protocols) were often needed to support many customers' internetwork environments.

The number of protocols that are important in new routing products today is much fewer than in the old days. While there is still an installed base of DECnet, AppleTalk, and even some XNS-based systems, the growth of the Internet has made IP the most important Network-layer protocol by far, and many of the protocols that used to be popular have fallen by the wayside. Most corporations, universities, and other institutions are building (or migrating) their enterprise networks to IP-only operation. In addition, most legacy protocols can be encapsulated into IP, making an IP-only routing solution acceptable for many high-speed backbone networks. Thus, a Layer 3 switch may only need to implement hardware-based routing for IP. Other protocols (if needed) can be implemented in software; they are generally required more for connectivity to a shrinking installed base than for performance reasons.[14]

In addition, IP has matured as a protocol. The operation and behavior of the IP routing core is well-defined, and is unlikely to change significantly. Indeed, it would be quite difficult to gain widespread acceptance for any change that caused an incompatibility with the tens-of-millions of installed IP devices.[15] This is an important factor for Layer 3 switching. A traditional software-based router is more amenable to changes and updates without incurring field hardware replacement. With the stability of IP, the risk of hardware implementation is greatly reduced.

### 4.4.2.1 Separating Fast Path Functionality

A router with even a few ports operating at very high data rates must be prepared to handle millions of packets per second. Enterprise routers supporting moderate-to-large numbers of ports operating at gigabit data rates and higher need to process tens-to-hundreds of millions of packets per second in real-time. However, most Network-layer protocols provide many features and functions that either are rarely used (e.g., routing options) or that can be performed in the background of high-speed data forwarding (routing protocol operation, performance monitoring, etc.). A complete IP routing implementation (including all of the necessary functionality and support protocols) is impractical in hardware today; fortunately, it is also unnecessary. A router does not need to be able to perform wire-speed routing when infrequently-used options are present in the packet. Since these boundary cases generally comprise only a small fraction of the total traffic, they can be handled as exception conditions. Similarly, there is no need to provide (and pay for) high performance for housekeeping and support functions such as ICMP, SNMP, and so on. The switch architecture can be optimized for those functions that must be performed in real-time, on a packet-by-packet basis, for the majority of packets, known as the *fast path* of the flow.[16] A Layer 3 switch only needs to implement this fast path in hardware. For background tasks, or exception conditions that must only be dealt with on an occasional basis, it is both easier and less expensive to use a traditional software implementation.[17]

**Seifert's Law of Networking #12**

Optimize for the typical case, not the boundary case.

### 4.4.2.2 The IP Fast Path

What are those functions of the protocol that are in the fast path? This varies somewhat from protocol-to-protocol, but for the purpose of this discussion we will consider the case of IP unicast traffic, because:

- IP is the most widely used protocol suite in enterprise networks today.
- IP comprises a superset of the functionality of popular connectionless network protocols; that is, most other protocols incorporate a subset of the capabilities of IP. The IP fast path is therefore the most complex that needs to be investigated.
- IP multicast traffic currently comprises a small fraction of the total traffic on most IP internetworks, and therefore does not currently justify fast path handling.[18]

[14] After IP, IPX and AppleTalk are the most widely deployed protocols in enterprise internetworks. Some commercial Layer 3 switches do support IPX routing in hardware in addition to IP.

[15] While there is considerable activity in the area of new routing protocols, multicast operation, resource reservation (RSVP), and so on, the *core* functionality of an IP router is quite stable. The operations required to perform packet parsing, routing table lookup, lifetime control, fragmentation, and so on, are unlikely to change and can be committed to silicon with little risk.

[16] The term *fast path* comes from the way protocol processing software is typically designed. The code thread that is traversed most often is scrutinized and optimized most by the programmer, as it has the greatest effect on system performance. Packets that do not deviate from the typical (i.e., they generate no exception conditions) receive the highest performance because they require fewer instructions to process (this code path is executed faster).

[17] This approach does have one nasty pitfall; there is a class of security attack that can be mounted by sending high volumes of traffic that the attacker knows will traverse the slow path, with the intent to overload the processor executing the exception condition software. Under such overload, it is possible that the router may fail altogether or be unable to perform some other important function, allowing an intruder to bypass security control mechanisms and/or avoid detection.

[18] This assumption may change if either voice/video conferencing or streaming multicast video over IP begins to see widespread use. Depending on the switch architecture and the organization of the routing tables, it is actually possible to implement multicast handling in the fast path with little impact on cost or complexity; some Layer 3 switches today already provide this capability. The discussion in the text is confined to the unicast case for simplicity and to avoid restricting the discussion to specific table organizations (e.g., compressed binary trees).

The Switch Book, Rich Seifert, John Wiley

The format of an IP datagram is depicted in Figure 4.6. The Options field(s) is present only if the Header Length field indicates that the header is longer than five 32-bit words (20 bytes). This fact will be useful for separating those packets that contain IP routing options, which do not normally need to be handled in the fast path.

IP addresses are 32-bit, fixed-length fields that comprise two portions:

- A *network identifier*, which indicates the network on which the addressed station resides.
- A *station identifier*, denoting the individual station within the network to which the address refers. IP station identifiers are locally-unique, being meaningful only in the context of the network identified in the network portion of the address.

An example of this separation is shown in Figure 4.7.

Each IP address has associated with it a *subnet mask* of the same length as the address (32 bits). The bits of the address that comprise the network portion are identified by setting the corresponding bits of the subnet mask to 1; the station portion of the IP address is identified by those bits of the subnet mask that are set to 0.

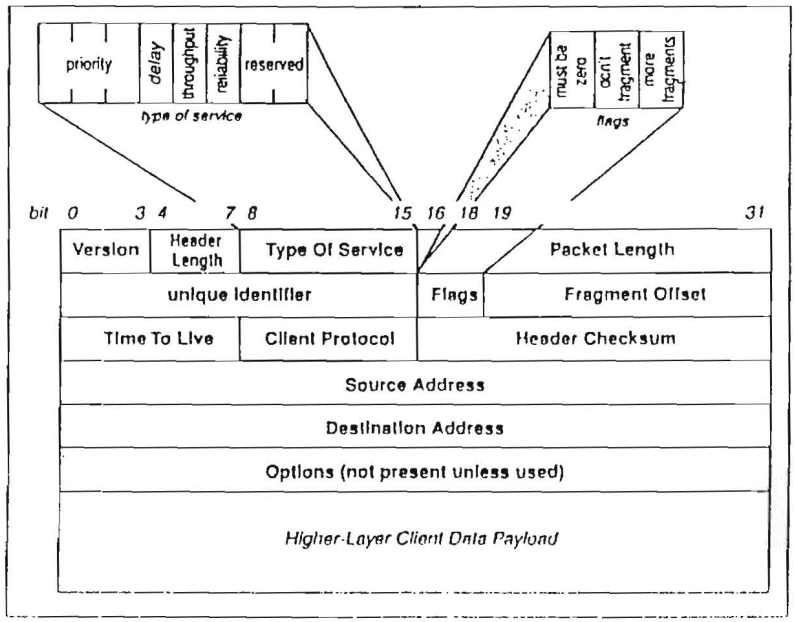


Figure 4.6 IP datagram format.



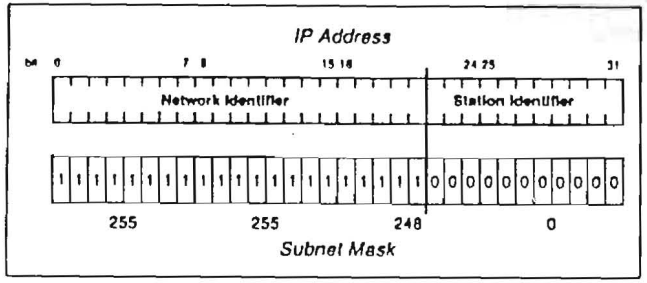


Figure 4.7 IP address format.

**Seifert's Law of Networking #29**

Anyone caught assigning discontiguous subnet masks will be summarily executed.

While not strictly required by IP, the network and station portions of the address generally comprise contiguous strings of bits, with the network portion being the first bits and the station portion the remaining bits. Using this convention, it is unnecessary to actually store subnet masks as 32-bit strings of ones and zeros; all of the relevant information can be provided by a 6-bit value indicating the number of leading bits that comprise the network portion of the address. This condensation can be used to advantage in high-speed routing table lookup operations. Aside from the dependence of some lookup algorithms on this common subnet convention, the use of discontiguous subnet masks can create huge difficulties in administering and managing an enterprise network. In particular, it becomes difficult even to determine which stations belong to the same network from a casual inspection of their addresses. As a result, any deviation from the convention of using contiguous subnet masks is highly discouraged in practice.

As depicted in Figure 4.8, the fast path for unicast IP routing entails:

**Packet parsing and validation** The router needs to separate the various fields in the received packet to determine the type of handling required and to check that the received packet is properly formed for the protocol before proceeding with protocol processing. In the case of IP, this means:

- Checking the protocol version number.
- Checking the header length field. The value must indicate a minimum of five 32-bit words (20 bytes) for a valid IP header; a higher value indicates that IP options are present in the packet.
- Calculating the header checksum.

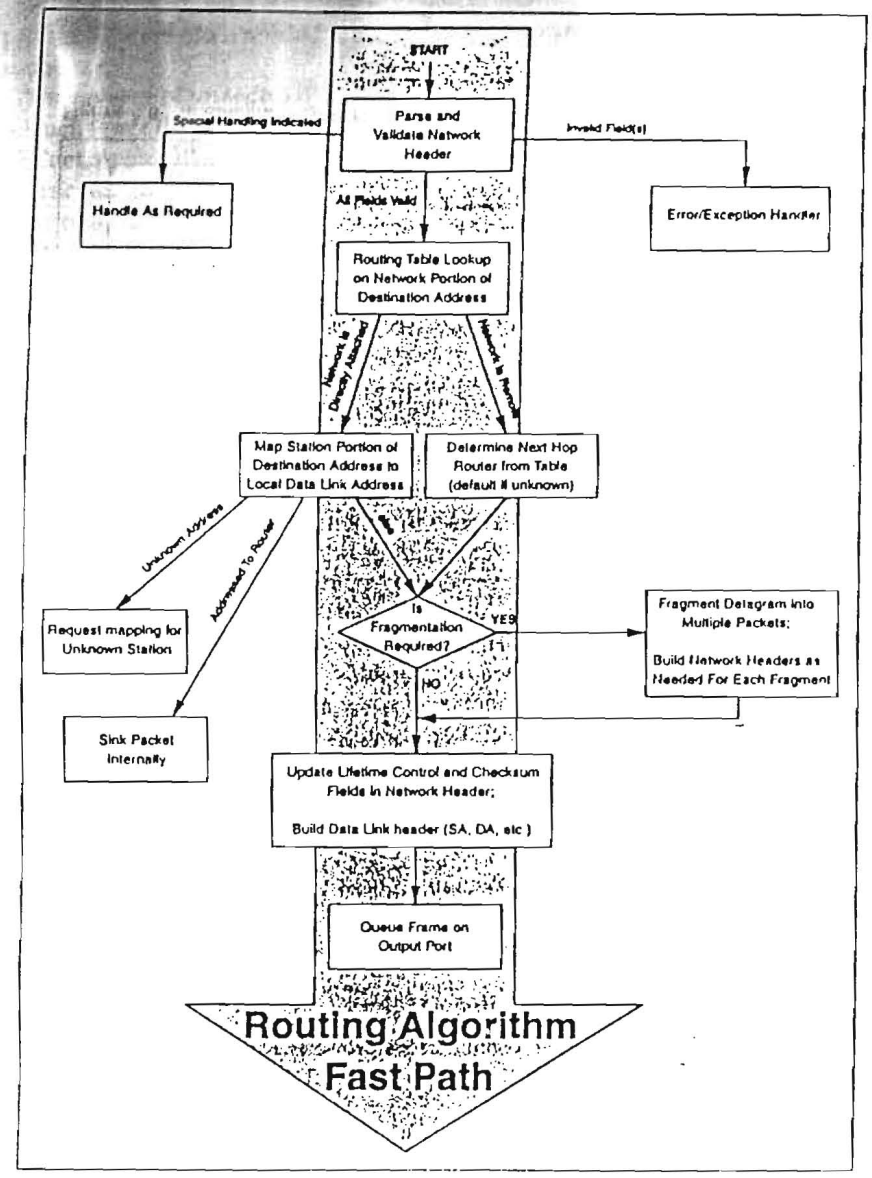


**Figure 4.8** IP fast path.



- Validating the Source Address (e.g., rejecting multicast sources).

Packets with errors can be passed to an error handler that operates outside the fast path. Similarly, packets requiring special handling (e.g., incorporating IP routing options such as source routing or route recording) can also be handled as exception cases outside of the fast path.[19]

**Routing table lookup** The router performs a table lookup to determine the output port onto which to direct the packet, and the next hop along this route, based upon the network portion of the Destination Address in the received packet. The result of this lookup will be that either:

- *The destination network is reachable only by forwarding the packet to another router (remote network).* This may occur due to a match of the destination network against a known table entry, or to the selection of a default route in the event of an unknown destination network. In either case, the lookup will return the address of the next-hop router, and the port through which that router can be reached.[20]
- *The destination network is known to be directly-attached to the router.* The lookup will return the port through which this directly-attached network is reachable, including the possibility of using an internal port (or pseudo-port) for sinking packets addressed to the router itself. For directly-attached networks, an additional step must be taken to map the station portion of the destination address to the data link address for the output port (address resolution using the ARP cache, discussed later).

It should be noted that table lookup in an IP router may be considerably more complex than for a bridge. At the Data Link layer, addresses are 48-bit, fixed-length fields. In addition, the address space is flat; there is no hierarchy or relevant subdivision of the address into distinct parts. Thus, address lookup in a bridge entails searching for an exact match on a fixed length field. This relatively straightforward operation lends itself well to the algorithms and technologies discussed in Chapter 2, *Transparent Bridges* (hash tables, CAMs, etc.).

IP addresses comprise two parts: the network identifier and the station identifier. The routing lookup operation in an IP router is used to deter-

[19] With increased silicon integration, it may be possible to handle these exception cases in fast path hardware as well. However, few IP packets today require option processing. IP options are used primarily for test, diagnostic, and control purposes, and comprise a tiny fraction of the total traffic on most internetworks.

[20] One special case must also be considered, where the destination network is unknown yet there is no default route configured. In this case, the packet will be discarded, and (optionally) an ICMP *Destination Unreachable* message will be sent to the originator.

mine the output port and next-hop data associated just with the network identifier portion of the address. The station identifier portion is examined only in the event that the network lookup indicates that the destination is locally-attached. In all but the simplest IP configurations, the dividing line between the network identifier and the station identifier will not be in a fixed position throughout the internetwork. Routing table entries can exist for network identifiers of various lengths, from 0 (usually signifying a default route) to 32 bits (for host-specific routes). A given destination address may yield a valid match simultaneously against multiple entries in the routing table, depending on the number of bits being considered. According to IP routing procedures, the lookup result returned should be the one corresponding to the entry that matches the maximum number of bits in the network identifier. Thus, unlike a bridge, where the lookup is for an exact match against a fixed-length field, IP routing lookups imply a search for the longest match against a variable-length field.

Appropriate algorithms for such a search are necessarily more complex than those suitable only for bridging. Many routers use a compressed binary tree (e.g., a radix or PATRICIA tree) data structure, which lends itself well to variable-length searches. In addition, a binary tree may allow the routing table to be integrated with the ARP cache, as discussed later. It may even be possible to incorporate Layer 2 bridge tables within the same data structure; a binary tree permits a combined Layer 2/Layer 3 switch to use one common data structure and lookup engine for both functions. While a pure Layer 2 device would generally not need the added complexity required to support variable length lookups, it costs little or nothing to use the more-powerful mechanism for the simpler bridge table lookups, if it is available.

Many hardware-based Layer 3 switches implement the lookup engine as a finite-state machine, with the data structure stored in RAM. Some semiconductor manufacturers produce merchant silicon products specifically designed for routing table lookup in IP routers, either as state machines or as content-addressable memories (CAM).

**Mapping the destination to a local Data Link address (ARP mapping)** The structure of Network layer addresses in IP does not provide a simple mapping to Data Link addresses for the common case of a Data Link that uses 48-bit addresses (i.e., for an IEEE 802-type LAN). That is, it is not possible to determine the 48-bit Data Link address for a given station solely from the station portion of the IP address. Thus, for packets destined for stations on locally-attached networks (i.e., the case where the router in question comprises the last Network layer hop in the route), we must perform a second lookup operation to find the destination address to use in the Data Link header of the frame encapsulating the forwarded packet. Depending on the organization of the lookup tables, this could be a secondary operation (i.e., independent routing table and ARP cache) or simply a continuation of the lookup operation that determined that the destination network was locally attached.

The result of this final lookup will fall into one of three classes:

1. *The packet is destined for the router itself.* That is, the IP Destination Address (network and station portion combined) corresponds to one of the IP addresses of the router. In this case, the packet must be passed to the appropriate higher-layer entity within the router and not forwarded to any external port.
2. *The ARP mapping for the indicated station is unknown.* In this case, the router must initiate a discovery procedure (ARP request) to determine the mapping. As this may take some time, the router may drop the packet that resulted in the initiation of the discovery procedure. Thus, ARP request generation can be outside the fast path of the routing code. Under steady-state conditions, the router will have a valid mapping available for all currently-communicating stations; the discovery procedure will only need to be invoked upon initiation of a new communication session with a station previously-unheard-from.
3. *The packet is destined for a known station on the directly-attached network.* In this, the most common case, the router successfully determines the mapping from the ARP cache and continues with the routing process.

**Fragmentation** Each available output port has associated with it a *Maximum Transmission Unit* (MTU). The MTU indicates the largest frame data payload that can be carried on the interface; it is generally a function of the particular networking technology in use (Ethernet, Token Ring, PPP, etc.). If the packet being forwarded is larger than the available payload space as indicated by the MTU, the packet must be fragmented into smaller pieces for transmission on this particular network.

Remember that a bridge is unable to fragment frames when forwarding between LANs of dissimilar MTUs, since connectionless Data Links generally have no mechanism for fragment reassembly in the receiver (see Chapter 3, *Bridging between Technologies*). At the Network layer, IP is capable of overcoming this limitation; packets can be subdivided into smaller pieces if needed to traverse a link with a smaller MTU. However, fragmentation is a mixed blessing. While it does provide the means to communicate across dissimilar link technologies, the processing burden to accomplish the fragmentation is significant.

In the case of a non-fragmented packet, the router's job (between packet reception and packet transmission on the output port) comprises