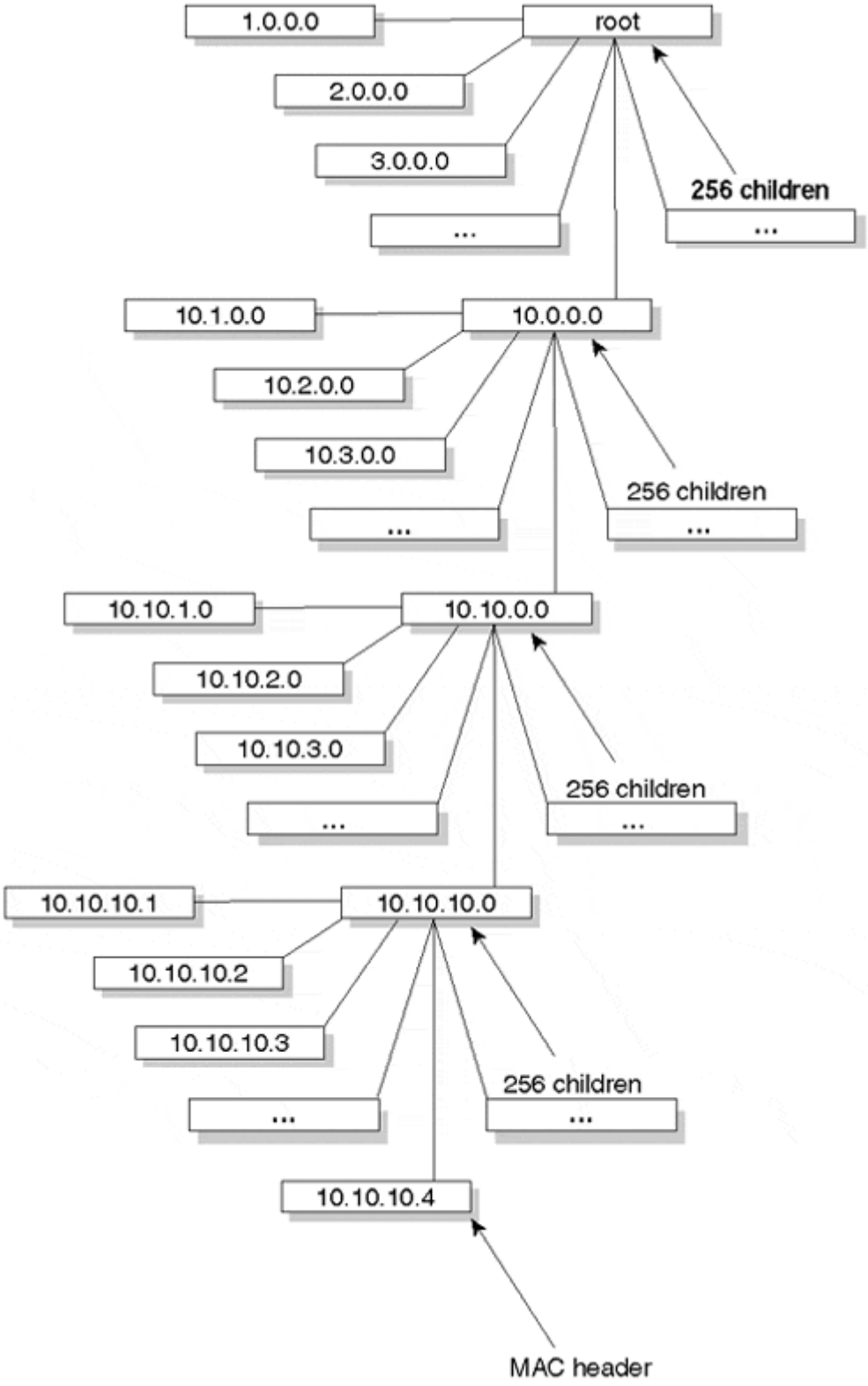
< BACK Make Note | Bookmark CONTINUE >

Optimum Switching

Optimum switching is essentially fast switching with some caching optimizations. Like fast switching, optimum switching accomplishes the entire task of switching a packet during a single interrupt. The primary difference between optimum and fast switching is in the way the route cache is accessed. The optimum switching software is also crafted to take advantage of specific processor architectures. The fast switching code is generic, and is not optimized for any specific processor. Unlike fast switching, optimum switching is available only for the IP protocol.

Earlier in this chapter, you learned the Fast Cache is accessed using a hash table in earlier releases, and is accessed via a two-way radix tree in later releases. The optimum cache is accessed via a *256-way multiway tree* (mtree). Figure 2-9 provides an illustration of a 256-way mtree.

Figure 2-9. The Optimum Cache



Reachability information is stored as a set of nodes, each with 256 children, and the pre-built MAC headers are stored in the nodes. Although this provides faster lookups than the Fast Cache's two-way radix tree, it still has the same limitations as the Fast Cache.

Optimum switching is still similar to fast switching in many ways, including:

- Cache entries are built when the first packet is process switched toward a destination.

- Cache entries are aged and invalidated as the routing table or other cache information changes.
- Load sharing still occurs based strictly on the destination address.
- The same rules are still used to decide what cache entry to build for each particular destination.

The output of **show ip cache optimum**, shown in Example 2-5, is very similar to the output of **show ip cache verbose** in Example 2-3. The primary differences are in the first few lines of output, which provide information about the mtree in which the information is stored.

Example 2-5. Displaying Optimum Cache Entries

router#**show ip cache optimum** Optimum Route Cache 1 prefixes, 1 nodes, 0 leaf refcount, 8K bytes 0 nodes pending, 0 node alloc failures 8 prefix updates, 4 prefix invalidations Prefix/Length Age Interface Next Hop 10.1.1.16/32-24 1w4d Ethernet0 10.1.1.16

The first few lines of information, under the heading **Optimum Route Cache**, note the number of nodes in the mtree, the number of leaf nodes that are referred to by other nodes (because of recursive routing), the memory allocation information, and the number of times the mtree has been updated.

The Optimum Cache mtree doesn't start with all its nodes populated; most of the children under each node will be NULL (or lead to nowhere). As the cache is built through the process switching of packets, it fills up, although you would probably never encounter an Optimum Cache mtree with all the nodes filled.

< BACK Make Note | Bookmark CONTINUE >

Index terms contained in this section