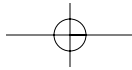


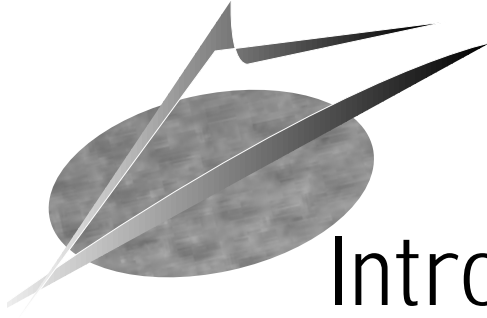


Part I

IP: Architecture, Addressing, and Routing



CHAPTER 1



Introduction to the Internet Protocol

WHAT IS IP?

Despite the growing popularity of the Internet Protocol (IP) as a general purpose networking protocol—as evidenced by the explosive growth of the Internet and corporate “intranets”—there is a comparatively low understanding of exactly what IP does and how all its pieces interact. There are many books about the Internet that talk about chat rooms, how to create snazzy web sites, and other applications that are available to users. This book will address the behind-the-scenes infrastructural aspects of the Internet rather than the surface aspects that users see. You will learn the practical aspects of how IP works. Despite the book’s focus on nitty-gritty infrastructure, this is not a book about communications theory; it is intended to be more practical and solution-oriented, especially for people who need to get up to speed on how IP works. So, what is IP? It is the fundamental packet format that many computers use to exchange information.

What is a packet? It is a well-defined format, usually consisting of a packet “header” followed by some “data,” which could be: 1) portions of files; 2) keystrokes and character echoes within a virtual terminal application; or 3) a portion of an e-mail message. All information that is transmitted over the Internet is broken into

independent packets. Packet switching was developed in the 1960s to provide a robust communications infrastructure, in the context of military applications. Since the information stream is divided into packets, each may follow its own path through the network, thereby avoiding parts of the network that have been blown up or otherwise incapacitated.

This book is not about web browsers, or writing web pages in XML, or how to find things on the Internet. In this book you will learn how to manipulate IP addresses well enough to design an addressing plan for your network; how IP operates over the most common LAN and WAN media, including diagnosing common connectivity problems; and about the functionality of several common nonproprietary routing protocols, including examples on how to use them in your network. Whenever possible, technical information will be illustrated with concrete examples and illustrative exercises.

In any “packet-switched” protocol, two communicating computers break up their data into “packets” that are transported by the “packet-switching network.” There are many packet-switched protocols, but the Internet Protocol alone is the subject of this book. By virtue of its being the building block of the Internet, IP is extremely widespread—and becoming more so every day. Other common packet-switching protocols include the first (international) standards-based packet-switching technology ITU-T’s¹ X.25, IBM’s Systems Network Architecture (SNA), and Novell’s Internetwork Packet Exchange (IPX) protocol. There are others, including AppleTalk Phases I and II, Banyan Vines, ChaosNet, Digital Equipment Corporation’s DECnet Phases IV and V, IP version 6, Open Systems Interconnection (OSI), Unix-to-Unix CoPy (UUCP), Xerox’s Xerox Network Services (XNS), and others.

The Internet Protocol allows data to flow across computer networks, such as the Internet and the many corporate networks that have elected to deploy IP internally. The “data” carried by IP packets can be traditional computer data, or digitized voice and video traffic which are emerging uses of IP. Once voice and video are digitized, they are just data, but they have specific requirements unlike that of, for example, file transfers. Voice and video are time-sensitive and are less tolerant of delay or delay variability. Curiously, voice and video traffic can tolerate some loss of data without experiencing audible or visible degradation, whereas data traffic must be assured of correct transmission, and so time may be spent retransmitting missing or damaged packets to ensure that the entire transmission arrives intact.

In the early IP world, packet switches were commonly known as “gateways,” presumably because they were often served as the gateway between a local campus’



Why Packet Switching?

Why should computers need “packet switching” to communicate? Why not have temporary “phone calls” between computers? The difference comes from the way the phone network is built, and the basic nature of computer communications. Most computer communications are very often brief—on the order of seconds—but intense. The best descriptive word is “bursty.” Compare this to voice calls, which may last a long time, but use a predictable amount of “bandwidth” for the entire duration of the call.

The usual phone network is optimized for voice calls, which tend to last a long time relative to the time it takes to set them up. It may take on the order of five seconds to dial and wait for a remote party to pick up the phone. From the phone network’s perspective, all the hard work of establishing the resources for the call takes place in those first few seconds, after which the call continues to use the resources that have been allocated to it. Due to the extra effort in establishing a call, versus continuing an already established call, the phone company historically charged a penny or two more for the first minute than the remaining minutes.

Another issue with computers calling each other is knowing when to hang up. In a voice call, both parties mutually end a conversation. On the other hand, computers may wish to keep a channel open for a while (how long?) in case one has data to send to the other in the near future. Maintaining a lot of idle but open connections wastes resources in a dedicated-circuit network, like the voice telephone network. Also, a computer would need to have 10 separate physical phone lines if it wanted to have 10 calls active at the same time, which costs money every month whether or not they are used.

Besides the bursty data versus nonbursty voice issue, the rate at which voice calls are set up is bounded by the limitations of human fingers, since we can’t dial quickly enough to overtax the telephone network’s control, or “signaling,” channels. Computers, on the other hand, may need to send brief bits of data to many peers in a short period of time, necessitating far more call setups per second.

(continued)

Packet switching has two main modes of operation, namely “virtual circuit,” also known as “connection-oriented,” versus the other mode, known as “datagram,” also known as “connectionless.” What makes a connection “virtual”? Basically, a number of virtual connections can share a single physical facility, like a single dial-up connection, or a single dedicated serial link to a wide area network (WAN) switch.

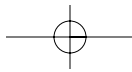
While people have difficulty carrying on multiple simultaneous conversations, computers have no such limitations, and can easily maintain simultaneous “conversations” with multiple peer computers. If each connection needed a unique physical facility, such as its own phone line, then having 10 connections to 10 peers would imply that the calling computer would need 10 physical phone lines (or any suitable dedicated physical medium).

However, when intercomputer communications are broken into packets, each packet can share a single wire, even if one packet may be headed for peer#1, while the next three may be headed for peer#7, and so on. All the “virtual” connections share a single physical wire into a packet-switching network. Each packet’s header tells the network where the packet should go, which relieves the sending computer from needing separate physical connections to each peer. Logically, the connectivity is the same, since the computer can still send the same data to each peer whether or not there is a separate physical link to each peer. However, packet switching can be much more efficient than the alternative, which is known as “circuit switching.”

Remember that there are two fundamental types of packet switching protocols, “virtual circuit” and “datagram.” In a virtual circuit scheme, a call setup still happens at the beginning of a virtual call, but all the previously established virtual connections share the same physical access link to the packet-switching network. This eliminates circuit-switching’s “problem,” wherein multiple connections required multiple physical phone lines. With packet switching, all the virtual connections share the same “phone line” to the packet switch, and each packet has its own destination address that depends on the virtual circuit ID that was assigned during the call setup phase. The virtual circuit ID is a short “label” or “handle,” which allows the packet switches to easily forward the packets to their destinations based on a simple table lookup.

Hanging up is less of an issue here, since an established connection consists simply of a table entry in the packet switches between the two computers. However, given

(continued)



that the packet switches do not have infinite memory, it is good to tear down unneeded virtual circuits so there will be room for others. The memory consumed by established virtual circuits is far less expensive than the bandwidth consumed by idle telephone calls. This is in stark contrast to the situation with physical connections, where the scaling limits are pushed out to the edge; in other words, if a computer has 16 modems, and more than 16 other computers want to send it data, then the physical capacity of that system will be exceeded.

The other main form of packet switching, “connectionless,” encompasses IP and many other network-layer protocols. In connectionless mode, we preserve the “single access link” aspect of the virtual circuit scheme, in that multiple destinations are reachable via the same connection to the packet switching network. However, instead of doing an explicit call setup before any data can be sent, and then using the assigned connection-specific short handle as each packet’s destination address, the packets in a connectionless network all carry the full destination address on every packet. Packets are simply sent into the network, relying on each intermediate packet switch (i.e., IP router) to decide how best to forward the packet toward its indicated destination.

machines and a wide-area network (e.g., the ARPANET). Today, these devices are most commonly known as “routers.”²

Figure 1.1 shows an icon for a router which interconnects five “subnetworks.” A router receives traffic on any of its interfaces, then must decide how to forward it toward its destination, which often involves the packet leaving the router by a different interface. The bidirectional arrows indicate the flow of packets to and from the router.

Routers are the fundamental building blocks of any IP-based network, including the Internet.

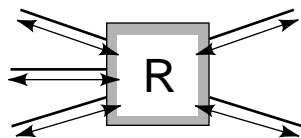
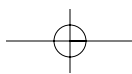
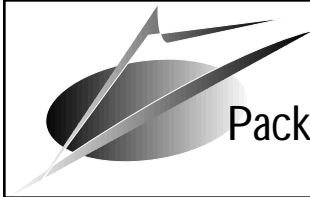


FIGURE 1.1 Representation of a router.





Packet Switching and the ARPANET

Generically, the network elements that move packets around have always been known as “packet switches.” Packet switching was invented by Paul Baran and others in the early 1960s. By 1966, plans were beginning for what would become the ARPANET, which was eventually built in 1969.³ Packet switches in the ARPANET were known as “Information Message Processors,” or IMPs. IMPs were Honeywell 516 minicomputers with 12 KB (!) of memory, running ARPANET-specific packet switching software. The IMP software was developed by Bolt Beranek and Newman, Inc. (BBN), currently part of GTE.

In 1970, the ARPANET began using the Network Control Protocol (NCP) packet format, which was used until the ARPANET was converted to employing the Transmission Control Protocol over the Internet Protocol (TCP/IP) suite beginning on 1 January 1983. The ARPANET researchers designed the Internet Protocol, which was the result of many years of experience with packet switching in the real-world testbed that the ARPANET provided.

The ARPANET gradually grew and evolved into the Internet we know today. Throughout the 1970s and 1980s it experienced considerable growth and supported essential research into computer networking. In the late 1980s, the ARPANET had become just one of the many thousands of interconnected IP-based networks of the Internet. After over two decades of faithful service, the ARPANET was deactivated in 1990.

COMMUNICATING OVER LANS AND WANS

When routers talk to other routers, or to endstations (usually over local area networks or LANs), they need a way to send packets to the neighbor. A packet cannot be sent to a neighbor directly using only its IP address; IP addresses are “higher-layer” addresses. What actually happens is that IP packets are “encapsulated,” or wrapped up, using a frame format that is specific to the subnetwork type. That frame contains the intended neighbor’s destination address, and usually also contains the router’s subnetwork-layer source address. Figure 1.2 depicts the relationship between

Chapter 1 Introduction to the Internet Protocol

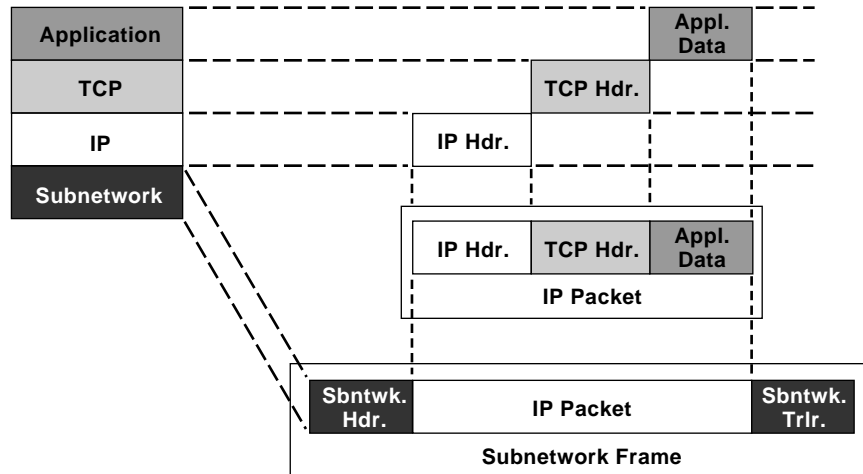


FIGURE 1.2 Layering and encapsulation.

the layers in the IP protocol stack, and the concept of encapsulation. In the diagram the User Datagram Protocol, (UDP) may be exchanged for TCP; they are identical as far as this discussion of layering is concerned.

The LAN and WAN “subnetworks” have their own addressing schemes, which routers must use to communicate with each other. Besides knowing their neighbors’ IP addresses, the routers must usually know their neighbor’s subnetwork layer addresses. IP uses different techniques, specific to each subnetwork medium, to learn what its neighbors’ subnetwork addresses are.

Systems with multiple layers of addresses are commonplace in everyday life. Consider the situation of two people that are in office buildings. One is in the Highgate Tower, suite 2301, and the other is in Trumpet Place, room 3107. Now, in order for them to communicate, they need to know the street address of their correspondent. In real life, each person probably also knows the street address of the other’s building, but that’s not the point. The point is that we deal every day with systems that have multiple levels of addresses, of different formats.

IP ARCHITECTURE OVERVIEW

IP is a layered protocol, designed to facilitate the exchange of data between two applications on two different computers. The fact that the solution is broken into layers reflects a divide-and-conquer approach to the problem of computer-computer communication. In the IP universe, the application is responsible for formatting data

such that its peer(s) can understand it. Applications employ a Transport layer protocol that provides the capability for multiple applications to be running on one machine. Optionally, a Transport layer protocol may provide reliability services, or ordered delivery services. Transport layer protocols may also provide a checksum over the Application-layer data, so that correct reception of unaltered data may be verified.

In the IP stack, the Transport layer offers two very common choices, the Transmission Control Protocol (TCP), which is a reliable transport protocol, and the User Datagram Protocol (UDP), which is a more basic protocol that provides only multiple-application “demultiplexing.”⁴ Both transport protocols consider the application’s data to be “opaque.” In other words, it has no meaning to the transport protocol.⁵ Below the transport layer is the Internet Protocol (IP) layer. IP carries TCP “segments” or UDP “datagrams,” again as opaque data, not knowing anything about the operation of TCP or UDP, much less the application data they are carrying. IP “packets” consist of IP’s header along with the higher-layer transport data “protocols.”

When IP entities need to communicate, they do so by employing any number of lower-layer “subnetwork” technologies. There are either LAN subnetworks (e.g., Ethernet, Token Ring, Arcnet, LocalTalk, etc.) or WAN subnetworks (e.g., static and dynamic point-to-point links, X.25 “clouds,” frame relay clouds, ATM clouds, Switched Multimegabit Data Service (SMDS) clouds, etc.). Figure 1.3 illustrates all the various media over which IP can operate. Routers are used to interconnect the various media; to keep the picture small, the WAN cloud routers do not show LANs that are present at the remote sites.

Each of these subnetworks has its own internal addressing format and framing format. Some subnetwork technologies employ both header and trailer fields, and some encapsulate IP with only a header. Each technology runs at a unique speed, or set of speeds. In short, each is completely different.

In the early days of IP, a tee shirt was produced that proclaimed “IP over Everything!”⁶ Today, rules do exist that describe how IP can run over virtually any subnetwork technology that has ever been invented. Lately, running IP over “IP tunnels” has become useful for Virtual Private Networks (VPNs) over the Internet; in this case, IP is using an IP tunnel as if IP itself were yet another subnetwork layer! Figure 1.4 depicts the layered Internet Protocol “stack,” and compares it to the seven-layer Open Systems Interconnection Reference Model’s layering. The IP model predated the OSI Reference Model (OSI-RM).⁷

Demultiplexing was mentioned above in the context of TCP and UDP, but it is a concept that recurs at every layer of the IP stack, not just at the Transport layer. Multiplexing occurs when multiple higher-layer objects share a common lower-layer

Chapter 1 Introduction to the Internet Protocol

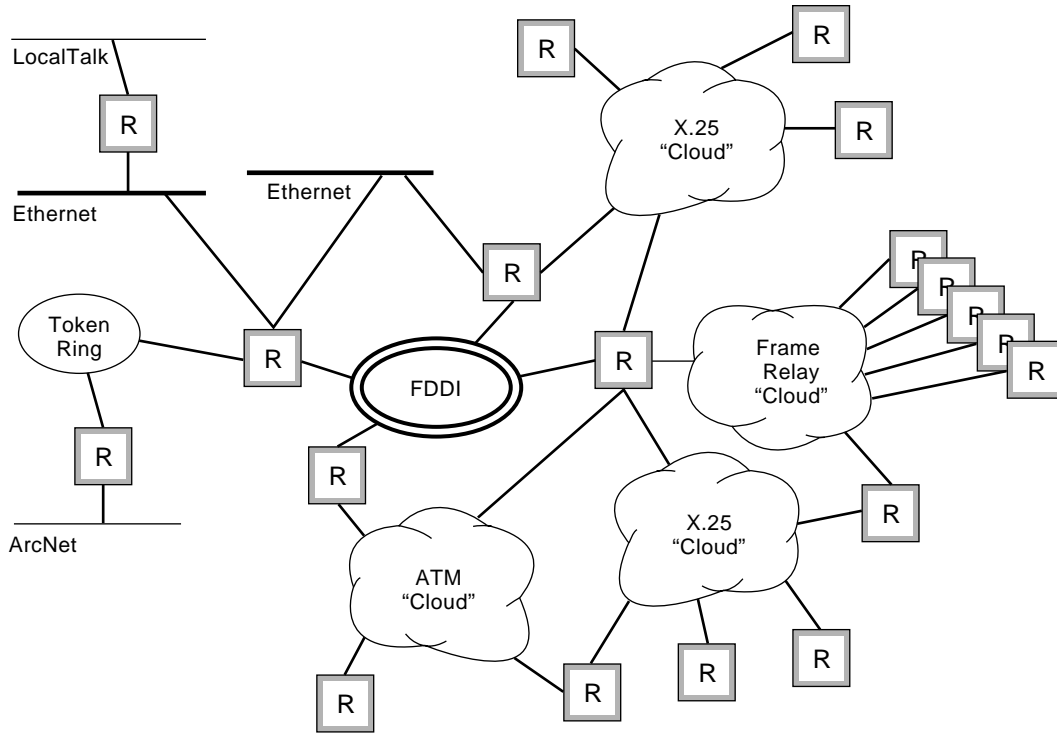


FIGURE 1.3 IP over everything.

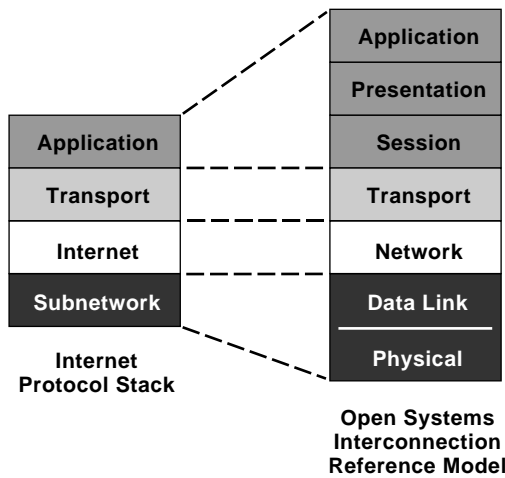


FIGURE 1.4 The layered Internet Protocol stack versus the OSI Reference Model.

facility. Demultiplexing is the process by which lower layers determine which higher-layer entity to deliver some data to. Remember that, ideally, each layer is independent of the others, considering the layer above it to be opaque data. Each layer's header information contains sufficient information so that any decisions regarding the disposition of the opaque payload (the next higher-layer header or data) can be made without needing to peek further inside the packet.

Inbound Packet Processing

In the subnetwork layer, a value in the subnetwork layer header that indicates that IP is the protocol “inside” the frame. At a minimum, there will be a Destination Address and some form of “Protocol Type” in the subnetwork layer header, as indicated in Figure 1.5. The Type field will contain a number X whose value means “the data that begins after the end of this header, and continues until the end of the frame,⁸ belongs to protocol X.” The DA and SA fields represent the subnetwork-layer destination and source addresses.

Each subnetwork may have its own list of values representing Network-layer protocols. For instance, over one important type of LAN, IP's protocol type value is 0x0800⁹ (one of 65,536 possible values of a two-byte Type field), but in a Frame Relay context, the protocol field is 0xCC (one of 256 possible values of a one-byte “Network Layer Protocol Identifier” field).

The subnetwork-layer demultiplexing feature allows multiple protocol stacks to share a common subnetwork medium, or more importantly, for multiple protocol stacks to be active on the same machine at the same time. Think of your PC—you probably have Microsoft[®] NetBEUI, Novell Internetwork Packet eXchange (IPX), and IP all active. For you Mac users out there,¹⁰ you probably have not only AppleTalk, which is Apple's proprietary protocol stack for printing and file server access, but also an IP stack (either MacTCP or OpenTransport).

Whether a PC, Mac, or Unix workstation is being used, all the active protocol stacks share the same Network Interface Card (NIC) subnetwork address, so when

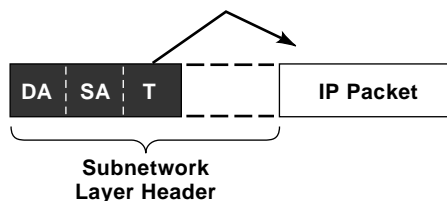


FIGURE 1.5 Subnetwork demultiplexing headers.

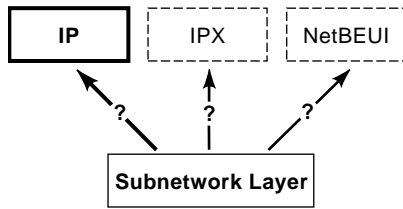
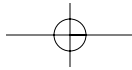


FIGURE 1.6 Passing a packet up the stack.

the NIC receives a frame it is clearly for one of the protocol stacks . . . but which one? The Protocol Type value tells the driver software which protocol stack should get the frame’s embedded packet. Figure 1.6 illustrates the decision-making process that the subnetwork-layer protocol software performs.

Once the IP layer has taken delivery of the packet from the subnetwork layer,¹¹ it must first verify that its locally-assigned address matches the packet’s destination address. If this is the case, then the IP layer has its own set of headers that mimic the demultiplexing function of the subnetwork. Figure 1.7 shows the parts of the IP header that are important for demultiplexing.

An important feature that makes the IP layer unique, and valuable, is the fact that it is “subnetwork-independent,” so it can run over almost any type of subnetwork. IP insulates the Transport layer above it from all the different underlying characteristics of all the many possible subnetworks that it supports. Just as there can be multiple network-layer protocols that share a common subnetwork layer, multiple-Transport layer protocols can share the Internet Protocol layer, as shown in Figure 1.8.

The IP header’s “Protocol” field is the indicator of which higher-layer protocol should receive the data encased within the packet. The most common higher-layer protocols are TCP and UDP (IP Protocol values 6 and 17, respectively). There are also many other protocols that make direct use of IP, including the Internet Control Message Protocol¹² (ICMP, IP Protocol = 1), the Internet Group Management Protocol (IGMP, IP Protocol = 2), the Open Shortest Path First (OSPF, IP Protocol =

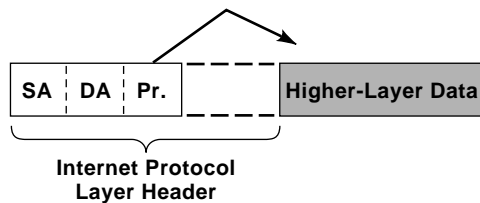


FIGURE 1.7 IP demultiplexing headers.

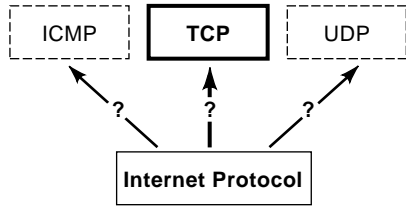
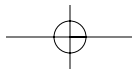


FIGURE 1.8 Passing higher-layer data up the stack.

89) routing protocol, the Protocol-Independent Multicast (PIM, IP Protocol = 103) multicast routing protocol(s), and many others. See the “Assigned Numbers” RFC (RFC-1700, or its successor) for a complete list of the IP Protocol field’s possible values. A key thing to remember here is that just because a higher-layer protocol is a client of IP, it is not necessarily a Transport-layer protocol. In such cases, you might say that an application, or an application-like entity, is running directly over IP, with no intervening Transport layer protocol.¹³

In the case where a Transport layer protocol does follow IP, its header it used to help identify which application needs to receive the data. In the case of TCP, which is used for a majority of IP-based applications, there are a lot of fields that facilitate features other than demultiplexing, i.e., sequence numbers so that segments can be delivered in order and so that gaps in the data can be corrected by requesting retransmission, plus flags that are used when opening and closing connections, and other items. Figure 1.9 shows how TCP aids in the delivery of data to the correct application. In this case, we depict both the abstracted packet format and the protocol stacking in the same figure.

In TCP, the Destination “Port” is the indication of which application should receive the data. The DP field is doing double duty as both the “destination address” field and the “protocol type” field. This is perhaps because TCP is the top layer of the protocol stack.

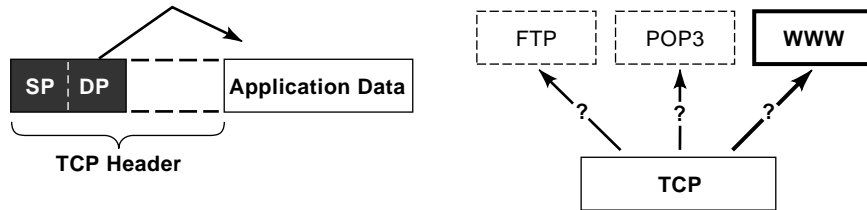


FIGURE 1.9 TCP demultiplexing.

Chapter 1 Introduction to the Internet Protocol

TCP Details

Every TCP “connection” may be represented by a pair of “sockets.” A socket is itself a pair of numbers, the IP address and a TCP port. First, a connection is limited to two machines almost by definition, so we need the two machines’ IP addresses, i.e., IP_a and IP_b . Secondly, the TCP layer identifies the connection by the source and destination port, so there is $Port_A$ and $Port_B$ as well. So, $Socket_1$ is $(IP_a, Port_A)$, and $Socket_2$ is $(IP_b, Port_B)$. A connection is then represented as either $[Socket_1, Socket_2]$, or $[(IP_a, Port_A), (IP_b, Port_B)]$.

In order to start up a new connection, one needs to know not only the other machine’s IP address, but also the port on which the desired application service is “listening” (within that server). Most IP applications make use of “well-known ports” so that the client does not need to look up which port to use. For instance, a client wishing to connect to a WWW server will use destination TCP port 80 by default, but a client wishing to connect to a Post Office Protocol version 3 server will connect to TCP port 110. The client picks a random source port for itself and attempts to open the connection to the desired service’s well-known port.

OVERVIEW OF THE IP HEADER

The IP header consists of a lot more than just demultiplexing and addressing functions. In all its glory, Figure 1.10 shows the IP Header. All the IP header’s fields are useful, but the highlighted ones are more commonly used. Routers generally pay attention only to the Destination Address as long as the header format is the “simple”

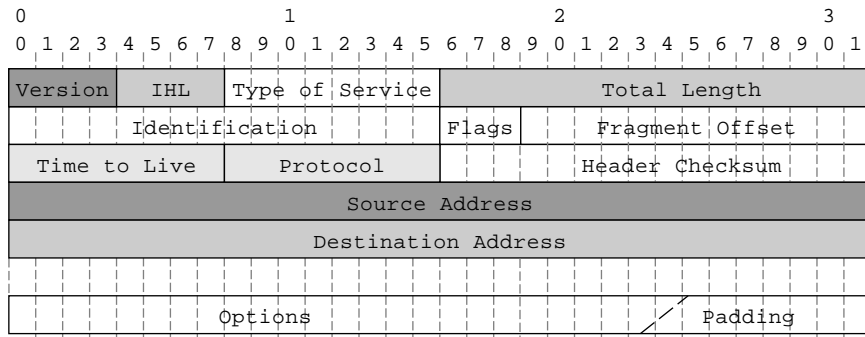


FIGURE 1.10 IP header fields.

one in the figure. The dashed vertical lines delineate bit positions. The header, as drawn, is 32 bits (four bytes) wide.

The most obvious fields are the ones which we are already familiar with, namely the Source and Destination Address fields, and the Protocol field. The other fields are all important, and will be discussed here, in order of relative importance. First, the Total Length field represents the length (in bytes) of the entire IP packet—including the IP header. Since it is a two-byte (16-bit) field, the largest IP packet may be $2^{16} - 1$, or 65,535, bytes long. The Internet Protocol RFC, RFC-791, mandates that all IP endstations be capable of sending and receiving 576-byte packets.

The Internet Protocol header is typically 20 bytes long, though certain “options” have been defined that can be appended to the end of the header. This is represented by the Internet Header Length (IHL) field, which counts the number of four-byte “words” in the IP header. Another way of saying this is that to determine the IHL value, one must take the IP header length and divide it by four. To avoid fractions, the IP header must be padded to a multiple of four bytes. The typical value of the IHL field is five, since five times four is 20. Since the IHL field is four bits long, the maximum value of the field is 15, making the maximum possible IP header 60 bytes long (15 times four bytes).

The Version field is set to four, since this header format is that of IPv4, the version of IP that is in use in today’s Internet.¹⁴ The Type of Service (ToS) byte has been getting a lot of attention lately within the industry, as ISPs and customers clamor for a way to provide different “class of service” levels within their networks. Work is under way to define a “Differentiated Services” (DS) model, that redefines the ToS field as the DS field. The first two RFCs defining differentiated services are RFC-2474 and -2475. Work is ongoing to define new “per-hop behaviors” and also to deploy the initial specifications. RFC-2430 is one example of how differentiated services might be deployed, however, other scenarios are likely to emerge as experience with this new technology is accumulated.

The Time to Live (TTL) field is decremented by one each time a packet crosses a router. This practice ensures that the packet will not persist in the Internet forever. The maximum initial TTL is 255, but many endstations will use a lower initial TTL value. Originally, the TTL was literally supposed to be interpreted as up to 255 seconds of time; presumably, the expectation of early-1980s line speeds and software-based routers was that it would take about one second for a router to receive and forward a packet, but the TTL has evolved into nothing more than a hop counter.

The Header Checksum value must be recomputed on a hop-by-hop basis, since each router hop decrements the TTL, thereby changing the header and invali-

Chapter 1 Introduction to the Internet Protocol

dating the previous hop's calculated checksum. Fragmentation, if performed, also changes the contents of the IP header, thus forcing the checksum to be recomputed.

Fragmentation of IP Packets

This section on IP packet fragmentation is somewhat abstract and can safely be skipped on a first reading; however, a discussion of the IP header would not be complete without covering fragmentation.

Three of the IP header fields are used together to support fragmentation. For example, suppose that a packet is sent by an initial endstation with a Total Length of 1500 bytes. Somewhere along the path to the packet's destination, there may be a link that only supports datagrams up to 512 bytes long. We say that the link's maximum transmission unit (MTU) is 512 bytes.

When faced with such a situation, a router can break a packet into fragments such that the pieces will each be small enough to pass through the narrow link. Each fragment has its own complete IP header, much of it the same as the original packet's header, as illustrated in Figure 1.11. This figure assumes the typical Internet Header Length of 20 bytes.

If options were present in the initial header, some of them must be replicated in each fragment, while others may remain in the first fragment. If any options remain in the IP header, the Internet Header Length will be at least 24 bytes, which would make the packets 512 bytes versus 508 bytes. If the Internet Header Length were 28 bytes or larger, the data portion of the packet would be reduced in multiples of eight bytes to keep the Total Length under the 512-byte MTU.

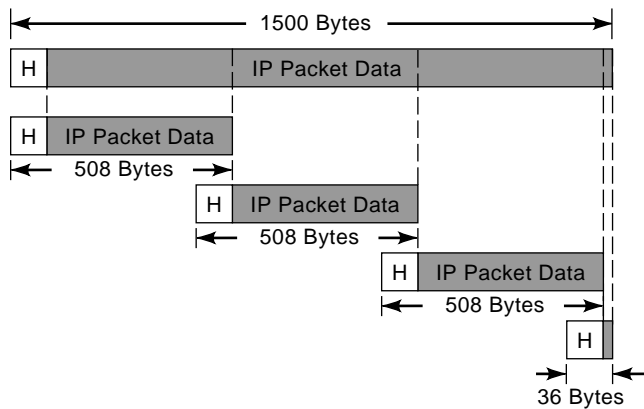


FIGURE 1.11 IP fragmentation.

Each packet fragment is then forwarded independently on its way to the destination, where it is ultimately reassembled. The Identification field is a unique 16-bit value that is used by the transmitting station to help the receiver keep track of related fragments during the reassembly process. It is possible for multiple senders to pick the same one of these 65,536 values, so the receiver must correlate not only the Identification header value, but also the packet's Source Address. Between a given pair of IP addresses, it is unlikely that a sender would pick the same Identification field for two different packets within a small time interval. Senders should increment the Identification field value in each transmitted packet, thereby guaranteeing that the same value will only be used once out of every 65,536 packets sent by any given station.

The Flags field controls the fragmentation process; the most significant of the three bits must be zero, the next one is the "Don't Fragment" bit, and the least significant is the "More Fragments" bit. If the "More Fragments" bit is set (i.e., if the bit is equal to one), then this is not the last fragment. The final fragment will have this bit clear, so the receiving station will know that it is the final fragment of the original packet. The meaning of the "Don't Fragment" bit is clear—the sending station wishes that the packet be carried in one piece, or not at all. If a packet marked with the "Don't Fragment" bit encounters too small a link, the last router before the link will discard the packet and send an error message back to the packet's Source Address.

Once all the packet's fragments have arrived at their Destination Address, the Fragment Offset values in each fragment allow the receiver to put them back together in order. These values also permit the receiver to know when it has received all the fragments, by using the Fragment Offset values for each packet, and the length of each fragment's data field, which is its Total Length minus its Internet Header Length.

Until they all arrive, those fragments that have been received must be buffered, or stored, at the receiver. It is tempting to think that since routers usually fragment packets, that they should re-assemble them as well. This is most definitely not the case! Due to the datagram nature of IP networks, all the fragments may not take the same path through the network. Remember that every IP packet is independently forwarded, based on each router's understanding of the best way to get to the packet's destination when it arrives at that router. A topology or administrative change among the intervening routers could cause some fragments to take a different path than their predecessors. No single "downstream" router is guaranteed to receive all of the fragments,¹⁵ so no single one of them could possibly be expected to re-assemble a fragmented packet. The only sure place where all the fragments ought to reappear is the ultimate destination station.



Fragmentation Minutiae

Each fragment must be a multiple of eight bytes long, because there are only 2^{13} Fragment Offset values, but the packet can be up to 2^{16} bytes long. Since $(2^{16})/(2^{13})$ is 2^3 , the fragments are forced to be multiples of eight bytes long. Note that the requirement that packet fragments be a multiple of eight bytes long *only applies to the original packet's data field*. A packet fragment's data field length must be a multiple of eight, but its Total Length will never be a multiple of eight, unless there are header options that make the IP header a multiple of eight as well.

Remember that the Total Length of any packet includes the length of the IP header. Since IP fragments are still IP packets in their own right, this is still the case. The Internet Header Length is always a multiple of four bytes long, between 20 and 60, inclusive, i.e., 20, **24**, 28, **32**, 36, **40**, 44, **48**, 52, **56**, or 60. Those IP header lengths that are multiples of eight bytes long are in boldface type. If the original packet had no options, which is the typical case, then all the packet's fragments will have an Internet Header Length of 20, meaning the Total Length of all the fragments will never be a multiple of eight.

Returning to Figure 1.10, one sees that each packet is less than 512 bytes long (the small link's maximum transmission unit). This is because the largest multiple of eight that can fit inside the 512-byte limit is 488, when one takes account of the fact that the header makes the packet another 20 bytes longer, for a Total Length of 508 bytes. The next higher multiple of eight is 496, but $496+20 = 516$, which would be too large for the link. The minimum size for a fragment is eight bytes of data, plus 20 bytes of header, or 28 bytes, up to 68 bytes, where the eight bytes of fragment are preceded by a maximum-length 60-byte IP header.

Now, consider the largest IP packet, with a Total Length of 65,535. The data portion of that packet is a variable number of bytes long, depending on the size of the packet's header. In Table 1.1, we see the header lengths versus the resulting data lengths, assuming that each row must add to the maximum Total Length—65,535.

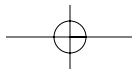
(continued)

TABLE 1.1 VALID DATA SIZES OF A 65,535-BYTE IP PACKET

IHL	65535-IHL
20	65,515
24	65,511
28	65,507
32	65,503
36	65,599
40	65,495
44	65,491
48	65,487
52	65,483
56	65,479
60	65,475

How many minimum-sized fragments would be needed to carry the complete packet in each of these cases? We must simply divide the data packet size by eight. If the answer is a round number, then it represents the number of eight-byte fragments required. If the answer includes a remainder, then another fragment is required to carry the extra data, and the total number of fragments required is indicated in parentheses.

In the case of a 20-byte Internet Header Length, we have a worst-case scenario of 8190 fragments. In such a case, the Fragment Offset field would contain values from 1 (00000000000001) through 8190 (1111111111110). In order to require such extreme fragmentation, a link would need to have a maximum transmission unit of between 28 and 35. IP cannot run over links that have an MTU smaller than 68, so in cases where the header length is 20 bytes, there is room for 48 bytes of data, which just happens to be a multiple of eight. So, the practical upper end on fragmentation is $65,515/48$, or 1364 48-byte fragments, with 43 bytes left over, for the 1365th fragment. In the Fragment Offset field, this would be 10101010101, believe it or not.



If all the fragments do not arrive within a reasonable period of time, then those fragments that did make it through must be discarded; losing a fragment is the same as losing the entire packet. The length of each fragment is reflected in its Total Length field. Once the receiver has accumulated all the fragments, it may reconstruct the original packet's Total Length by measuring the length of the concatenated fragments (not including the lengths of their headers).

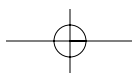
Important IP "Helper" Protocols

There are other protocols that are part of the IP stack, without which IP would be incomplete. The Internet Control Message Protocol (ICMP, RFC-792) is used for error-reporting and network diagnostic functions. ICMP is actually considered to be part of IP, in the sense that every IP module must support ICMP. The Internet Group Management Protocol (IGMP, RFC-2236) is used to support multicast IP,¹⁶ which is beyond the scope of this book. Most modern IP stacks support IGMP, but IP does not mandate the inclusion of IGMP.

Another absolutely critical protocol that IP could not live without is the Address Resolution Protocol (ARP). ARP is used in a LAN subnetwork to allow IP endstations to learn the subnetwork-layer addresses of their neighbor endstations on that LAN. Several WAN variants of ARP exist, including Inverse ARP and Asynchronous Transfer Mode ARP (ATM-ARP), but not all WAN subnetworks require, or support, address resolution.

WHAT IS ROUTING?

Once packets have been formed with the proper source and destination addresses, they are transmitted into the network; it is then up to the routers to forward them to the indicated destination. A router is a special purpose computer that is designed to "forward" packets. Each of the router's interfaces must be configured with a unique address, each having its own unique prefix¹⁷ (i.e., leading bits). The following chapters will cover the topic of IP addressing completely, but for now just remember that each interface on a router must have a unique IP address. Routers can have anywhere from two to more than 1,000 interfaces, each of which is connected to a LAN or WAN "subnetwork." In the case of a multiaccess subnetwork (there are both LAN and WAN technologies that can be classified as multiaccess), there may be a set of other routers attached to that subnetwork, each of which goes to different places.





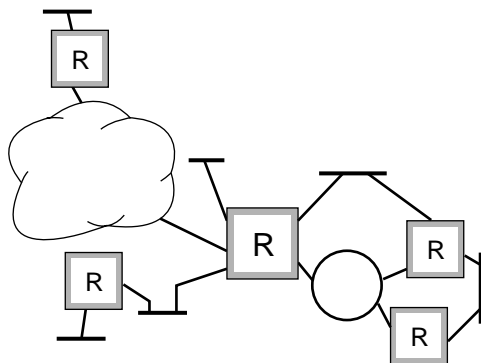
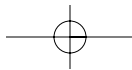
Nomenclature

The term “routing” is often used to refer to two completely different, but closely related, concepts. People use “routing” to refer to the process whereby routers exchange special “routing protocol” packets to describe their local topology to one another. “Routing” is also used to describe the process of deciding how to forward a packet, consisting of receiving a packet, looking up its destination in the forwarding table (commonly called the “routing table”), determining the next-hop router, and transmitting to the next-hop router (or perhaps the packet’s ultimate destination) on the appropriate outbound interface.

Usually it is clear from the context what a speaker means when they say routing.¹⁸ However, in this book the author will make every effort to say “forwarding” when describing the act of deciding where a packet needs to go and then delivering it on its way, versus “routing” when discussing the act of learning and sharing information about the topology. In short, routing is a process that facilitates packet forwarding.

Figure 1.12 shows a small subset of a topology, including subnetworks that have just one router, and one subnetwork that has three total routers. LAN subnetworks are often drawn as circles or lines, depending on the technology being represented. Symbols for WAN subnetworks are usually either clouds or simply lines.¹⁹ Endstations are almost always attached via the LAN subnetworks.

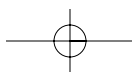
In order to learn the lay of the land, the routers communicate with each other using routing protocols. It is also possible for an administrator to manually tell their routers that “destination X is reachable via router P,” “destination Y is reachable via router Q,” and so on. This static method requires no information exchanges among the routers, but only works until the topology changes (e.g., a link or neighbor router fails or otherwise becomes unreachable). In this “static routing” case, a router has no way to detect such failures, and keeps forwarding traffic into oblivion, via a path that is not functional. Another problem with static routing is that it is difficult to configure more than a few routers with all possible destinations. As the network grows, this becomes more and more difficult to do, and thus ever more prone to error.

**FIGURE 1.12** Router topology.

The alternative, used in the vast majority of situations, is to let the routers use their own knowledge of their individual attachments to build up a routing table dynamically. These messages boil down to a router saying, “here’s what I am attached to,” or “here’s the places you can get to by going through me.” Remember that each router must be configured in advance to know what address(es) it has on each of its interfaces. Each router shares this connectivity information with its neighbors, and eventually they all “converge” on a stable understanding of all the reachable destinations, and which neighbor router is the best next-hop to reach each destination.

So, besides data traffic from endstations, which routers must forward as quickly as they can, there are also control messages, mostly routing protocol packets, which routers must listen to (according to their configuration). Most routers support a large suite of routing protocols. There are numerous standards-based routing protocols, and some well-known proprietary ones as well. At this point, it will suffice to say that a given set of routers must all be speaking the same “routing protocol” in order for them to know about the reachability of all destinations within that “routing domain.” Routing protocols define a set of messages that let routers advertise and receive information about reachable or unreachable destinations within the routing domain. A routing domain is essentially a collection of routers that speak the same protocol.

Later in the book we will look at ways to effectively use more than one routing protocol at a time. We will also examine ways to use the addressing structure of your network to give your routers strong hints about where certain destinations might be, thereby limiting the need for them to describe each and every little gory detail to each other, yet still maintaining enough reachability information to do the job of forwarding packets.



How is the Internet Built?

The Internet is a large collection of routers, organized as follows. At the highest level, there are a handful of large Internet Service Providers (ISPs), commonly called “Tier-1” providers. Tier-1 providers have networks that may span continents. These providers are attached to each other at convenient places called regional “peering points” or “exchanges,” as shown in Figure 1.13 as circles with Xs inside.

Tier-1 providers also typically attach directly to each other at “private peering” points, represented by the dashed lines in the figure. In fact, direct private peering is one of the distinguishing characteristics of a Tier-1 ISP; these ISPs have so much data to exchange that they need private peering points in order to function well. Another characteristic of Tier-1 providers is that they are present at all the major regional peering points. Lower-tier ISPs generally only “peer” with Tier-1 ISPs at the regional exchanges, where many dozens of ISPs of all levels peer with each other and deliver packets.

There are easily more than three Tier-1 ISPs; this figure is meant to illustrate the concept of peering at regional exchange points versus private peering. The end of each of the line in the figure is a router interface.

The large providers also attach to smaller providers, Tier-2, which are usually limited to a continent or a country. Tier-2 providers generally attach to Tier-1 ISPs at the regional peering points. If a Tier-2 ISP is large enough, it may peer with a Tier-1 ISP in more than one geographical location to help balance the traffic between the two, and also to provide a measure of resiliency. Tier-2 ISPs usually attach to multiple Tier-1 providers to provide their customers with richer connectivity.

Generally, Tier-2 peering rules are set up so that they do not inadvertently become a forwarding path between two Tier-1 providers. This is known as becoming a “transit” provider. Tier-1 providers are generally selling “transit services” to Tier-2 and below because the Tier-1 ISPs have high-capacity backbone circuits,

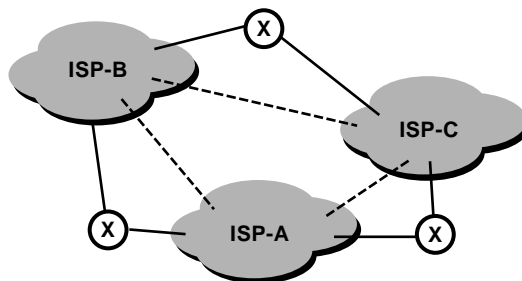
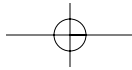


FIGURE 1.13 Tier-1 ISPs.



but the lower-tier providers do not have the capacity to carry data between two Tier-1 ISPs.

Below Tier-2, there are Tier-3 providers that purchase service from the Tier-2 providers, and so on. Each of these providers' networks, from Tier-1 on down, is based on IP routers, devices that understand the format of IP packets and know how to forward them toward their destinations. A large provider's network may have thousands of routers, while a small provider may have a dozen, more or less.

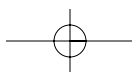
In total, there are hundreds of thousands of routers in the ISP networks, and besides the routers, there are computers that are used by the ISP's "operations" staff to configure, observe, and otherwise manage the network. Each ISP manages its own internal infrastructure, and the peering arrangements that they have with other ISPs represent how they make their money. If ISP A carries traffic for ISP B, then ISP B owes them money for that service. On the other hand, if ISP B carries traffic for ISP A, or its customers, then ISP A owes money to ISP B.

Ultimately, businesses or home-based users attach to some provider, at some level. There is no reason why Tier-1 providers couldn't have customers connecting directly to them, or have businesses attached [semi-]permanently to them. Customers may attach at any level of the hierarchy, wherever they can get a good deal, or otherwise obtain service that meets their specific requirements.

WHY IS THE INTERNET SO USEFUL?

The Internet is a general purpose infrastructure, in the same sense that the highway system is a general purpose infrastructure, as are the Postal service and the telephone network. Each of these are pervasive (in the geographic sense), and they can all carry a variety of payloads to suit their many types of users.

The mere existence of useful general purpose infrastructures actually seems to spur the creation of interesting applications for them. Of course, there had to be some applications or it would not have been built in the first place, but uses of such infrastructures evolve well beyond those envisioned by its creators. A good example is the revolution in commerce that was enabled by the arrival of overnight shipping companies, pioneered by Federal Express. Presumably, the reason it was built was that there was no easy way to get letters and packages delivered overnight. Once the capability existed, new businesses arose that had never been possible before. For example, a high-end florist business now basically cuts flowers to order and ships the floral arrangement overnight. Given the freshness of the flowers, they may last twice as long as those purchased from a local florist. This business would not be possible



without a reliable way to ship packages overnight. Because of the success of the pioneers, many other businesses joined the overnight shipping business, creating competition and a healthy marketplace. Many other examples exist of unforeseen applications of new infrastructures.

Examples of General Purpose Infrastructures

Transportation: Roads, Railroads, Airplanes, Shipping, etc.

Motorcycles, tanks, delivery vans, long-haul trucks, and cars all may use the interstate highway system and all its feeder roads. Some roads are wide, with many lanes, and others are narrow; some underpasses limit the size of trucks that can use certain roads. The highway system interconnects with train stations, airports, and shipping ports to facilitate the exchange of people and cargo among the different modes of transportation. Automobiles can be either a user of the transportation system (i.e., when cars are carried on ships or trucks), or a direct application of it (i.e., when people drive their kids to school, or go to a ball game).

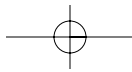
Postal Service

The postal service is but one application that makes use of the transportation system. The post carries parcels, postcards, letters, and provides extra services such as certified mail, for which you pay an additional fee. They pick up and deliver mail from all addresses once a day.

Telephone System

Using the telephone network, one can place a call to see how grandma is doing, or set up a job interview, or to order a pizza for delivery. One can also plug in a fax machine and call another fax machine to transfer a document. Other devices such as computers may use modems attached to the telephone network to send data to each other. The trick is that modems, fax machines, etc., have to speak the phone network's "language." Since the phone network was designed to carry human voices, it allows a relatively narrow audio frequency range (about 0 – 4 kHz) to pass through, enough so that voices are recognizable. Thus, any devices making use of the phone network must transmit tones that lie within this frequency range in order to have them pass through successfully.

All of these applications can be performed over local, long-distance, or international circuits, each of which has different rate structures that depend on the length



of the call, the geographic distance between the call's endpoints, the time of day, and whether the caller subscribes to a discount calling plan.

The fastest growing use of the telephone network over the last 15 years has been fax and other "data" (i.e., nonvoice) applications, such as two computers calling each other to transmit data on behalf of their users. The Internet is an ever-larger component of this data traffic explosion. Ironically, voice traffic is beginning to be carried over the Internet. Perhaps someday, all (or most) voice traffic will use the Internet.

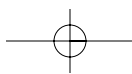
The Internet as a General Purpose Infrastructure

The Internet has much in common with all of these (and other) general purpose infrastructures. Just as you can send a letter to anyone, drive to almost anywhere, and call almost anyone, the Internet is becoming ubiquitous, asymptotically approaching universal planet-wide connectivity.

The packets which traverse the Internet may be large or small. They may represent any of a large, and growing, list of applications. IP packets cross links whose speed ranges across six orders of magnitude, each potentially congested by the presence of other traffic. Analogies can be made to any of the other "networks" above, but the Internet seems to have more in common with the highways or the postal service than the others. The highway system has some high-capacity roads, some two-lane roads, and some very congested roads (seemingly regardless of their size).²⁰

The way IP packets are delivered has much in common with the postal service. There are a lot of dedicated people that work at the postal service, but the quantity of mail they must process dictates that machines must be used to help deliver the mail in a timely fashion. Most of the time, a letter can cross the country in three to five days, but sometimes it takes longer, and sometimes the mail is lost or damaged en route to its destination. Due to unavoidable situations, it is impossible to get all the mail through, despite the best intentions of the post office. So, when you put a letter in the mail, you are not guaranteed how long it will take to get to its destination, nor are you guaranteed that it will even get to its destination. You know that it is very unlikely to be delayed for more than a few days, and you know that it may be destroyed in transit, but that this is also unlikely. The postal service makes their best effort to ensure that your mail will get through. IP networks also offer "best-effort" service.

Each packet that is transmitted into the network has its own "destination address" that tells the network where the packet needs to go. Each packet also has a return address (the packet's "source address"), so that the destination will know which



address the packet came from and will be able to send a reply (if necessary). Also, if a packet cannot get through, the network can use the return address to inform the source of the trouble, similar to the way the post office uses the return address on a letter.

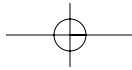
IP is also similar to the highway system. Just as a car can drive over a freshly-paved six-lane interstate highway, or a pothole-ridden city street, or across water on a ferry boat, IP packets can be sent over various different types of links, which are known as “subnetworks.” IP operates at what is known as the “network layer,” so it makes sense that links over which IP operates should be known as “sub-” networks. There are many kinds of subnetworks, the most common of which will be discussed later in the book.

Some IP packets are large and some are small. IP defines a maximum packet size, and some links over which IP operates have minimum packet sizes, or smaller maximum sizes. The fact that some subnetworks have different capacities than others is similar to roads which may have width- or weight-limited bridges, or height-limited underpasses.

INTERNET APPLICATIONS

The feature set of IP will be covered in more detail shortly. For now, it is worthwhile to focus on the subject of applications. There are numerous applications of IP. In the early days of the Internet, the most common applications were remote login and file transfers. Researchers at one university may have needed to access specific computing resources at another university, so they could remotely log in over the Internet rather than traveling there and touching a hard-wired terminal. Once the computers finished with their jobs, the data needed to be transferred back home, so file transfer was another logical application.

Electronic mail was another early application, and it arrived in two forms. First was the point-to-point variety, in which a user sends to only one other user, or a group of users (a “mailing list”). Another form was developed that evolved from a separate network called Usenet. In this case, messages are posted to topical “news-groups” that interested parties subscribe to. One person posts a question or observation and others respond, creating a “thread” of messages. This form of message exchange on Usenet, when it originated, was not based on IP, but the Unix-to-Unix CoPy protocol (uucp). Usenet became a worldwide network in its own right, for a long time much larger than the Internet, but as the Internet became larger and more



pervasive, it became possible to use the Internet to transfer Usenet messages. Today, most—but not all—of the old uucp-based Usenet has been replaced by the Network News Transfer Protocol, which operates over the TCP/IP-based Internet.

One of the most important applications of the Internet is its name-lookup system, known as the Domain Name Service (DNS).²¹ Virtually every IP-speaking device also supports the DNS application, which allows the devices to convert human-friendly names like “ftp.gnu.ai.mit.edu” into the equivalent IP address 18.159.0.42. This is a critical service, since IP packets must be addressed to IP addresses, but the Internet would be unusable for humans if we all had to remember IP addresses like 18.159.0.42 instead of human-friendly names like ftp.gnu.ai.mit.edu or ftp.ietf.org.

The World Wide Web (WWW) versus the Internet

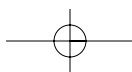
The Internet is the “packet hauling” infrastructure that has been in place, and growing rapidly, since the early 1980s. The WWW is an application that runs over the Internet. The term “web” is usually used interchangeably with the Internet these days, because so few people know that they are, in fact, different things. Electronic mail, remote terminal access, the domain name system, file transfer, network news, directory services, etc., are all applications of the Internet. They all predated the web, but the web was the “killer app” for the Internet—the application that made everyone want to be “on” the Internet so they could use the WWW applications.

REFERENCES

- Kleinrock, Leonard, “Information Flow in Large Communication Nets,” MIT, first paper on packet-switching (PS) theory, July 1961.
- Licklider, J.C.R., Clark, W. “On-Line Man Computer Communication,” MIT, Galactic Network concept encompassing distributed social interactions, August 1962.
- Baran, Paul, “On Distributed Communications Networks,” RAND, packet-switching networks; no single outage point.

Request for Comment (RFC)

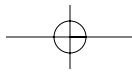
- 2430 A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE). T. Li, Y. Rekhter. October 1998. (Format: TXT=40148 bytes) (Status: INFORMATIONAL)



- 2474 Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. K. Nichols, S. Blake, F. Baker, D. Black. December 1998. (Format: TXT=50576 bytes) (Obsoletes RFC1455, RFC1349) (Updates RFC791, RFC1122, RFC1123, RFC1812) (Status: PROPOSED STANDARD)
- 2475 An Architecture for Differentiated Service. S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss. December 1998. (Format: TXT=94786 bytes) (Status: PROPOSED STANDARD)

ENDNOTES

1. The ITU-T is an international standards-setting body, whose acronym stands for International Telecommunications Union, Telecommunication Standardization Sector. The ITU-T was formerly known as the CCITT, or *Comité Consultatif International de Télégraphique et Téléphonique*. The ITU-T is a subset of the ITU, which is an agency of the United Nations.
2. The term gateway lives on as part of the term “default gateway,” which is a nearby router to which each IP endstation sends all its nonlocally-destined traffic.
3. The first four sites on the ARPANET were: UCLA, the Stanford Research Institute (SRI), UCSB, and the Univ. of Utah.
4. Transport-layer demultiplexing allows multiple applications to coexist on one machine, by virtue of different transport-layer labels called “ports.” Each layer in the protocol stack facilitates demultiplexing, or sharing, that layer’s services among multiple higher-layer entities.
5. Each layer in the protocol stack treats the layer above it as opaque; the transport layer is not unique in this characteristic.
6. Anyone familiar with this industry knows that tee shirts are the most influential force at trade shows. Clearly the impact of that tee shirt is still being felt!
7. A suite of “OSI protocols” were developed and implemented that correspond closely to the OSI-RM. Despite the Internet Protocol’s insurmountable head start, OSI protocols are in use, predominantly in Europe and in other non-US locations. Despite the OSI suite’s lack of commercial success (relative to the Internet Protocol suite), the OSI-RM still provides the context for classification of all layered protocols.
8. Some subnetwork layers have both a header and a trailer. The trailer may provide a data protection function, allowing the receiver to determine if the frame was received error-free or not. In cases where IP operates over such subnetworks, the presence of the trailer is understood, and the packet then includes all data up to, *but not including*, the subnetwork trailer.



Chapter 1 Introduction to the Internet Protocol

31

9. The 0x (that's a zero and a lower-case 'x') means that the following number is hexadecimal, or base-16. The 0x notation originated in the context of the C programming language, but it has achieved widespread use outside those circles.
10. This book was written on a Macintosh PowerBook 2400c/180 with 80 MB RAM, running Mac OS 8.1.
11. Before handing the packet up to the proper Network-layer protocol, the subnetwork-layer strips off the frame's header (and trailer, if present). The data which is handed up to the Network layer is just the Network-layer packet; no traces of the subnetwork layer remain.
12. Technically, ICMP is a part of the IP software module, but it is still encapsulated in an IP header.
13. After reading that paragraph, you may have the impression that routing protocols run directly over IP. Not quite; the Routing Information Protocol (RIP) is a client of UDP rather than using IP directly.
14. Curiously, version four was the first version of IP.
15. One might be tempted to think that the ultimate destination's router could do this re-assembly, but there is no reason why the ultimate destination of the fragments should be served by only one router. In such a case, fragments could be arriving from more than one "direction" (i.e., via more than one router). Again, the only place where the fragments are guaranteed to converge is the original packet's destination address.
16. Interested readers may wish to consult the Bibliography for references on multicast IP.
17. Think of the prefix as a telephone area code. A router interface consumes just one of the many available "phone numbers."
18. Some people pronounce this "rooting," a process performed by "rooters." Others (including the author) pronounce it "rowting," a process performed by "rowters." Either pronunciation is correct.
19. According to a telco "old-timer" friend of mine, a common "mistake" is to draw all point-to-point WAN links as a lightning bolt. The lightning-bolt icon was originally used to represent dial-up links. Permanent "nailed-up" point-to-point "leased line" links will be drawn as straight lines in this book.
20. Have you ever noticed that it doesn't seem to matter how many lanes there are—there is always congestion?
21. Recently, Microsoft has co-opted the acronym DNS to refer to an imaginary Digital Nervous System. Why not just call it the Internet? It is left as an exercise for the reader to consider what the motivation for such an action might be.

