful when a majority of the traffic is adaptive, such as TCP. TCP uses packet drops as implicit signals of network congestion, reducing the rate at which traffic is sent (or *backing off*) when a packet is missed.

When a majority of the traffic on a congested link consists of TCP streams from various senders, FIFO queuing tail drops when the queue fills, causing all the TCP senders to back off at the same time. Over a period of time (*t*), the TCP senders ramp up the sending rate, and when the link gets congested again, the TCP senders all back off at the same time. This oscillating behavior is called *global synchronization*, as charted in Figure 8-14.

Figure 8-14. Synchronized TCP Streams



Various congestion management techniques, such as RED, are aimed at preventing this synchronization from occurring. The idea behind RED is quite simple: packets are randomly dropped *before* a queue is full, rather than waiting until it's actually full. If packet drops are spaced out over time, any TCP sessions passing traffic through the queue react to packet loss at different times and they don't synchronize.

RED defines a minimum threshold (min), a maximum threshold (max), and an average queue size (avg). RED computes the average queue size based on the following formula:

$$Avg = (old\_avg * (1 - 1/2^{exponential\text{-}weight\text{-}constant})) + (current\_queue\_size * 1/2^{exponential\text{-}weight\text{-}constant})$$

Note that RED computes an average queue size and not an instantaneous queue size—the average queue size depends on the previous average queue size. Therefore, RED is not biased towards bursty traffic. In other words, RED allows bursts. It typically is not recommended to change the exponential-weight constant.

NOTE

You can find a detailed description of RED in the *Random Early Detection for Congestion Avoidance* paper by Sally Floyd and Van Jacobson (*IEEE/ACM Transaction on Networking*, v. 1, n. 4, August 1993).

RED drops packets based on a probability, which is computed based on the following formulas:

If *avg* < *min*, no packets are dropped.

If *min*< *avg* < *max*, packets are dropped with increased probability as avg increases.

If *avg* > *max*, all packets are dropped.

Going a step further, Weighted Random Early Detection (WRED) weights the drop probability for each packet based on the precedence bits in the IP header, which allows for different classes of service. As with WFQ, there are two WRED implementations available on Cisco routers:

- Platform-independent WRED
- Distributed WRED (dWRED)

Platform-independent WRED is available on all platforms except the Cisco 12000 series of routers; dWRED is available only on Cisco 7500 series routers using VIP processors with dCEF.

Configuration and Monitoring of WRED

WRED is configured with a single command on the interface:

```
Router(config-if)#random-detect
```

To monitor WRED, use **show interface random**, as demonstrated in Example 8-15. The example is the output of dWRED configured on a VIP interface on a 7500.

Example 8-15. *show interface random* Monitors WRED

Router#**show interface random** POS8/1/0 queue size 0 packets output 151051, wred drops 1, nobuffer drops 0 WRED: queue average 0, weight 1/512, max available buffers 23523 Precedence 0: 5880 min threshold, 11761 max threshold, 1/10 mark weight 151043 packets output, drops: 0 random, 0 threshold Precedence 1: 6615 min threshold, 11761 max threshold, 1/10 mark weight (no traffic) Precedence 2: 7350 min threshold, 11761 max threshold, 1/10 mark weight (no traffic) Precedence 3: 8085 min threshold, 11761 max threshold, 1/10 mark weight (no traffic) Precedence 4: 8820 min threshold, 11761 max threshold, 1/10 mark weight (no traffic) Precedence 5: 9555 min threshold, 11761 max threshold, 1/10 mark weight (no traffic) Precedence 6: 10290 min threshold, 11761 max threshold, 1/10 mark weight (no traffic) Precedence 7: 11025 min threshold, 11761 max threshold, 1/10 mark weight (no traffic)

The following list explains the relevant fields from the output of **show interface random**:

- **packets output—**

  Total number of packets transmitted on this interface.

- **wred drops—**

  Packets dropped by WRED.

- **no buffer drops—**

  Indicates the number of packets dropped because there was insufficient packet memory (SRAM) on the VIP.

- **queue average—**

  The average queue length.

- **mark weight—**

  The probability of a packet being dropped when the average queue size is at the maximum threshold. In Example 8-15, 1/512 means one out of every 512 packets is dropped when the average queue is at the maximum threshold.

- **max available buffers—**

  Number of buffers allocated in SRAM for storing packets in the outbound VIP's SRAM.

- **min threshold—**

  Average queue depth at which RED begins to mark (drop) packets for this precedence level. Also represents the number of packets in the RED queue.

- **max threshold—**

  Average queue depth beyond which RED marks (drops) all packets. This value also represents the number of packets in the RED queue.

  NOTE

  The default values for the min and max threshold are calculated based on the value of the output queue length for the interface. Remember, this output queue is not the interface output hold queue, but rather the interface transmit queue on the VIP.

  The min threshold for the precedence queue 0 is set to half the output queue length. The interval between the min-threshold for precedence 0 and the max-threshold is divided into eight values, and each value is assigned as the min-threshold for the remaining precedences. Precedence 0 has the lowest min-threshold and precedence 7 gets the highest min-threshold.

If a majority of the traffic on a link doesn't react to packet drops (UDP, for example), then RED cannot help in any way.

< BACK   Make Note | Bookmark   CONTINUE >

Index terms contained in this section