1

Networking and Network Routing: An Introduction

Not all those who wander are lost.

J. R. R. Tolkien

D. Medhi and K. Ramasamy, Network Routing: Algorithms, Protocols, and Architectures

Morgan Kaufmann Publishers, ISBN13: 978-0-12-088588-6

It is often said that if anyone were to send a postcard with minimal address information such as "Mahatma Gandhi, India" or "Albert Einstein, USA," it would be routed to them due to their fame; no listing of the street address or the city name would be necessary. The postal system can do such routing to famous personalities usually on a case-by-case basis, relying on the name alone.

In an electronic communication network, a similar phenomenon is possible to reach *any* website or to contact *any* person by telephone anywhere in the world without knowing where the site or the person is currently located. Not only that, it is possible to do so very efficiently, within a matter of a few seconds.

How is this possible in a communication network, and how can it be done so quickly? At the heart of the answer to this question lies *network routing*. Network routing refers to the ability of an electronic communication network to send a unit of information from point A to point B by determining a path through the network, and by doing so efficiently and quickly. The determination of an efficient path depends on a number of factors, as we will be discussing in detail throughout this book.

First, we start with a key and necessary factor, known as *addressing*. In a communication network, addressing and how it is structured and used plays a critical role. In many ways, addressing in a communication network has similarities to postal addressing in the postal system. Thus, we will start with a brief discussion of the postal addressing system to provide an analogy.

A typical postal address that we write on a postcard has several components—the name of the person, followed by the street address with the house number ("house address"), followed by the city, the state name, and the postal code. If we, on the other hand, take the processing view to route the postcard to the right person, we essentially need to consider this address in the reverse order of listing, i.e., start with the postal code, then the city or the state name, then the house address, and finally the name of the person. You may notice that we can reduce this information somewhat; that is, you can just use the postal code and leave out the name of the city or the name of the state, since this is redundant information. This means that the information needed in a postal address consists of three main parts: the postal code, the street address (with the house number), and the name.

A basic routing problem in the postal network, then, is as follows: the postcard is first routed to the city or the geographical region where the postal code is located. Once the card reaches the postal code, the appropriate delivery post office for the address specified is identified and delivered to. Next, the postman or postwoman delivers the postcard at the address, without giving much consideration to the name listed on the card. Rather, once the card arrives at the destination address, the residents at this address take the responsibility of handing it to the person addressed.

You may note that at a very high-level view, the routing process in the postal system is broken down to three components: how to get the card to the specific postal code (and subsequently the post office), how the card is delivered to the destination address, and finally, how it is delivered to the actual person at the address. If we look at it in another way, the place where the postcard originated in fact does not need to know the detailed information of the street or the name to start with; the postal code is sufficient to determine to which geographical area or city to send the card. Thus, we can see that postal routing uses *address hierarchy* for routing decisions. An advantage of this approach is the decoupling of the rout-

ing decision to multiple levels such as the postal code at the top, then the street address, and so on. An important requirement of this hierarchical view is that there must be a way to divide the complete address into multiple distinguishable parts to help with the routing decision.

Now consider an electronic communication network; for example, a critical communication network of the modern age is the Internet. Naturally, the first question that arises is: how does addressing work for routing a unit of information from one point to another, and is there any relation to the postal addressing hierarchy that we have just discussed? Second, how is service delivery provided? In the next section, we address these questions.

1.1 Addressing and Internet Service: An Overview

In many ways, Internet addressing has similarities to the postal addressing system. The addressing in the Internet is referred to as *Internet Protocol (IP) addressing*. An IP address defines two parts: one part that is similar to the postal code and the other part that is similar to the house address; in Internet terminology, they are known as the *netid* and the *hostid*, to identify a network and a host address, respectively. Thus, a host is the end point of communication in the Internet and where a communication starts. A host is a generic term used for indicating many different entities; the most common ones are a web-server, an email server, and certainly the desktop, laptop, or any computer we use for accessing t he Internet. A netid identifies a contiguous block of addresses; more about IP Addressing later in Section 1.3.

Like any service delivery system, we also need a delivery model for the Internet. For example, in the postal system, one can request guaranteed delivery for an additional fee. The Internet's conceptual framework, known as *TCP/IP* (*Transmission Control Protocol/Internet Protocol*), relies on a delivery model in which TCP is in charge of the reliable delivery of information, while IP is in charge of routing, using the IP addressing mechanism. IP, however, does not worry about whether the information is reliably delivered to the address or is lost during transit. This is somewhat similar to saying that the postal system will route a postcard to the house address, while residents at this address (not the postal authority) are responsible for ensuring that the person named on the card receives it. While this may seem odd at first, this paradigm has been found to work well in practice, as the success of the Internet shows.

A key difference in the Internet as opposed to the postal system is that the sending host first sends a beacon to the destination address (host) to see if it is reachable, and waits for an acknowledgment *before* sending the actual message. Since the beacon also uses the same transmission mechanism, i.e., IP, it is possible that it may not reach the destination. In order to allow for this uncertainty to be factored in, another mechanism known as a *timer* is used. That is, the sending host sends the beacon, then waits for a certain amount of time to see if it receives any response. If it does not hear back, it tries to send the beacon a few more times, waiting for a certain amount of time before each attempt, until it stops trying after reaching the limit on the maximum number of attempts. The basic idea, then, requires that the receiving host should *also* know the address of the sender so that it can acknowledge the receipt of the beacon. As you can see, this means that when the sending host sends its beacon, it must also include its source IP address.

Once the connectivity is established through the beacon process, the actual transmission of the content transpires. This is where a good analogy is not available in the postal system;

rather, the road transportation network is a better fit to describe an analogy. If we imagine a group of 100 friends wanting to go to a game, then we can easily see that not all can fit in one car. If we consider that a car can hold five people, we will need twenty cars to transport this entire group. The Internet transfer model also operates in this fashion. Suppose that a document that we want to download from a host (web-server) is 2 MB. Actually, it cannot be accommodated entirely into a single fundamental unit of IP, known as *packet* or *datagram*, due to a limitation imposed by the underlying transmission system. This limitation is known as the *Maximum Transmission Unit* (MTU). MTU is similar to the limitation on how many people can fit into a single car. Thus, the document would need to be broken down into smaller units that fit into packets. Each packet is then labeled with both the destination and the source address, which is then routed through the Internet toward the destination. Since the IP delivery mechanism is assumed to be unreliable, any such packet can possibly get lost during transit, and thus would need to be retransmitted if the timer associated with this packet expires. Thus another important component is that content that has been broken down into smaller packets, once it arrives at the destination, needs to be reassembled in the proper order before delivering the document.

We conclude this section by pointing out that the acknowledgment and retransmission mechanism is used for most well-known applications on the Internet such as web or email. A slightly different model is used for applications that do not require reliable delivery; this will be discussed later in the chapter.

1.2 Network Routing: An Overview

In the previous section, we provided a broad overview of addressing and transfer mechanisms for data in Internet communication services. Briefly, we can see that eventually packets are to be routed from a source to a destination. Such packets may need to traverse many cross-points, similar to traffic intersections in a road transportation network. Cross-points in the Internet are known as *routers*. A router's functions are to read the destination address marked in an incoming IP packet, to consult its internal information to identify an outgoing link to which the packet is to be forwarded, and then to forward the packet. Similar to the number of lanes and the speed limit on a road, a network link that connects two routers is limited by how much data it can transfer per unit of time, commonly referred to as the *bandwidth* or *capacity* of a link; it is generally represented by a data rate, such as 1.54 megabits per second (Mbps). A network then carries *traffic* on its links and through its routers to the eventual destination; traffic in a network refers to packets generated by different applications, such as web or email.

Suppose that traffic suddenly increases, for example, because of many users trying to download from the same website; then, packets that are generated can possibly be queued at routers or even dropped. Since a router maintains a finite amount of space, known as a *buffer*, to temporarily store backlogged packets, it is possible to reach the buffer limit. Since the basic principle of TCP/IP allows the possibility of an IP packet not being delivered or being dropped enroute, the finite buffer at a router is not a problem. On the other hand, from an efficient delivery point of view, it is desirable not to have any packet loss (or at least, minimize it) during transit. This is because the reliable delivery notion works on the principle of retransmission and acknowledgment and any drop would mean an increase in delay due

to the need for retransmission. In addition, during transit, it is also possible that the content enclosed in a data packet is possibly corrupted due to, for example, an electrical signaling problem on a communication link. This then results in garbling of a packet. From an end-to-end communication point of view, a garbled packet is the same as a lost packet.

Thus, for efficient delivery of packets, there are several key factors to consider: (1) routers with a reasonable amount of buffer space, (2) links with adequate bandwidth, (3) actual transmission with minimal error (to minimize packets being garbled), and (4) the routers' efficiency in switching a packet to the appropriate outgoing link. We have already briefly discussed why the first two factors are important. The third factor, an important issue, is outside the scope of this book since encoding or development of an error-free transmission system is an enormous subject by itself; interested readers may consult books such as [666]. Thus, we next move to the fourth factor.

Why is the fourth factor important? A packet is to be routed based on the IP address of the destination host; however, much like street address information in a postal address, there are far too many possible hosts; it is impossible and impractical to store *all* host addresses at any router. For example, for a 32-bit address, theoretically a maximum of 2^{32} hosts are possible—a very large number (more about IP addressing in the next section). Rather, a router needs to consider a coarser level of address information, i.e., the netid associated with a host, so that an outgoing link can be identified quickly just by looking up the netid. Recall that a netid is very much like a postal code. There is, however, a key difference—netids do not have any geographical proximity association as with postal codes. For example, postal codes in the United States are five digits long and are known as ZIP (Zonal Improvement Plan) codes. Consider now Kansas City, Missouri, where a ZIP code starts with 64 such as 64101, 64102, and so on. Thus, a postcard can be routed to Kansas City, MO ("64") which in turn then can take care of routing to the specific ZIP code. This idea is not possible with IP addressing since netids do not have any geographical proximity. In fact, an IP netid address such 134.193.0.0 can be geographically far away from the immediately preceding IP netid address 134.192.0.0. Thus, at the netid level, IP addressing is flat; there is no hierarchy.

You might be wondering why IP address numbering is not geographic. To give a short answer, an advantage of a nongeographic address is that an organization that has been assigned an IP address block can keep its address block even if it moves to a different location or if it wants to use a different provider for connectivity to the Internet. A geographically based address system usually has limitations in regard to providing location-independent flexibility.

In order to provide the flexibility that two netids that appear close in terms of their actual numbering can be geographically far away, core routers in the Internet need to maintain an explicit list of all valid netids along with an identified outgoing link so that when a packet arrives the router knows which way to direct the packet. The list of valid netids is quite large, currently at 196,000 entries. Thus, to minimize switching time at a router, efficient mechanisms are needed that can look up an address, identify the appropriate outgoing link (direction), and process the packet quickly so that the processing delay can be as minimal as possible.

There is, however, another important phase that works in tandem with the lookup process at a router. This is the updating of a table in the router, known as the *routing table*, that contains the identifier for the next router, known as the *next hop*, for a given destination

netid. The routing table is in fact updated ahead of time. In order to update such a table, the router would need to store all netids it has learned about so far; second, if a link downstream is down or congested or a netid is not reachable for some reason, it needs to know so that an alternate path can be determined as soon as possible. This means that a mechanism is required for *communicating* congestion or a failure of a link or nonreachability of a netid. This mechanism is known as the *routing protocol* mechanism. The information learned through a routing protocol is used for generating the routing table ahead of time.

If new information is learned about the status of links or nodes, or the reachability of a netid through a routing protocol, a *routing algorithm* is then invoked at a router to determine the best possible next hop for each destination netid in order to update the routing table. For efficient packet processing, another table, known as the *forwarding table*, is derived from the routing table that identifies the outgoing link interfaces. The forwarding table is also known as the Forwarding Information Base (FIB). We will use the terms forwarding table and FIB interchangeably.

It should be noted that a routing algorithm may need to take into account one or more factors about a link, such as the delay incurred to traverse the link, or its available bandwidth, in order to determine the best possible path among a number of possible paths. If a link along a path does not have adequate bandwidth, congestion or delay might occur. To minimize delay, an important function, called *traffic engineering*, is performed. Traffic engineering is concerned with ways to improve the operational performance of a network and identifies procedures or controls to be put in place ahead of time to obtain good network performance.

Finally, there is another important term associated with networking in general and network routing in particular, labeled as *architecture*. There are two broad ways the term architecture from the architecture of a building is applicable here: (1) a floor inside a building may be organized so that it can be partitioned efficiently for creating office spaces of different sizes by putting in flexible partitions without having to tear down any concrete walls, (2) it provides standardized interfaces, such as electrical sockets, so that equipment that requires power can be easily connected using a standardized socket without requiring modification to the building or the floor or the equipment. Similarly, there are several ways we use the term *architecting a network*: for example, from the protocol point of view, various functions are divided so that each function can be done separately, and one function can depend on another through a well-defined relationship. From a router's perspective, architecting a network refers to how it is organized internally for a variety of functions, from routing protocol handling to packet processing. From a network perspective, this means how the network topology architecture should be organized, where routers are to be located and bandwidth of links determined for efficient traffic engineering, and so on. Later, we will elaborate more on architectures.

To summarize, we can say that the broad scope of network routing is to address routing algorithms, routing protocols, and architectures, with architectures encompassing several different aspects for efficient routing. In this book, we will delve into these aspects in depth. With the above overview, we now present IP addressing in detail.

1.3 IP Addressing

If one has to send data to any host in the Internet, there is a need to uniquely identify all the hosts in the Internet. Thus, there is a need for a global addressing scheme in which no two

hosts have the same address. Global uniqueness is the first property that should be provided in an addressing scheme.

1.3.1 Classful Addressing Scheme

An IP address assigned to a host is 32 bits long and should be unique. This addressing, known as IPv4 addressing, is written in the bit format, from left to right, where the left-most bit is considered the most significant bit. The hierarchy in IP addressing, similar to the postal code and the street address, is reflected through two parts, a network part and a host part referred as the pair *(netid, hostid)*. Thus, we can think of the Internet as the *interconnection* of networks identified through netids where each netid has a collection of hosts. The network part (netid) identifies the network to which the host is attached, and the host part (hostid) identifies a host on that network. The network part is also referred as the *IP prefix*. All hosts attached to the same network share the network part of their IP addresses but must have a unique host part.

To support different sizes for the (netid, hostid) part, a good rule on how to partition the total IP address space of 2^{32} addresses was needed, i.e., how many network addresses will be allowed and how many hosts each of them will support. Thus, the IP address space was originally divided into three different classes, Class A, Class B, and Class C, as shown in Figure 1.1 for networks and hosts. Each class was distinguished by the first few initial bits of a 32-bit address.

For readability, IP addresses are expressed as four decimal numbers, with a dot between them. This format is called the *dotted decimal notation*. The notation divides the 32-bit IP address into 4 groups of 8 bits and specifies the value of each group independently as a decimal number separated by dots. Because of 8-bit breakpoints, there can be at most 256 ($= 2^8$) decimal values in each part. Since 0 is an assignable value, no decimal values can be more than 255. Thus, an example of an IP address is 10.5.21.90 consisting of the four decimal values, separated by a dot or period.

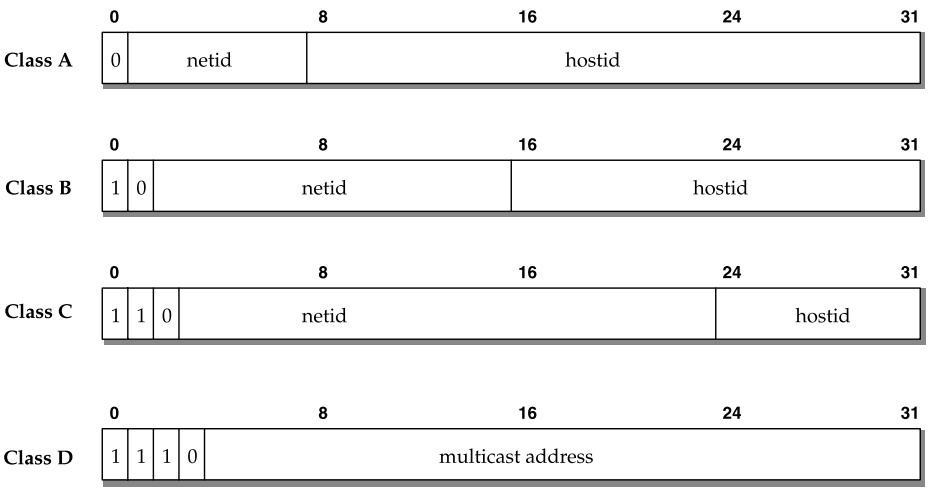


FIGURE 1.1 Classful IP addressing scheme.

Each Class A address has the first bit set to 0 and is followed by 7 bits for the network part, resulting in a maximum of 128 ($= 2^7$) networks; this is then followed by a 24-bit host part. Thus, Class A supports a maximum of $2^{24} - 2$ hosts per network. This calculation subtracts 2 because 0s and 1s in the host part of a Class A address may not be assigned to individual hosts; rather, all 0s that follows a netid such as 10.0.0.0 identify the network, while all 1s that follow a netid such as 10.255.255.255 are used as the broadcast address for this network. Each Class B network address has the first two bits set to "10," followed by a 14-bit network part, which is then followed by a 16-bit host part. A maximum of 2^{14} networks can be defined with up to $2^{16} - 2$ hosts per network. Finally, a Class C network address has the first three bits set as "110" and followed by a 21-bit network part, with the last 8 bits to identify the host part. Class C provides support for a maximum of $2^{21} (= 2{,}097{,}152)$ networks with up to 254 ($2^8 - 2$) hosts. In each class, a set of network addresses is reserved for a variety of purposes; see [319].

Three address classes discussed so far are used for unicasting in the Internet, that is, for a host-to-host communication. There is another class of IP addresses, known as Class D addressing, that is used for *multicasting* in the Internet; in this case, the first four bits of the 32-bit address are set to "1110" to indicate that it is a multicast address. A host can use a multicast address as the destination address for a packet generated to indicate that the packet is meant for any hosts on the Internet; in order for any hosts to avail this feature, they must use another mechanism to tune into this address. Multicast address on the Internet can be thought of as similar to a radio station frequency; a radio station transmits on a particular frequency—any listener who wants to listen to this radio station must tune the radio dial to this frequency.

The original rationale behind classes of different sizes was to provide the flexibility to support different sized networks, with each network containing a different number of hosts. Thus, the total address length can still be kept fixed at 32 bits, an advantage from the point of view of efficient address processing at a router or a host. As the popularity of the Internet grew, several disadvantages of the addressing scheme came to light. The major concerns were the rate at which the IP address blocks that identify netids were being exhausted, especially when it was necessary to start assigning Class C level netids. Recall from our earlier discussion that IP netids are nongeographic; thus, all valid netids are required to be listed at the core routers of the Internet along with the outgoing link, so that packets can be forwarded properly. If we now imagine all Class C level netids being assigned, then there are over 2 million entries that would need to be listed at a core router; no current routers can handle this number of entries without severely slowing packet processing. This issue, first recognized in the early 1990s, led to the development of the concept of *classless* addressing. In order to understand this concept, we first need to understand subnetting/netmask.

1.3.2 Subnetting/Netmask

Consider the IP address 192.168.40.3 that is part of Class C network 192.168.40.0. A subnet or sub-network is defined through a network mask boundary using the specified number of significant bits as 1s. Since Class C defines networks with a 24-bit boundary, we can then consider that the most significant 24 bits are 1s, and the lower 8 bits are 0s. This translates to the dotted decimal notation 255.255.255.0, which is also compactly written as "/24" to

indicate how many most significant bits are 1s. We can then do a bit-wise logical "AND" operation between the host address and the netmask to obtain the Class C network address as shown below:

```
      11000000 10101000 00101000 00000011   → 192.168.40.3
AND   11111111 11111111 11111111 00000000   → netmask (/24)
      11000000 10101000 00101000 00000000   → 192.168.40.0
```

As you can see, both the host address and the netmask have 1s in the first two positions from the left; thus, the "AND" operation results in 1s for these two positions. For the third position from left, the host has 0 while the netmask has 1; thus, the result of the "AND" operation is zero; and so on. Note that for network addresses such as Class C address, the netmask is implicit and it is on a /24 subnet boundary. Now consider that we want to change the netmask *explicitly* to /21 to identify a network larger than a 24-bit subnet boundary. If we now do the bit-wise operation

```
      11000000 10101000 00101000 00000011   → 192.168.40.3
AND   11111111 11111111 11111000 00000000   → netmask (/21)
      11000000 10101000 00101000 00000000   → 192.168.40.0
```

we note that the network address is again 192.168.40.0. However, in the latter case, the network boundary is 21 bits. Thus, to be able to clearly distinguish between the first and the second one, it is necessary to explicitly mention the netmask. This is commonly written for the second example as 192.168.40.0/21, where the first part is the netid and the second part is the mask boundary indicator. In this notation, we could write the original Class C address as 192.168.40.0/24 and thus, there is no ambiguity with 192.168.40.0/21.

1.3.3 Classless Interdomain Routing

Classless Interdomain Routing (CIDR) uses an explicit netmask with an IPv4 address block to identify a network, such as 192.168.40.0/21. An advantage of explicit masking is that an address block can be assigned at any bit boundaries, be it /15 or /20; most important, the assignment of Class C level addresses for networks that can show up in the global routing table can be avoided or minimized. For example, a contiguous address block can be assigned at the /21 boundary which can be thought of as an aggregation of subnets at the /24 boundary. Because of this, the term *supernetting* or *variable-length subnet masking* (VLSM) is also used in reference to the explicit announcement of the netmask.

Through such a process, and because of address block assignment at boundaries such as /21, the routing table growth at core routers can be delayed. In the above example, only the netid 192.168.40.0/21 needs to be listed in the routing table entry, instead of listing *eight* entries from 192.168.40.0/24 to 192.168.47.0/24. Thus, you can see how the routing table growth can be curtailed. CIDR was introduced around the mid-1990s; the current global routing table size, as of this writing, is about 196,000 entries. The routing table growth over time, along with projection, is shown later in Figure 9.10. In order for CIDR to take effect, any network address reachability announcement that is communicated with a routing protocol such as the *Border Gateway Protocol* must also carry the mask information explicitly. Its usage and applicability will be discussed in more detail in Chapter 8 and Chapter 9. In Table 1.1, we show a set of IP addresses reserved for a variety of purposes; see [319] for the complete list.

TABLE 1.1 Examples of reserved IP address blocks.

Address Block	Current Usage
0.0.0.0/8	Identifies source hosts in the current network
10.0.0.0/8	Private-use IP networks
127.0.0.0/8	Host loopback address
169.254.0.0/16	Link local for communication between links on a single link
172.16.0.0/12	Private-use IP networks
192.168.0.0/16	Private-use IP networks
240.0.0.0/4	Reserved for future use

1.4 On Architectures

Architectures cover many different aspects of networking environments. Network routing must account for each of the following architectural components. Some aspects of the architectures listed below are critical to routing issues:

- *Service Architecture:* A service model gives the basic framework for the type of services a network offers.
- *Protocol Stack Architecture:* A protocol stack architecture defines how service delivery may require different functions to be divided along well-defined boundaries so that responsibilities can be decoupled. It does not describe how actual resources might be used or needed.
- *Router Architecture:* A router is a specialized computer that is equipped with hardware/software for packet processing. It is also equipped for processing of routing protocols and can handle configuration requirements. A router is architected differently depending on its role in a network, such as a core router or an edge router, although all routers have a common set of requirements.
- *Network Topology Architecture:* For efficient operation as well as to provide acceptable service to its users, a network is required to be organized based on a network topology architecture that is scalable and allows growth. In order to address efficient services, there is also a direct connection among the topology architecture, traffic engineering, and routing.
- *Network Management Architecture:* A network needs to provide several additional functions in addition to carrying the user traffic from point A to point B; for clarity, the user data traffic forwarding is considered as the *data plane*. For example, from an operational point of view, a *management plane* handles the configuration responsibility of a network, and a *control plane* addresses routing information exchanges.

In the following sections, we elaborate on the above architectural facets of networking. To simplify matters, most of the following discussions will center around IP networks. Keep in mind that these architectures are applicable to most communication networking environments as well.

1.5 Service Architecture

An important aspect of a networking architecture is its service architecture. The service architecture depends partly also on the communication paradigm of its information units. Every networking environment has a service architecture, much like the postal delivery system. In the following, we focus on discussing three service models associated with IP networks.

Best-Effort Service Architecture

Consider an IP network. The basic information unit of an IP network is a packet or a datagram which is forwarded from one router to another towards the destination. To do that, the IP network uses a switching concept, referred to as *packet switching*. This means that a router makes decisions by identifying an outgoing link on a packet-by-packet basis instantaneously after the packet arrives. At the conceptual level, it is assumed that no two packets are related, even though they might arrive one after another and possibly for the same web-page downloaded. Also, recall that at the IP level, the packet forwarding function is provided without worrying about reliable delivery; in a sense, IP makes its best effort to deliver packets. Because of this, the IP service paradigm is referred to as the *best-effort service*.

Integrated Services Architecture

Initially, the best-effort service model was developed for the reliable delivery of data services, since it was envisioned that services would be data-oriented services that can tolerate delay, but not loss of packets. This model worked because the data rate provided during a session can be adaptive.

The concept for integrated services ("int-serv") architecture was developed in the early 1990s to allow functionalities for services that are real-time, interactive, and that can tolerate some loss, but require a bound on the delay. Furthermore, each session or connection requires a well-defined bandwidth guarantee and a dedicated path. For example, interactive voice and multimedia applications fall into this category. Note that the basic best-effort IP framework works on the notion of statelessness; that is, two consecutive packets that belong to the same connection are to be treated independently by a router. Yet, for services in the integrated services architecture that require a connection or a session for a certain duration of time, it became necessary to provide a mechanism to indicate the longevity of the session, and the ability for routers to know that resources are to be reserved for the entire duration.

Since the basic IP architecture works on the notion of statelessness, and it was infeasible to completely change the basic IP service architecture, a soft-state concept was introduced to handle integrated-services. To do that, a session setup and maintenance protocol was also developed that can be used by each service—this protocol is known as the resource ReSerVation Protocol (RSVP). The basic idea was that once a session is established, RSVP messages are periodically generated to indicate that the session is alive. The idea of integrated services was a novel concept that relies on the soft-state approach. A basic problem is the scalability of handling the number of RSVP messages generated for all sessions that might be simultaneously active at a router or a link.

DIFFERENTIATED SERVICES ARCHITECTURE

The differentiated services ("diff-serv") architecture was developed to provide prioritized service mechanisms without requiring connection-level information to be maintained at routers. Specifically, this approach gives priority to services by marking IP packets with diff-serv code points located in the IP header. Routers along the way then check the diff-serv code point and prioritize packet processing and forwarding for different classes of services. Second, this model does not require the soft-state concept and thus avoids the connection-level scalability issue faced with RSVP. Diff-serv code points are identified through a 6-bit field in the IPv4 packet header; in the IPv6 packet header, the traffic class field is used for the same purpose.

SUPPLEMENTING A SERVICE ARCHITECTURE

Earlier in this section, we introduced the best-effort service model. In a realistic sense, and to provide acceptable quality of service performance, the basic concept can be supplemented with additional mechanisms to provide an acceptable service architecture, while functionally it may still remain as the best-effort service architecture. For example, although the basic conceptual framework does not require it, a router can be designed to do efficient packet processing for packets that belong to the same web-page requested by a user since they are going to the same destination. That is, a sequence of packets that belongs to the same pair of origination and destination IP addresses, to the same pair of source and destination port numbers, and to the same transport protocol (either TCP or UDP) can be thought of as a single entity and is identified as a *microflow*. Thus, packets belonging to a particular microflow can be treated in the same manner by a router once a decision on forwarding is determined based on the first packet for this microflow.

Another way to fine-tune the best-effort service architecture is through traffic engineering. That is, a network must have enough bandwidth so that delay or backlog can be minimal, routers must have adequate buffer space, and so on, so that traffic moves efficiently through the network. In fact, both packet processing at a router and traffic engineering work in tandem for providing efficient services.

Similarly, for both integrated-services and differentiated-service architecture, packet handling can be optimized at a router. Furthermore, traffic engineering can be geared for integrated services or differentiated services architectures.

1.6 Protocol Stack Architecture

Another important facet of a networking environment is the protocol stack architecture. We start with the OSI (Open Systems Interconnections) reference model and then discuss the IP protocol stack architecture and its relation to the OSI reference model.

1.6.1 OSI Reference Model

The OSI reference model was developed in the 1980s to present a general reference model for how a computer network architecture should be functionally divided. As part of OSI, many protocols have also been developed. Here, we will present the basic reference model.

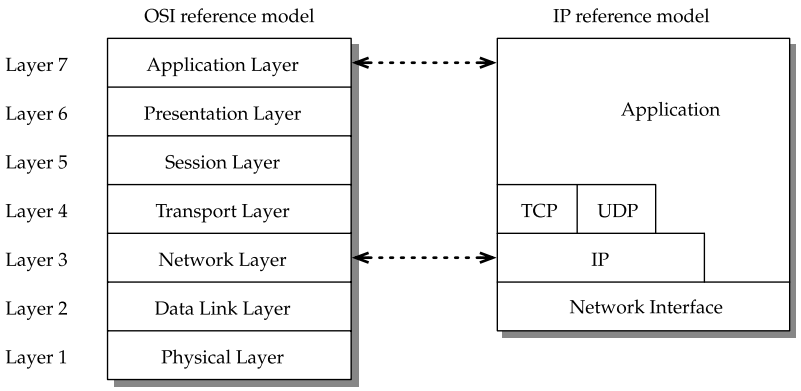


FIGURE 1.2 The OSI reference model and the IP reference model.

The OSI reference model uses a layered hierarchy to separate functions, where the layering is strictly enforced. That is to say that an N-layer uses services provided by layer $N-1$; it cannot receive services directly from layer $N-2$. In the OSI model, a seven-layer architecture is defined; this is shown in Figure 1.2. The seven layers are also referenced by layer numbering counting from bottom up. From a functional point of view, layer 1 provides physical layer functions. Layer 2 provides the data link function between two directly connected entities. Layer 3 is the network layer, where addressing and routing occurs. Layer 4 is the transport layer that can provide either reliable or unreliable transport services, with or without defining a connection ("connection-oriented" or "connection-less"). Layer 5 is the session layer, addressing communication that may transcend multiple connections. Layer 6 is the presentation layer that addresses structured information and data representation. Layer 7 is where the application layer protocols are defined.

While not every computer networking environment strictly adheres to the OSI reference model, it does provide an easy and simple way to check and compare what a particular networking environment might have to consider. Thus, this reference model is often quoted; in fact, you will hear use of terms such as "layer 2" device or "layer 3" device in the technical community quite often, assuming you know what they mean.

1.6.2 IP Protocol Stack Architecture

The IP architectural model can be classified into the following layers: the network interface, the IP layer, the transport layer, and the application layer (see Figure 1.2). We can easily see that it does not exactly map into the seven-layer OSI reference model. Actual applications are considered on the top of the application layer, although the IP model does not strictly follow layering boundaries as in the OSI reference model. For example, it allows an application to be built without using a transport layer; *ping* is such an example. We have discussed earlier that IP includes both the destination and the source address—this is accomplished through a header part in the IP packet that also contains additional information. The IP model does not explicitly declare how the layer below the IP layer needs to be; this part is simply referred to as the network interface that can support IP and will be discussed later in the chapter.

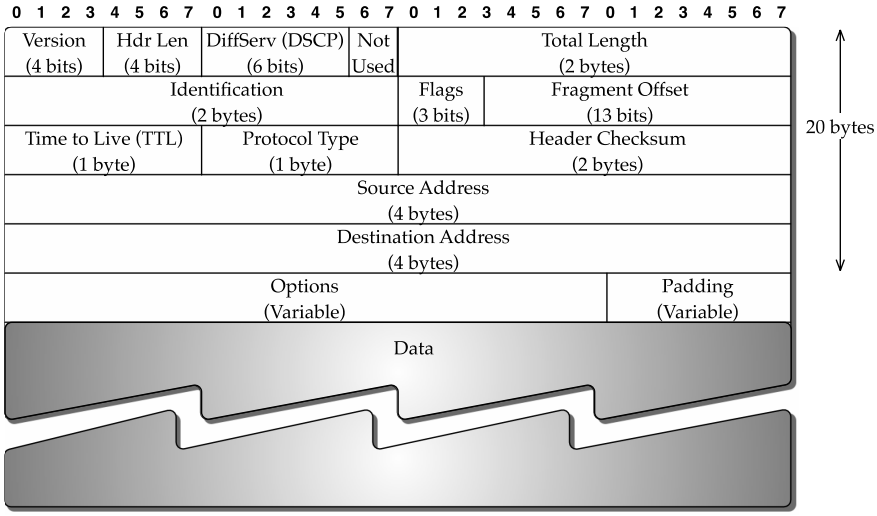
Network and Transport Layer

The IP addressing is defined at the IP layer, where the delivery mode is assumed to be unreliable. The transport layer that is above the IP layer provides transport services, which can be either reliable or unreliable. More important, the transport layer provides another form of addressing, commonly known as the *port number*. Port numbers are 16 bits long. Thus, the unreliable transport layer protocol, known as the User Datagram Protocol (UDP), can be thought of as allowing the extension of the address space by tagging a 16-bit port number to the 32-bit IP address. However, the role of the port number is solely at the host while routing is still done using the IP address. This is similar to the decoupling of the postal code and the house address in the postal addressing system. The reliable transport counterpart of UDP is known as the Transmission Control Protocol (TCP) which also uses a 16-bit port number, but provides reliable transport layer service by using a retransmission and acknowledgment mechanism. To be able to include the port number and other information, both TCP and UDP have well-defined headers. Because of two-way communication, similar to an IP packet including both the source and the destination address, TCP and UDP also include port numbers both for the source and the destination side. Since both TCP and UDP are above IP, a field in the IP header, known as the protocol type field, is used to be able to distinguish them. That is, through five pieces of information consisting of the source and the destination IP addresses, the source and the destination port numbers, and the transport protocol type, a connection in the Internet can be uniquely defined. This is also known as a microflow.

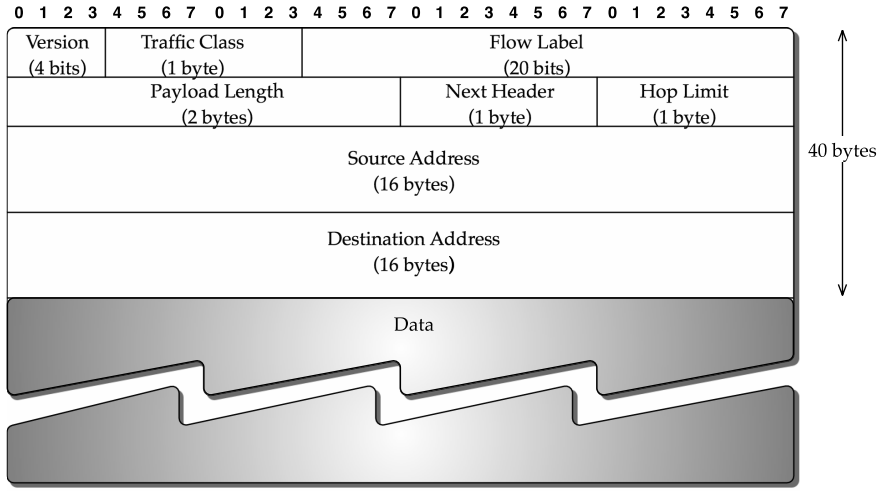
There are two IP packet formats: IPv4 and IPv6 (see Figure 1.3). IPv4 uses the 32-bit IP address and is the most widely deployed addressing scheme. IPv6 uses a longer 128-bit address that was developed in the mid-1990s; initially, it was designed anticipating that IPv4 addresses would be running out soon. This did not happen as initially thought, partly because of the proliferation of private IP address usage (see Table 1.1) that has been made possible by mechanisms known as network address translation (NAT) devices, which can map and track multiple private IP addresses to a single IP address. Packet formats for TCP and UDP are shown in Figure 1.4. So far, we have already discussed several well-known fields in these packets, such as IP source and destination addresses, source and destination port numbers, the protocol type field, and the diff-serv code point; other key fields shown in packets formats will be discussed later in Appendix B.14.

Application Layer and Applications

Information structure at the transport layer is still at the byte level; there is no structured, semantic information considered at this level. However, structural information is needed for a particular application. For example, an email requires fields such as "From," "To" before the body of a message is added; this then helps the receiving end know how to process the structured information. In order to provide the structured information for different applications, the IP architectural model allows the ability to define application layer protocols on the top of the transport layer protocols. Application layer protocols use unique listening port numbers from the transport layer level to distinguish one application from another. In other words, the IP architectural model cleverly uses the transport layer port number to streamline different application layer protocols, instead of defining yet another set of addresses at the application layer protocol level. Examples of application layer protocols are Simple Mail Transfer Protocol (SMTP), and HyperText Transport Protocol (HTTP), which are used by email and web



(a) IPv4 packet

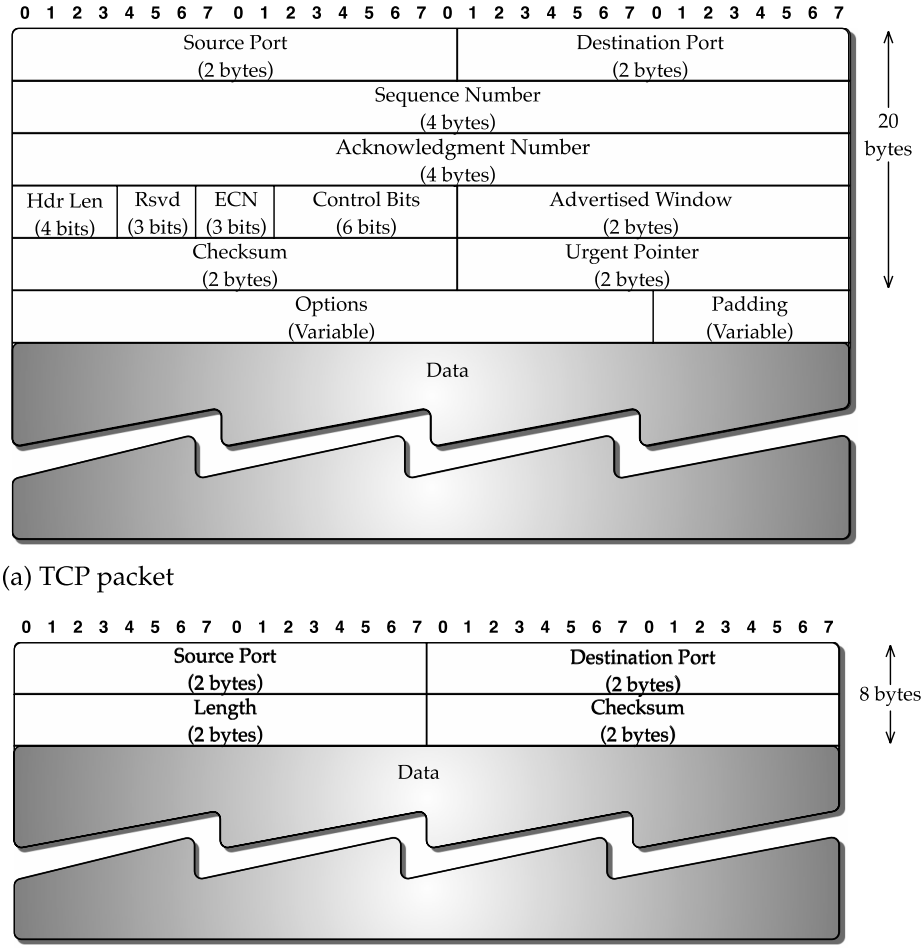


(b) IPv6 packet

FIGURE 1.3 Packet formats: IPv4 and IPv6.

applications, respectively. However, the terminology is a bit confusing with some of the older application layer protocols, since both the application layer protocol and its associated application are described by the same name; for example, File Transfer Protocol (FTP), and telnet. It may be noted that this set of application layer protocols (SMTP, HTTP, FTP, telnet) requires reliable data delivery and, thus, uses TCP as the transport layer protocol.

There are other applications that do not require reliable data delivery. Voice over IP protocol, commonly referred to as VoIP, is one such application that can tolerate some packet loss and thus, retransmission of lost packets is not necessary. Such an application can then use



(a) TCP packet

(b) UDP packet

FIGURE 1.4 Packet formats: TCP and UDP.

UDP. Since UDP does not provide any structural boundaries, and because many real-time communications, such as voice and video, require similar structural formats with the ability to distinguish different encoding mechanisms, Real-time Transport Protocol (RTP) has been defined above UDP. For example, a voice stream, with its coding based on G.711 PCM coding standards, can use RTP, while a motion JPEG video can also use RTP; they are distinguished through a payload-type field in RTP.

Role of Headers

By now, it might have become apparent that each layer needs to add a *header* to provide its functionality; and it then encapsulates the content received from the layer above. For example, RTP adds a header so that the payload time, among other things, can be indicated. How is then a message or a web page generated at an application level related to the layered data units, along with a header? To see this, consider transferring a web page. First, the

HTTP protocol adds its header to the entire content of the page. Suppose that the combination of this header and the content of the page comes to 50 kbytes. This content is then broken into smaller units. If such a unit is to be of 1000 bytes each, for example, due to a limitation imposed by the maximum transmission unit of a link, then we have to create 50 units of information. First, TCP will include its header which is 20 bytes in the default case, to make each unit, commonly referred to as a *segment*, to be 1020 bytes. Then, IP will include its own header, which is 20 bytes in the default mode. Thus, altogether each unit becomes a packet of size 1040 bytes at the IP level.

Where Do Routing Protocols Fit In?

We next discuss the exchange of information required for routing protocols. It is important to note that such exchanges of information for routing protocols also use the same protocol architectural framework. The information content of a routing protocol exchange has specific semantic meaning so that two routers can exchange and understand this information using these semantics. Interestingly, a router in the Internet is a also a host and is assigned an IP address. Thus, any communication between two adjacent routers is similar to any communication between any two hosts. Since IP is the core network layer, this means that IP is also used for this information exchange, much like using IP for communications related to the web or email. This is where the protocol-type field in the IP header, and the port numbering at the transport layer, can be used for distinguishing information exchanges related to different routing protocols. Three well-known routing protocols that we will be discussing later in the book are: Routing Information Protocol (RIP), Open Shortest Path First protocol (OSPF), and Border Gateway Protocol (BGP). Each of these protocols uses a different approach and exchanges different types of information. RIP is a protocol defined on top of UDP through a well-known listening port number and the unreliable delivery provided by UDP is used. Although not a transport layer protocol, OSPF is defined directly on top of IP by being assigned a protocol-type field at the IP level. It has its own retransmission and acknowledgment mechanism since it requires reliable delivery mechanisms. BGP is defined on top of TCP through a well-known listening port number, and BGP relies on TCP's reliable service to transfer its structured contents. An application, usually a command-line interface, is available with each router so that specific commands can be issued for each of these routing protocols, which are then translated into respective routing protocol exchange messages for communication with its adjacent routers.

Auxiliary Applications

Besides applications for actual user data traffic and applications for providing routing information exchanges, the IP architecture also supports auxiliary applications needed for a variety of functions. A well-known application is the name-to-address translation function provided through the Domain Name System (DNS), such that a domain name like www.NetworkRouting.net can be mapped into a valid IP address. This function can be either invoked indirectly when a user accesses a website or can be invoked directly by using the command, *nslookup*. DNS is an application layer protocol that typically uses UDP for the transport layer function, but it can use TCP if needed. This example also shows that it is possible to define end applications that may depend on more than one transport layer protocol.

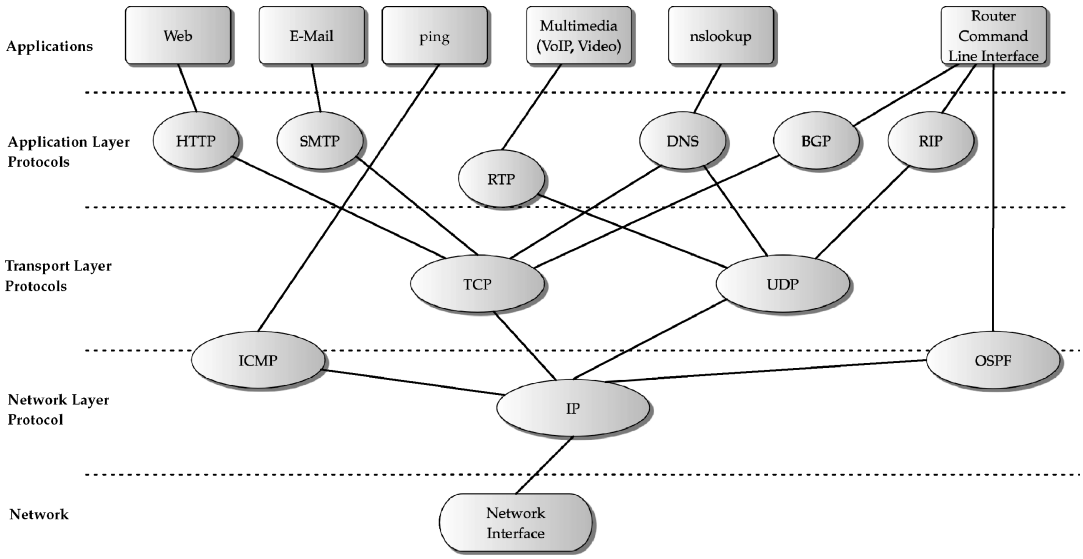


FIGURE 1.5 Protocol layering in IP architecture.

Another well-known utility application is *ping*, which is written on top of Internet Control Message Protocol (ICMP), that is directly over IP.

In Figure 1.5, we summarize the protocol dependency of different applications in terms of the application, transport, and network layer in the IP architecture.

1.7 Router Architecture

A router provides several important functions in order to ensure proper packet forwarding, and to do so in an efficient manner. A router is a specialized computer that handles three primary functions:

- *Packet Forwarding:* On receiving an incoming packet, a router checks whether the packet is error free. After inspecting the header of a packet for destination address, it performs a table lookup function to determine how to find the appropriate outgoing link.
- *Routing Protocol Message Processing:* A router also needs to handle routing protocol packets and determine if any changes are needed in the routing table by invoking a routing algorithm, when and if needed.
- *Specialized Services:* In addition, a router is required to handle specialized services that can aid in monitoring and managing a network.

A high-level functional view of a router is shown in Figure 1.6; it also shows how the routing table and the forwarding table fit in the overall process. In Part IV of this book, we will examine in detail router architectures, address lookup, packet processing, and so on.

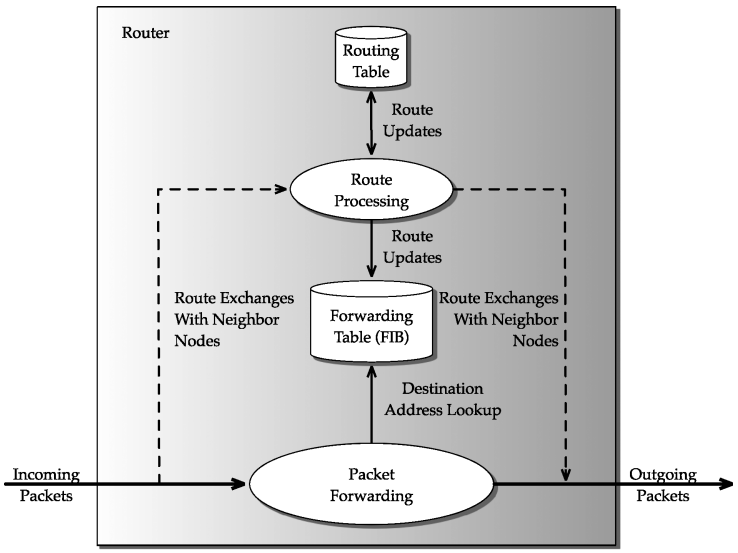


FIGURE 1.6 Router architecture: a functional view.

1.8 Network Topology Architecture

The network topology architecture encompasses how a network is to be architected in an operational environment while accounting for future growth. What does topology mean? It refers to the form a network will adopt, such as a star, ring, manhattan-street network, or a fully mesh topology, or a combination of them. The topological architecture then covers architecting a network topology that factors in economic issues, different technological capabilities, and limitations of devices to carry a certain volume of expected traffic and types of traffic, for an operational environment. Certainly, a network topology architecture also needs to take into account routing capability, including any limitation or flexibility provided by a routing protocol. It is up to a network provider, also referred to as a network operator or a service provider, to determine the best topological architecture for the network.

It is important to note that the operational experience of an existing network can contribute to the identification of additional features required from a routing protocol, or the development of a new routing protocol, or the development of a new routing algorithm or modification of an existing algorithm. We briefly discuss two examples: (1) when it was recognized in the late 1980s that the Internet needed to move from being under one network administrative domain to more flexible loosely connected networks managed by different administrative domains, BGP was developed, (2) when it was felt in the late 1970s that the telephone network needed to move away from a hierarchical architecture that provided limited routing capability to a more efficient network, dynamic call routing was developed and deployed. This also required changes in the topological architecture.

It may be noted that the term *network architecture* is also fairly commonly used in place of network topology architecture. One difficulty with the term *network architecture* is that it is also used to refer to a protocol architecture. It is not hard to guess that network providers are the ones who usually use the term network architecture to refer to a topological architecture.

1.9 Network Management Architecture

From the discussion in the previous sections, we can see that the routing information exchange uses the same framework as the user data traffic in the Internet. For an operational network, it is important to have a network management architecture where various functions can be divided into "planes." Specifically, we consider three different planes: the management plane, the control plane, and the data plane.

The management plane addresses router configuration and collection of various statistics, such as packet throughput, on a link. Router configuration refers to configuration of a router in a network by assigning an IP address, identifying links to its adjacent routers, invoking one or more routing protocols for operational usage, and so on. Statistics collection may be done, for example, through a protocol known as Simple Network Management Protocol (SNMP). The management plane of a router is closely associated with network operations.

The control plane exchanges control information between routers for management of a variety of functions, such as setting up a virtual link. The control plane is also involved in identifying the path to be taken between the endpoints of this virtual link, which relies on the routing information exchange.

Another clarification is important to point out. Since these functions are different, the routing-related functions are in the *control plane*, and the data transfers, such as the web or email, are in the *data plane*. These two planes, as well as the management plane, use IP for communication, so at the IP layer, there is no distinction between these functional planes. As we go through additional networking environments in this book, you will find that there are environments in which the control plane and the management plane are completely partitioned from the data plane.

It may be noted that for efficient traffic engineering of a network, certain information is also required from different routers. Such information exchanges can be conducted either through the control plane or through the management plane. In certain networking environments, some functions can overlap across different planes. Thus, the three planes can be thought of as interdependent. A schematic view is presented in Figure 1.7.

1.10 Public Switched Telephone Network

So far, our discussions have been primarily related to the Internet. In this section, we present a brief overview of Public Switched Telephone Network (PSTN), another important communication network.

An information unit in the PSTN is a call. Many of the architectural aspects discussed so far apply to the PSTN as well. The PSTN has a global addressing scheme to uniquely identify an end device; an end device is commonly referred to as a telephone, while a more generic term is customer premise equipment (CPE). The global addressing scheme is known as E.164 addressing. It is a hierarchical addressing scheme that identifies the country code at the top level followed by the city or area code, and finally the number assigned to a subscriber. Nodes in the PSTN are called *switches*, which are connected by intermachine trunks (IMTs), also known as *trunkgroups*.

From a protocol architecture point of view, and using the OSI reference model, PSTN can be simply summed up as consisting of application layer, network layer, and physical layer. The application layer enables the telephone service, the network layer handles addressing

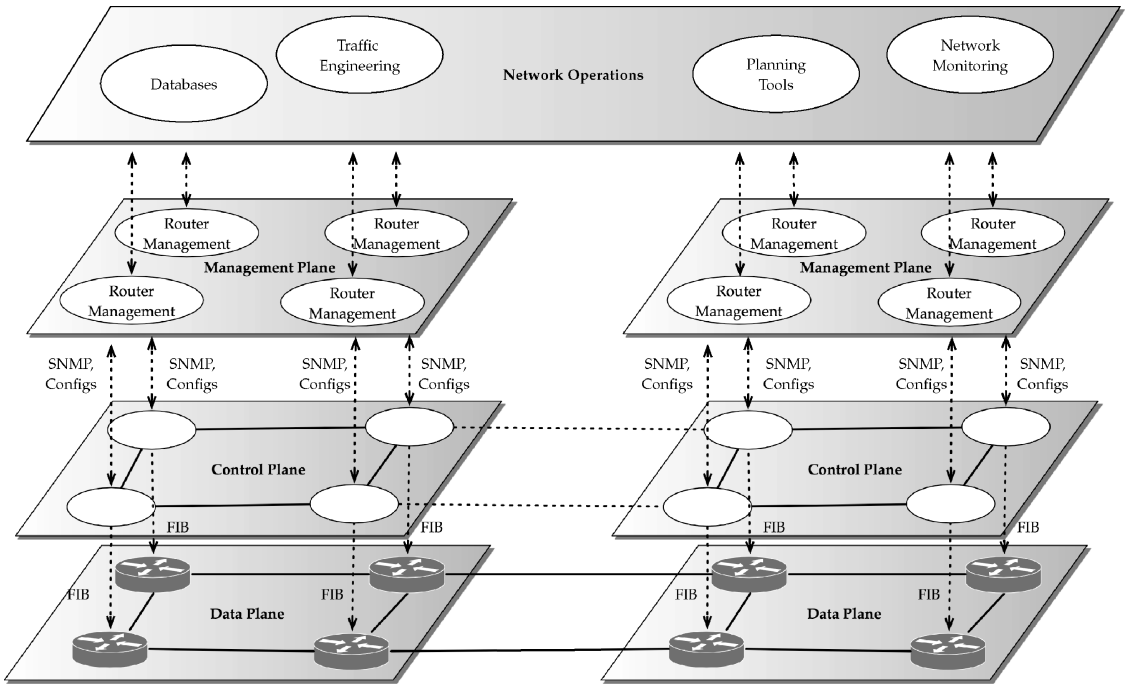


FIGURE 1.7 Network management architecture: data plane, control plane, and management plane.

and routing, while the physical transmission system carries the actual signal for voice communication. From a service architecture perspective, it provides the service model of *blocked-calls-cleared* mode using circuit switching. Circuit switching means that for a call requesting to be connected, a dedicated path is to be established instantaneously on demand from the source to the destination. The dedicated path is in fact a dedicated circuit with a specific bandwidth allocated—this value is 4 kilohertz (kHz) in an analog circuit and 64 kbps in a wireline digital circuit. The bandwidth of the circuit cannot be used by any other calls as long as this call is actively using it. Blocked-calls-cleared mode means that if the sequence of trunkgroups on all possible paths attempted from the source to destination does not have a circuit available for this call, then the call request is blocked and cleared from the system (not queued). Typically, a blocked call is indicated through a fast busy tone. Certainly, a user may retry.

More detail about routing in PSTN and its evolution will be covered later in Part III of this book. Routing in the IP-PSTN interworking environment will be presented in Chapter 20.

1.11 Communication Technologies

Communication technologies are used for carrying network layer services, whether for the Internet or PSTN. In this sense, communication technologies provide *transport* services for both the Internet and PSTN. Note that the use of the term *transport services* is not to be confused with the term *transport layer* of the OSI reference model. Unfortunately, the term transport is used in several ways in networking; these are two such examples. To provide transport

TABLE 1.2 Modular data rates.

Signal/data rate name	Bit rate (Mbps)
DS0 (voice circuit)	0.064
T1 (DS-1)	1.54
E1	2.04
Ethernet	10.00
T3 (DS-3)	45.00
E3	34.36
STS-1	51.84
Fast Ethernet	100.00
OC-3/STS-3/STM-1	155.52
OC-12/STS-12/STM-4	622.08
Gigabit Ethernet	1,000.00
OC-48/STS-48/STM-16	2,488.32
OTU1 (Optical Transport Unit-1)	2,666.06
OC-192/STS-192/STM-64	9,953.28
OTU2 (Optical Transport Unit-2)	10,709.22
OC-768/STS-768/STM-256	39,813.12
OTU3 (Optical Transport Unit-3)	43,018.41

services, transport networks are deployed that may be based on one or more communication technologies. At the real physical (duct) level though, fibers or coaxial cables are used for wired transport services. Such cables are either buried underground or carried overground on poles; submarine cabling is used for connecting different continents. Nowadays, submarine cables are almost exclusively based on fiber cables; for a recent map of global submarine cabling, see [693].

On top of cabling, a set of digital communication technologies can be provided; for example, SONET, T1/E1, T3/E3, and so on with well-defined data rates. A summary of different technologies and data rates is listed in Table 1.2, with all data rates listed using Mbps. A network is formed at any technological level, for example, SONET can use different rates such as OC-3 or OC-12. Similarly, a network can be formed at the T1 level or the T3 level. In particular, data rate multiplexing is also possible to go from one rate to another, such as from T1 to T3. The telecommunication infrastructure uses a mix of technologies, and transport services are provided either through networks at different levels, such as a network of T1s, a network of T3s, a network of SONET rings, or a combination of them. Each such transport network also needs to handle routing. For example, if a customer wants a T1 dedicated permanent circuit from Los Angeles to New York, the routing path needs to be mapped out. Certainly, the customer who wants the T1 transport service does not care how the T1 is routed in the transport network. However, for the T1 provider, it is an important problem since for all its T1 customers it needs to find efficient routing between different places.

In reference to the OSI terminology, the communication technologies reside mostly at layer 1 and sometimes in layer 2. Thus, instead of thinking about routing "purely" at the network layer (layer 3), routing problems also arise below layer 3 for transport network

providers. In recent years, virtual private networking has become immensely popular. It requires another form of routing that is above layer 2, but below layer 3, often dubbed as layer 2.5. For example, MultiProtocol Label Switching (MPLS) and Asynchronous Transfer Mode (ATM) fall into this category.

Essentially, to provide transport services using communication technologies, a variety of transport network routing problems arises that need to take into account the capability of a particular communication technology and the "routing" device. Second, multilayered networking and multilayered routing can also be envisioned going from layer 3 down to layer 1 due to transport network routing. Third, new technologies for transport networking are being continually developed with new capabilities, creating new opportunities in transport network routing. Finally, traditionally, different transport networks had very little capability to communicate with each other and thus relied on manual configurations. We are now starting to see development of new capabilities that allow dynamic configuration and the ability to exchange information between networks at different layers so that dynamically reconfigurable multilayer routing will be possible in the coming years. However, such multilayer routing brings new challenges. In Part V and Part VI of this book, we will cover transport network routing and multilayered routing, and the evolution of next-generation routing.

1.12 Standards Committees

It is important to note that for all technologies developed, standards play important roles. In fact, standards have been defined from a specific technology, such as T1, to packet formats, such as an IP packet. Standardization allows different vendors to develop products that can talk to each other so that customers can choose products from multiple vendors; this helps bring the price down. Furthermore, vendors look for innovative ways to implement specific standards to reduce their costs and be competitive with other vendors, who are offering similar products.

There are two types of standards: *de jure* and *de facto*. De jure standards are arrived at through consensus by national or international standards bodies; for example, ITU-T and IETF. De facto standards are usually the result of an effort by one or more vendors to standardize a technology by forming a consortium. Sometimes, an early effort for de facto standards eventually transitions to de jure standards. There are many standards bodies that address issues related to networking and networking technologies. We briefly discuss some of them below.

1.12.1 International Telecommunication Union

ITU (http://www.itu.int/) plays the role of standardizing international telecommunications; it is a United Nations specialized agency. One of the key sections of ITU is known as ITU Telecommunication Standardization Sector (ITU-T). ITU-T brings both the public and private sectors together in an international forum. ITU-T is in charge of standardization of the international telephone numbering system, such as E.164 addressing. It also defines signaling protocol standards, and so on. Standards generated by ITU-T are called *Recommendations*.

You will see in the bibliography at the end of the book a number of ITU-T recommendations that we have referenced.

1.12.2 Internet Engineering Task Force

IETF (http://www.ietf.org/), as its web site says, "is a large, open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual." The IETF is structured around working groups, which then are grouped into areas. Areas have Area Directors (ADs). The ADs are members of the Internet Engineering Steering Group (IESG).

Standards generated by IETF are published as *Requests for Comments* (RFCs). This name stuck since its original use. The intent was to request for comments from the networking community; over time, it has become the avenue for IETF to publish standards. It may be noted that IETF also publishes informational documents as RFCs. Thus, each RFC is marked with a category such as standards track or informational. RFCs are available online from many web sites, for example, http://www.rfc-editor.org/. In the bibliography, we have referenced many routing-related RFCs.

In relation to IETF, there are several associated bodies. For example, the Internet Advisory Board (IAB) is chartered as a committee of IETF; it is also an advisory body of the Internet Society (ISOC). IAB handles architectural oversight of IETF activities, Internet Standards Process oversight and appeal. The IAB is also responsible for the management of the IETF protocol parameter registries.

Another important organization, the Internet Corporation for Assigned Names and Numbers (ICANN) (http://www.icann.org/), is an internationally organized, nonprofit corporation that now has responsibility for IP address space allocation, protocol identifier assignment, generic and country code top-level domain name system management, and root server system management functions. These services were originally performed by the Internet Assigned Numbers Authority (IANA) (http://www.iana.org/) and other entities. ICANN now performs the IANA function. Any new protocol parameter values identified by the IETF in a standard must be coordinated with the IANA to avoid any ambiguity.

1.12.3 MFA Forum

The MPLS and Frame Relay Alliance (MFA) Forum (http://www.mfaforum.org/) is an international, industry-wide forum consisting primarily of telecommunications and networking companies. It is focused on the creation of specifications on how to build and deliver MPLS, Frame Relay and ATM networks, and services. MFA also handles interoperability testing of different vendors' products.

1.13 Last Two Bits

In this section, we present two topics. The first, TLV, is a concept used in many protocols. The second topic is the protocol analyzer.

1.13.1 Type-Length-Value

An important concept used in protocol messages is *Type-Length-Value* (TLV). This concept is used in headers as well as the body of a packet, and by different layers of a networking

architecture. For simplicity, consider that the IP header includes 32-bit IP addresses, one for the source and the other for the destination. First, for each receiving end to interpret properly, the source and the destination address must be listed in the same order in the header. Second, such information has a well-defined structure: it is of a certain type (IP address, in this case), it is of certain length (32 bits in this case), and it contains a value (the actual IP address). When such information is well-structured within a packet header and because of the well-known nature of such information, it is not often necessary to explicitly indicate the type and the length; just the allocation of the 32-bit space for an IP address in the header suffices. That is, for well-structured information that has a well-defined position in a packet header, the type and the length can be *implicit*.

In many instances, the length may vary, or the type is preferred to be left open for future extensions of a protocol. To do that, the type and the length need to be explicitly declared along with the value—this notion is what is known as TLV. As you go through this book, you will see many examples of how the TLV notion is used. Briefly, when the type and the length are to be explicit, then the length for each of these must be clearly defined, so that the value can be allowed to be of variable length. For example, a byte may be assigned to indicate the type (so that up to 256 different types can be defined), followed by two bytes for the length (to indicate through its 16 bits the length of value, that is counted in bytes), such that the value field can be up to 65,536 ($=2^{16}$) bytes. Because of the well-defined structure of TLV, the information content can be processed and another TLV can follow. Furthermore, a nested notion of TLV is also possible where the "V" part may include one or more TLV encoded sets of data.

1.13.2 Network Protocol Analyzer

Packet formats for networking protocols are described in standards documents by respective standards bodies. Many details about what a protocol can do lie in the header of a packet. Yet, just by looking at a packet format and reading a standards document, it is still difficult to grasp. Network protocol analyzers are used to capture packets from live networks. By studying headers captured through such analyzers, it is often easier to understand a packet header, and more important, a protocol.

In this book, we have presented sample headers (or relevant parts of headers) associated with a few protocols to illustrate them. Sample header captures for many routing protocols are available from the website of public-domain network protocol analyzers such as WIRESHARK [743]. Additionally, packet headers of both request-and-response messages of a protocol can be studied from such captures—this is sometimes very helpful in understanding a protocol. Sample captures using WIRESHARK for many protocols are found at [744]. We strongly recommend studying sample captures from this site or similar sites for helping you to understand protocols better.

1.14 Summary

In this introductory chapter, we have presented a brief overview of networking, and the scope and goal of network routing. We have also presented architectural aspects of communication networks that are useful in network routing.

All of these components have a history and a set of issues to address. The state of network routing today is the result of theoretical progress, technological advances, and operational experience. It is also impacted by economic and policy issues. From which angle should these interconnected facets of network routing be viewed? In an email to the authors, Ibrahim Matta wrote:

> "To me, it would be invaluable to highlight concepts and techniques in routing that survived the various instances in different networks; for example, the concepts of scalability-performance tradeoff (scalability techniques include area hierarchy, virtual paths, periodic updates . . .), routing information propagation vs. forwarding, etc."

The rest of the book will explore each aspect of network routing, with a nod toward the historical part, due respect for the scalability-performance tradeoff, and lessons learned from operational experience.

Further Lookup

Early works in the development of ARPANET have been instrumental in understanding today's computer communication network. ARPANET design decisions are discussed in [464]. Cerf and Kahn's seminal paper [112] discusses the TCP/IP protocol communication. The design philosophy of the Internet is discussed, for example, in [143]. A comprehensive discussion on architecture can be found in [142].

A comprehensive summary of the telecommunication network can be found in Bell System's Engineering and Operations handbook, last published in 1984 [596]. While this book is almost a quarter century old and out of print, it still serves as a good resource book on basic telecommunication networking.

Naming, addressing, and routing are interrelated topics for a communication network. In 1978, Shoch [639] wrote "The name of a resource indicates what we seek, an address indicates where it is, a route tells how to get there." Shoch's original work has a lot to do with how we think about naming, addressing, and routing in the Internet, even today. Certainly we can no longer say that an address is where it is. Also, the naming and addressing are now blurry. For additional discussions on naming, addressing, and routing, see [285], [366], [497], [618].

Finally, the focus of this book, as the title says, is network routing. You may consult books such as [152], [386], [562], [668], [683], to improve your understanding of computer networking in general; in fact, it might be handy to have one of them with you as you read through this book. If you are interested in understanding in depth the OSI architecture and protocols that were developed for OSI, you may consult books such as [567], [684]. For a comprehensive discussion of protocols developed by IETF for the Internet, you may consult [211]. For a summary of technology-specific standards, see [560].

Exercises

1.1 Review questions:

(a) Given the IP address of a host and the netmask, explain how the network address is determined.

(b) Identify the key differences between the differentiated services architecture and the integrated services architecture.

(c) What is TLV?

1.2 Consider IP address 10.22.8.92 that is given to be part of a /14 address block. Determine the IP prefix it belongs to in the CIDR notation.

1.3 Consider IP address 10.21.5.90 that is given to be part of a /17 address block. Determine the IP prefix it belongs to in the CIDR notation.

1.4 From the TCP packet format, you will notice that it does not have a field that indicates the length of a TCP packet. How can you determine the TCP payload length, i.e., the length of the data carried in a TCP packet?

1.5 Why is it necessary to reserve some addresses from an address space rather than making all of them available?

1.6 Consider an IPv4 packet going through a router.

(a) Determine which fields in the header are minimally changed before the packet is forwarded.

(b) Which fields are also possibly changed at a router?

1.7 Are there any fields from the header of an IPv4 packet that are no longer maintained in the header of an IPv6 packet?

1.8 Investigate why the header fields in an IPv6 packet are significantly different than the header fields in an IPv4 packet.

1.9 Visit the IETF web-site (http://www.ietf.org/), and identify routing related working groups. Familiarize yourself with the type of routing protocols issues currently being addressed by these working groups.

1.10 Find out about other standards bodies, such as Institute of Electrical and Electronics Engineers (IEEE), American National Standards Institute (ANSI), Optical Internetworking Forum (OIF), especially regarding networking standards they are actively involved in.