CHAPTER 18

Internet Protocols

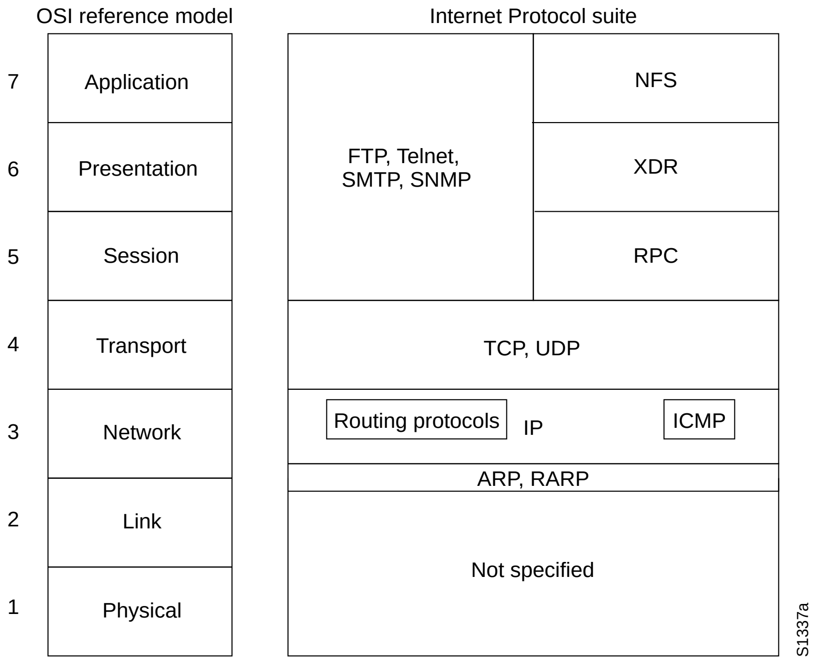
Background

In the mid-1970s, the Defense Advanced Research Projects Agency (DARPA) became interested in establishing a packet-switched network to provide communications between research institutions in the United States. DARPA and other government organizations understood the potential of packet-switched technology and were just beginning to face the problem virtually all companies with networks now have—communication between dissimilar computer systems.

With the goal of heterogeneous connectivity in mind, DARPA funded research by Stanford University and Bolt, Beranek, and Newman (BBN) to create a series of communication protocols. The result of this development effort, completed in the late 1970s, was the Internet Protocol suite, of which the *Transmission Control Protocol* (TCP) and the *Internet Protocol* (IP) are the two best known.

The Internet protocols can be used to communicate across any set of interconnected networks. They are equally well suited for local-area network (LAN) as well as wide-area network (WAN) communications. The Internet suite includes not only lower-layer specifications (like TCP and IP), but also specifications for such common applications as mail, terminal emulation, and file transfer. Figure 18-1 shows some of the more important Internet protocols and their relationship to the OSI reference model.

Figure 18-1 Internet Protocol Suite and the OSI Reference Model

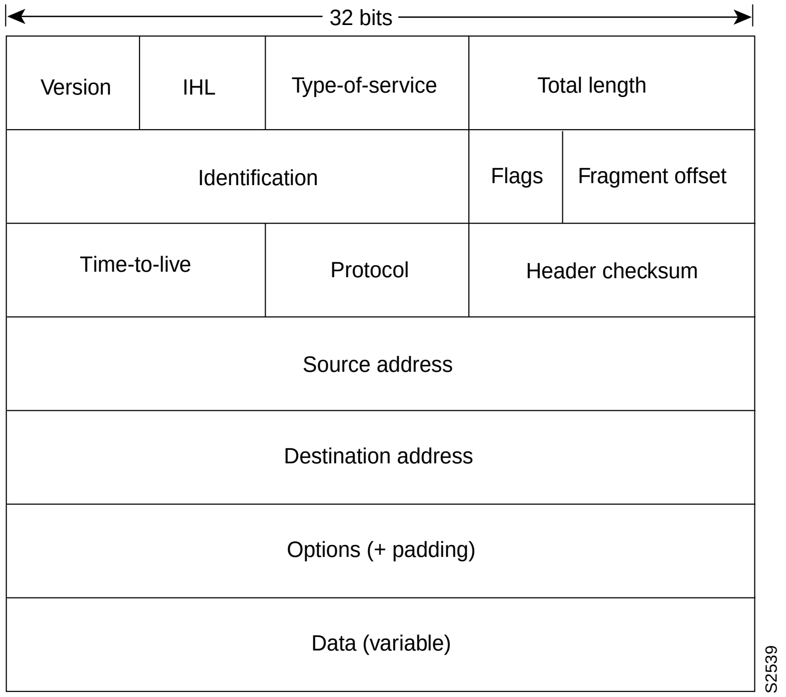


Creation and documentation of the Internet Protocol suite closely resemble an academic research project. The protocols are specified in documents called *Request for Comments* (RFCs). RFCs are published and then reviewed and analyzed by the Internet community. Protocol refinements are published in new RFCs. Taken together, the RFCs provide a colorful history of the people, companies, and trends that shaped the development of what is today the world's most popular open-system protocol suite.

Network Layer

IP is the primary Layer 3 protocol in the Internet suite. In addition to internetwork routing, IP provides fragmentation and reassembly of datagrams and error reporting. Along with TCP, IP represents the heart of the Internet Protocol suite. The IP packet format is shown in Figure 18-2.

Figure 18-2 IP Packet Format



The fields of the IP packet are as follows:

- *Version*—Indicates the version of IP currently used.
- *IP header length* (IHL)—Indicates the datagram header length in 32-bit words.
- *Type-of-service*—Specifies how a particular upper-layer protocol would like the current datagram to be handled. Datagrams can be assigned various levels of importance through this field.
- *Total length*—Specifies the length of the entire IP packet, including data and header, in bytes.
- *Identification*—Contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.
- *Flags*—A 3-bit field of which the low-order 2 bits control fragmentation. One bit specifies whether the packet can be fragmented; the second bit specifies whether the packet is the last fragment in a series of fragmented packets.
- *Time-to-live*—Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.
- *Protocol*—Indicates which upper-layer protocol receives incoming packets after IP processing is complete.
- *Header checksum*—Helps ensure IP header integrity.
- S*ource address*—Specifies the sending node.
- *Destination address*—Specifies the receiving node.
- *Options*—Allows IP to support various options, such as security.
- *Data*—Contains upper-layer information.

Addressing

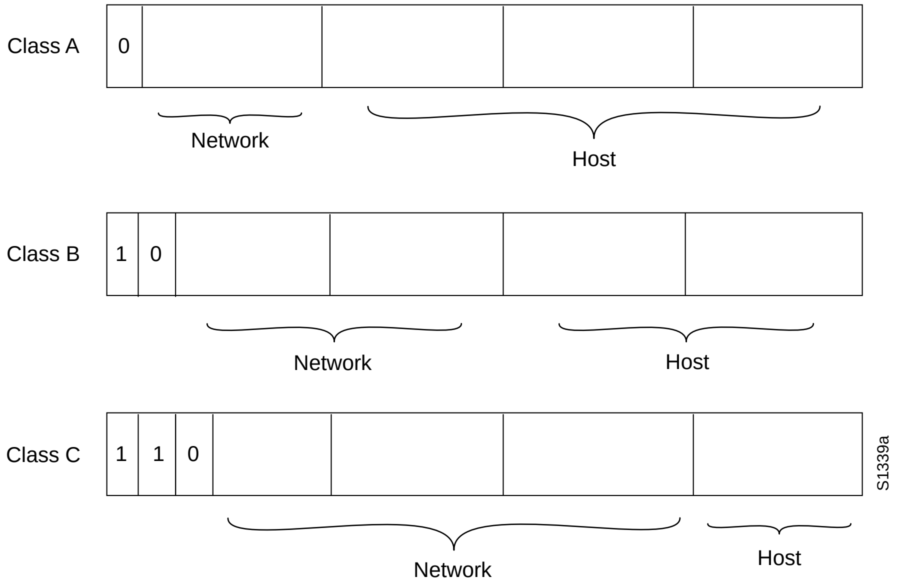
As with all network-layer protocols, the addressing scheme is integral to the process of routing IP datagrams through an internetwork. An IP address is 32 bits in length, divided into either two or three parts. The first part designates the network address; the second part (if present) designates the subnet address; and the final part designates the host address. Subnet addresses are only present if the network administrator has decided that the network should be divided into subnetworks. The lengths of the network, subnet, and host fields are all variable.

IP addressing supports five different network classes. The far left bits indicate the network class.

- *Class A* networks are intended mainly for use with a few very large networks because they provide only seven bits for the network address field.
- *Class B* networks allocate 14 bits for the network address field and 16 bits for the host address field. This address class offers a good compromise between network and host address space.
- *Class C* networks allocate 22 bits for the network address field. Class C networks provide only 8 bits for the host field, however, so the number of hosts per network may be a limiting factor.
- *Class D* addresses are reserved for multicast groups, as described formally in RFC 1112. In class D addresses, the four highest-order bits are set to 1, 1, 1, and 0.
- *Class E* addresses are also defined by IP but are reserved for future use. In class E addresses, the four highest-order bits are all set to 1.

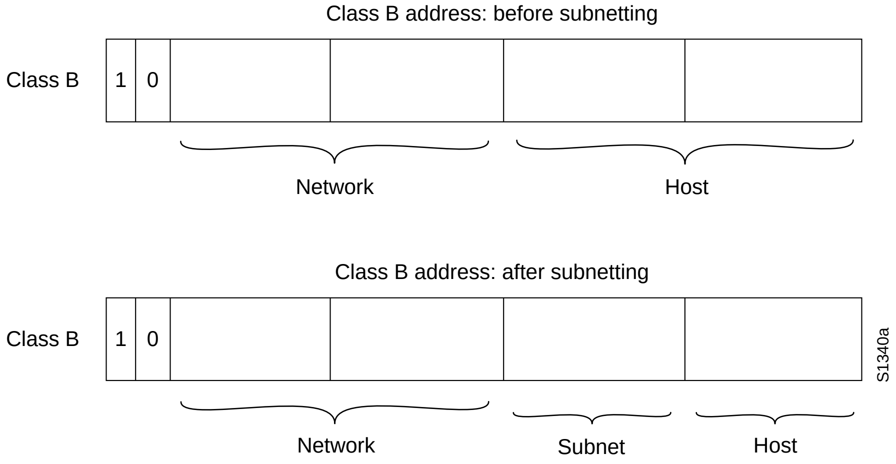
IP addresses are written in dotted decimal format—for example, 34.10.2.1. Figure 18-3 shows the address formats for class A, B, and C IP networks.

Figure 18-3 Class A, B, and C Address Formats



IP networks can also be divided into smaller units, called *subnets*. Subnets provide extra flexibility for network administrators. For example, assume that a network has been assigned a class B address, and all the nodes on the network currently conform to a class B address format. Then assume that the dotted decimal representation of this network's address is 128.10.0.0 (all zeros in the host field of an address specifies the entire network). Rather than change all the addresses to some other basic network number, the administrator can subdivide the network using subnetting. This is done by borrowing bits from the host portion of the address and using them as a subnet field, as shown in Figure 18-4.

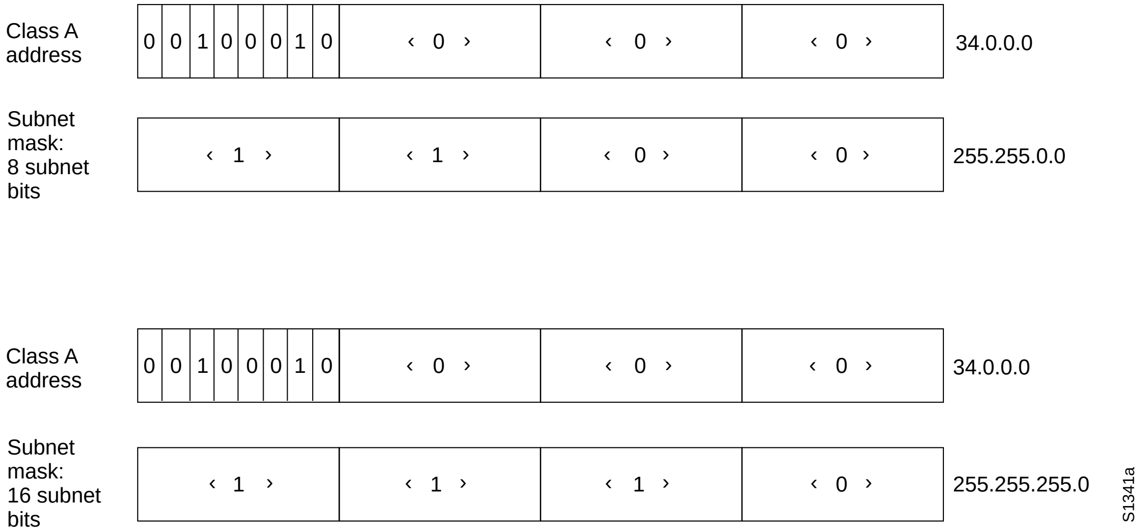
Figure 18-4 Subnet Addresses


Class B address: before subnetting
Class B
1
0
Network
Host
Class B address: after subnetting
Class B
1
0
S1340a
Network
Subnet
Host

If a network administrator has chosen to use 8 bits of subnetting, the third octet of a class B IP address provides the subnet number. For example, address 128.10.1.0 refers to network 128.10, subnet 1; address 128.10.2.0 refers to network 128.10, subnet 2; and so on.

The number of bits borrowed for the subnet address is variable. To specify how many bits are used, IP provides the subnet mask. Subnet masks use the same format and representation technique as IP addresses. Subnet masks have ones in all bits except those bits that specify the host field. For example, the subnet mask that specifies 8 bits of subnetting for class A address 34.0.0.0 is 255.255.0.0. The subnet mask that specifies 16 bits of subnetting for class A address 34.0.0.0 is 255.255.255.0. Both of these subnet masks are shown in Figure 18-5.

Figure 18-5 Sample Subnet Mask

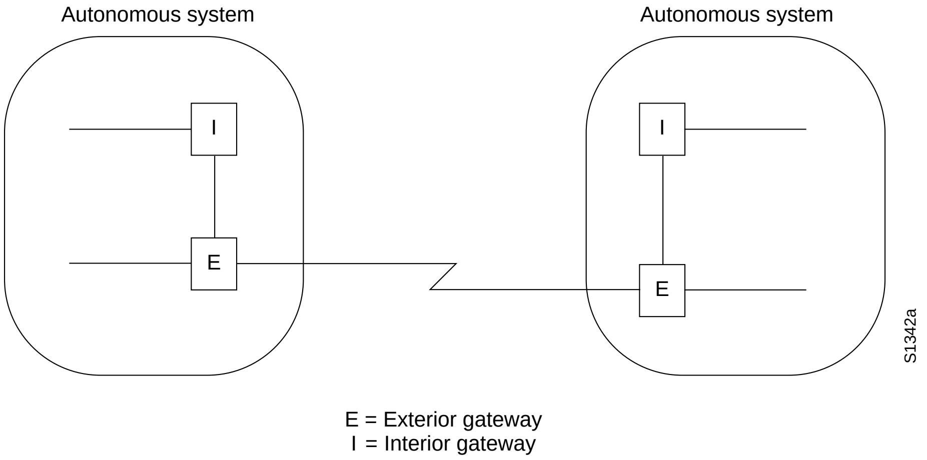


On some media (such as IEEE 802 LANs), media addresses and IP addresses are dynamically discovered through the use of two other members of the Internet protocol suite: the *Address Resolution Protocol* (ARP) and the *Reverse Address Resolution Protocol* (RARP). ARP uses broadcast messages to determine the hardware Media Access Control (MAC)-layer address corresponding to a particular internetwork address. ARP is sufficiently generic to allow use of IP with virtually any type of underlying media-access mechanism. RARP uses broadcast messages to determine the Internet address associated with a particular hardware address. RARP is particularly important to diskless nodes, which may not know their internetwork address when they boot.

Internet Routing

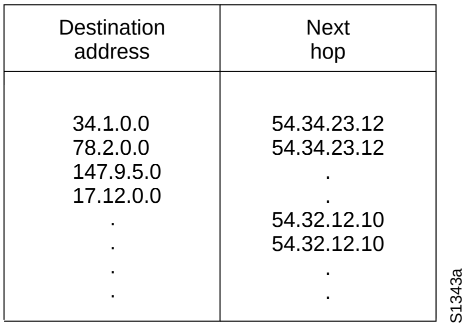
Routing devices in the Internet have traditionally been called *gateways*—an unfortunate term because, elsewhere in the industry, the term applies to a device with somewhat different functionality. Gateways (which we will call routers from this point on) within the Internet are organized hierarchically. Some routers are used to move information through one particular group of networks under the same administrative authority and control (such an entity is called an *autonomous system*). Routers used for information exchange within autonomous systems are called *interior routers*, and they use a variety of *interior gateway protocols* (IGPs) to accomplish this purpose. Routers that move information between autonomous systems are called *exterior routers*, and they use an exterior gateway protocol for this purpose. The Internet architecture is shown in Figure 18-6.

Figure 18-6 Internet Architecture



IP routing protocols are dynamic. *Dynamic routing* calls for routes to be calculated at regular intervals by software in the routing devices. This contrasts with *static routing*, where routes are established by the network administrator and do not change until the network administrator changes them. An IP routing table consists of *destination address/next hop* pairs. A sample entry, shown in Figure 18-7, is interpreted as meaning “to get to network 34.1.0.0 (subnet 1 on network 34), the next stop is the node at address 54.34.23.12.”

Figure 18-7 IP Routing Table



Destination address	Next hop
34.1.0.0	54.34.23.12
78.2.0.0	54.34.23.12
147.9.5.0	.
17.12.0.0	.
.	54.32.12.10
.	54.32.12.10
.	.
.	.

IP routing specifies that IP datagrams travel through internetworks one hop at a time. The entire route is not known at the outset of the journey. Instead, at each stop, the next destination is calculated by matching the destination address within the datagram with an entry in the current node's routing table. Each node's involvement in the routing process consists only of forwarding packets based on internal information, regardless of whether the packets get to their final destination. In other words, IP does not provide for error reporting back to the source when routing anomalies occur. This task is left to another Internet protocol, the *Internet Control Message Protocol* (ICMP).

ICMP

ICMP performs a number of tasks within an IP internetwork. The principal reason it was created was for reporting routing failures back to the source. In addition, ICMP provides helpful messages such as the following:

- *Echo* and *reply* messages to test node reachability across an internetwork
- *Redirect* messages to stimulate more efficient routing
- *Time exceeded* messages to inform sources that a datagram has exceeded its allocated time to exist within the internetwork
- *Router advertisement* and *router solicitation* messages to determine the addresses of routers on directly attached subnetworks

A more recent addition to ICMP provides a way for new nodes to discover the subnet mask currently used in an internetwork. All in all, ICMP is an integral part of any IP implementation, particularly those that run in routers.

Other chapters of this publication discuss specific IP routing protocols. RIP is discussed in Chapter 23, "Routing Information Protocol." IGRP is discussed in Chapter 24, "Interior Gateway Routing Protocol and Enhanced IGRP." OSPF is discussed in Chapter 25, "Open Shortest Path First." EGP is discussed in Chapter 26, "Exterior Gateway Protocol." BGP is discussed in Chapter 27, "Border Gateway Protocol." The Intermediate System-to-Intermediate System (IS-IS) protocol is discussed in Chapter 28, "OSI Routing."

IRDP

The *ICMP Router Discovery Protocol* (IRDP) uses router advertisement and router solicitation messages to discover addresses of routers on directly attached subnets.

The way IRDP works is that each router periodically multicasts router advertisement messages from each of its interfaces. Hosts discover the addresses of routers on the directly attached subnet by listening for these messages. Hosts can use router solicitation messages to request immediate advertisements, rather than waiting for unsolicited messages.

IRDP offers several advantages over other methods of discovering addresses of neighboring routers. Primarily, it does not require hosts to recognize routing protocols, nor does not it require manual configuration by an administrator.

Router advertisement messages allow hosts to discover the existence of neighboring routers, but not which router is best to reach a particular destination. If a host uses a poor first-hop router to reach a particular destination, it receives a redirect message identifying a better choice.

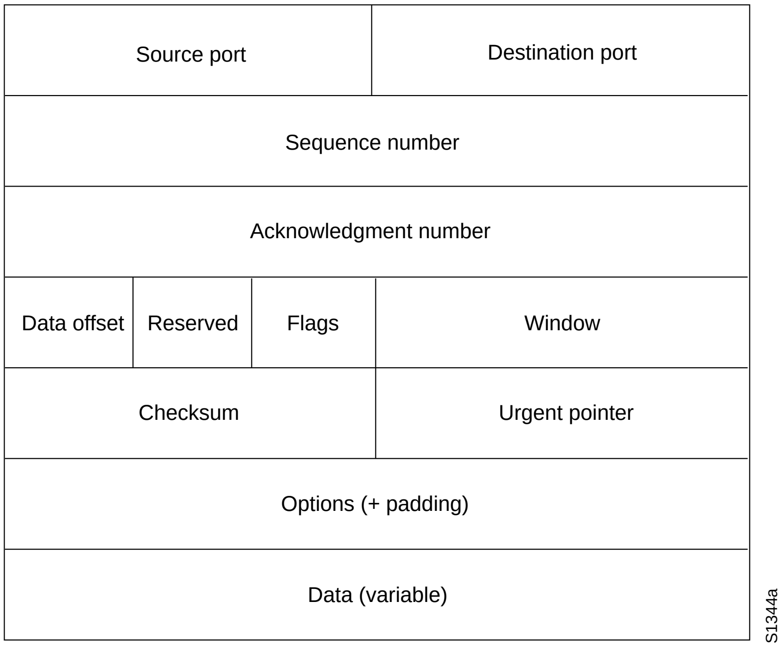
Transport Layer

The Internet transport layer is implemented by TCP and the *User Datagram Protocol* (UDP). TCP provides connection-oriented data transport, while UDP operation is connectionless.

Transmission Control Protocol (TCP)

TCP provides full-duplex, acknowledged, and flow-controlled service to upper-layer protocols. It moves data in a continuous, unstructured byte stream where bytes are identified by sequence numbers. TCP can also support numerous simultaneous upper-layer conversations. The TCP packet format is shown in Figure 18-8.

Figure 18-8 TCP Packet Format



The fields of the TCP packet are as follows:

- *Source port* and *destination port*—Identify the points at which upper-layer source and destination processes receive TCP services.
- *Sequence number*—Usually specifies the number assigned to the first byte of data in the current message. Under certain circumstances, it can also be used to identify an initial sequence number to be used in the upcoming transmission.
- *Acknowledgment number*—Contains the sequence number of the next byte of data the sender of the packet expects to receive.
- *Data offset*—Indicates the number of 32-bit words in the TCP header.
- *Reserved*—Reserved for future use.
- *Flags*—Carries a variety of control information.
- *Window*—Specifies the size of the sender's receive window (that is, buffer space available for incoming data).
- *Checksum*—Indicates whether the header was damaged in transit.
- *Urgent pointer*—Points to the first urgent data byte in the packet.
- *Options*—Specifies various TCP options.
- *Data*—Contains upper-layer information.

User Datagram Protocol (UDP)

UDP is a much simpler protocol than TCP and is useful in situations where the reliability mechanisms of TCP are not necessary. The UDP header has only four fields: *source port*, *destination port*, *length*, and *UDP checksum*. The source and destination port fields serve the same functions as they do in the TCP header. The length field specifies the length of the UDP header and data, and the checksum field allows packet integrity checking. The UDP checksum is optional.

Upper-Layer Protocols

The Internet Protocol suite includes many upper-layer protocols representing a wide variety of applications, including network management, file transfer, distributed file services, terminal emulation, and electronic mail. Table 18-1 maps the best-known Internet upper-layer protocols to the applications they support.

Table 18-1 Internet Protocol/Application Mapping

Application	Protocols
File transfer	FTP
Terminal emulation	Telnet
Electronic mail	SMTP
Network management	SNMP
Distributed file services	NFS, XDR, RPC, X Windows

The *File Transfer Protocol* (FTP) provides a way to move files between computer systems. *Telnet* allows virtual terminal emulation. The *Simple Network Management Protocol* (SNMP) is a network management protocol used for reporting anomalous network conditions and setting network threshold values. *X Windows* is a popular protocol that permits intelligent terminals to communicate with remote computers as if they were directly attached. *Network File System* (NFS), *External Data Representation* (XDR), and *Remote Procedure Call* (RPC) combine to allow transparent access to remote network resources. The *Simple Mail Transfer Protocol* (SMTP) provides an electronic mail transport mechanism. These and other network applications use the services of TCP/IP and other lower-layer Internet protocols to provide users with basic network services.

IP Multicast

The Internet Protocol suite was designed for communications between two computers using unicast addresses (that is, an address specifying a single network device). To send a message to all devices connected to the network, a single network device uses a broadcast address. These two forms of addressing have been sufficient for transferring traditional data (such as files and virtual terminal connections).

Now that application developers are trying to deliver the same data (such as the audio and video required for conferencing) to some, but not all, devices connected to the network, another form of addressing is required. The new form of addressing is called *multicast addresses*, and it involves the transmission of a single IP datagram to multiple hosts. This section describes the following techniques for supporting IP multicast addresses:

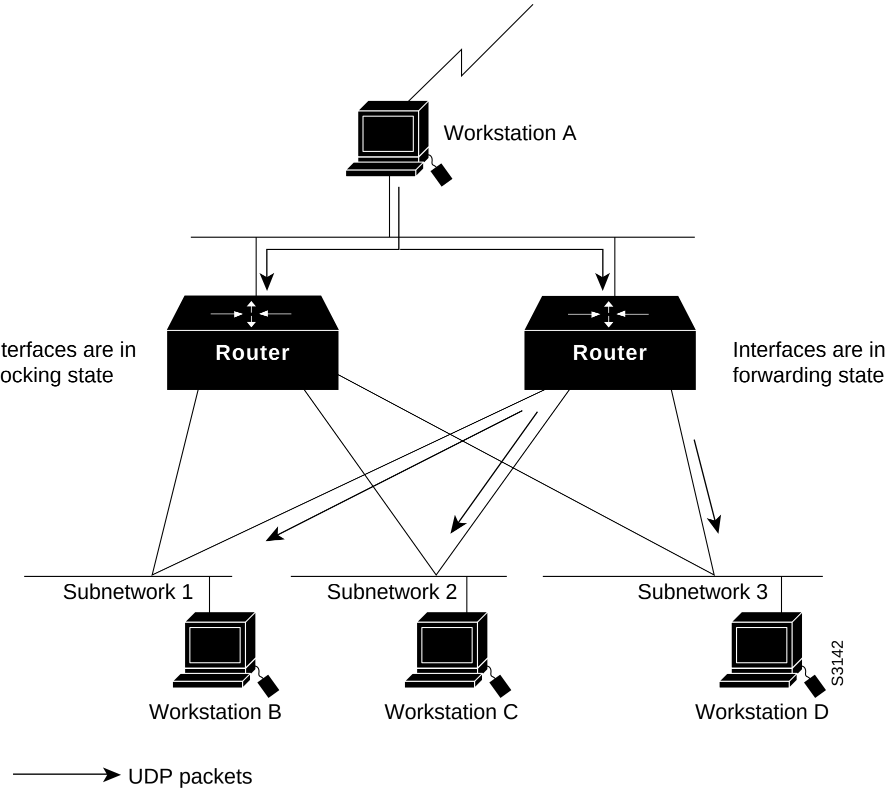
- UDP Flooding
- Subnet Broadcast
- Internet Group Membership Protocol

Because IP networks tend to have complex topologies with alternate paths built in for redundancy, each technique is evaluated for its ability to deliver data without burdening the network with duplicate packets.

UDP Flooding

UDP flooding depends on the spanning tree algorithm to place interfaces in the forwarding and blocking states. By placing certain interfaces in the blocking state, the spanning tree algorithm prevents the propagation of duplicate packets. The router sends specific packets (typically UDP packets) out the interfaces that are in the forwarding state. This technique saves bandwidth by controlling packet flow in topologies that feature redundant routers and alternate paths to the same destination. Figure 18-9 illustrates packet flow.

Figure 18-9 UDP Flooding

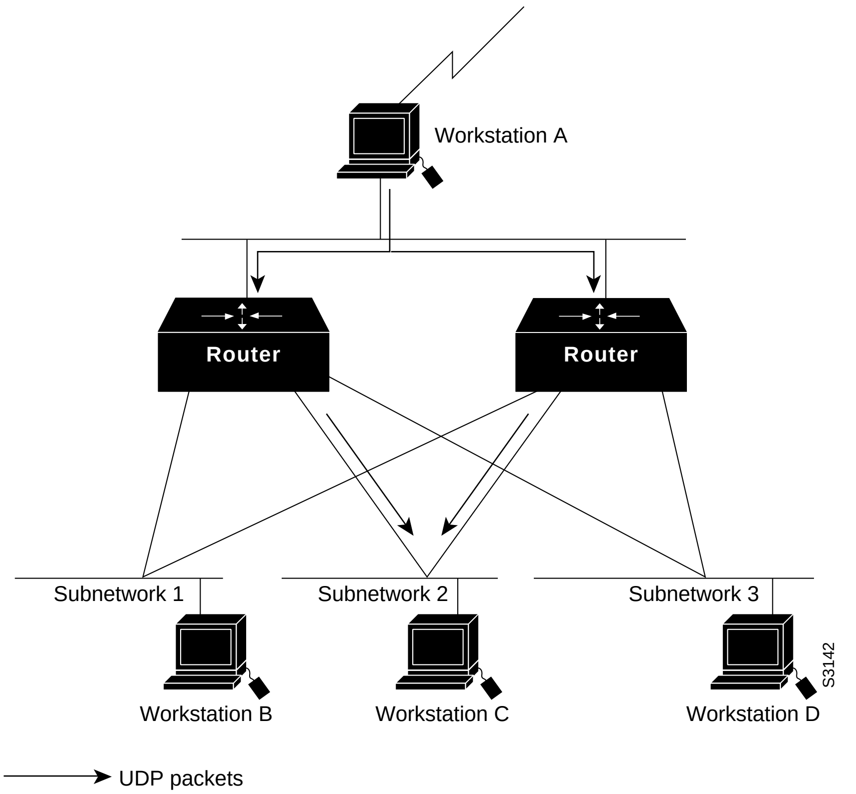


For more information about how the spanning tree algorithm works, see Chapter 29, "Transparent Bridging."

Subnet Broadcast

Subnet broadcast (defined in RFC 922) supports broadcasting to all the subnets of a particular network number. Packet duplication occurs when there are alternative paths in a network. In Figure 18-10, when Workstation A uses subnet broadcasting to send a packet to each workstation on Subnetwork 2, a duplicate packet also arrives.

Figure 18-10 Subnet Broadcast



Whenever there is a duplicate path in the network, a duplicate packet is delivered. Because many multicast applications are data intense, packet duplication is a significant disadvantage of subnet broadcast.

Internet Group Membership Protocol

Internet Group Membership Protocol (IGMP), defined in RFC 1112, relies Class D IP addresses for the creation of multicast groups. By using a specific Class D address, an individual host dynamically registers itself in a multicast group. Hosts identify their group memberships by sending IGMP messages. Traffic is then sent to all members of that multicast group.

Routers listen to IGMP messages and periodically send out queries to discover which groups are active on which LANs. To build multicast routes for each group, routers communicate with each other using one or more of the following routing protocols:

- Distance Vector Multicast Routing Protocol
- Multicast Open Shortest Path First
- Protocol Independent Multicast

These routing protocols are discussed in the following sections.

Distance Vector Multicast Routing Protocol

Distance Vector Multicast Routing Protocol (DVMRP), defined in RFC 1075, uses a technique called *reverse path flooding*. With reverse path flooding, on receipt of a packet, the router floods the packet out all paths except the path that leads back to the source of the packet, which insures that a data stream reaches all LANs. If the router is attached to a LAN that does not want to receive a particular multicast group, the router sends a “prune” message back to the source to stop the data stream. When running DVMRP, routers periodically reflood the network to reach new hosts, using an algorithm that takes into account the frequency of flooding and the time required for a new multicast group member to receive the data stream.

To determine which interface leads back to the source of a data stream, DVMRP implements its own unicast routing protocol. The DVMRP unicast routing protocol is similar to RIP and is based on hop counts only. The path that multicast traffic follows may not be the same as the path that unicast traffic follows.

The need to reflood prevents DVMRP (especially early versions that do not implement pruning) from scaling well. In spite of its limitations, DVMRP is widely deployed in the IP research community. It has been used to build the multicast backbone (MBONE) across the Internet.

The MBONE is used to transmit conference proceedings and deliver desktop video conferencing. Networks that wish to participate in the MBONE dedicate special hosts to the MBONE. The hosts establish tunnels to each other over the IP Internet and run DVMRP over the tunnels. The MBONE is a very high consumer of bandwidth both because of the nature of the traffic (audio and video) and because it is implemented with host-based tunnels. Host-based tunnels tend to result in packet duplication, which the backbone networks transmit unnecessarily.

In addition, the MBONE relies on extremely knowledgeable administrators for support. In spite of their efforts, the MBONE has caused significant disruption to the Internet when popular events or multiple events are active.

Multicast Open Shortest Path First

Multicast Open Shortest Path First (MOSPF) is an extension to OSPF. OSPF is a unicast routing protocol that requires each router in a network to be aware of all available links in the network. Each OSPF router calculates routes from itself to all possible destinations. MOSPF works by including multicast information in OSPF link states. MOSPF calculates the routes for each source/multicast group pair when the router receives traffic for that pair. These routes are cached until a topology change occurs, which requires MOSPF to recalculate the topology.

MOSPF works only in internetworks that are using OSPF and is best suited for environments in which relatively few source/group pairs are active at any one time. MOSPF performance degrades in environments that have many active source/group pairs and in environments in which links are unstable.

Protocol Independent Multicast

Multicast traffic tends to fall into one of two categories: traffic that is intended for almost all LANs (known as *dense*) and traffic that is intended for relatively few LANs (known as *sparse*). Protocol Independent Multicast (PIM) is an Internet draft (under discussion by the IETF Multicast Routing Working Group) that has two modes of behavior for the two traffic types: dense mode and sparse mode. A router that is running PIM can use dense mode from some multicast groups and sparse mode for other multicast groups.

Dense Mode

In dense mode, PIM uses reverse path flooding and is similar to DVMRP. One significant difference between PIM and DVMRP is that PIM does not require a particular unicast protocol to determine which interface leads back to the source of a data stream. Instead, PIM uses whatever unicast protocol the internetwork is using.

Sparse Mode

In sparse mode, PIM is optimized for environments in which there are many data streams but each data stream goes to a relatively small number of the LANs in the internetwork. For this type of traffic, reverse path flooding wastes bandwidth.

PIM-SM works by defining a rendezvous point. When a sender wants to send data, it first sends to the rendezvous point. When a host wants to receive data, it registers with the rendezvous point. Once the data stream begins to flow from the sender, to the rendezvous point, and to the receiver, the routers in the path optimize the path automatically to remove any unnecessary hops, including the rendezvous point.

Comparison of Multicast Routing Protocols

Table 18-2 compares the characteristics of each routing protocol when handling multicast traffic.

Table 18-2 Comparison of Multicast Routing Protocols

Protocol	Unicast Protocol Requirements	Flooding Algorithm	Environment
DVMRP	RIP	Reverse path flooding	Small
MOSPF	OSPF	SPF	Few senders, stable links
PIM-dense mode	Any	RPF	Dense distribution pattern
PIM-sparse mode	Any	None	Sparse distribution pattern