1

TRANSMISSION CONTROL PROTOCOL/INTERNET PROTOCOL OVERVIEW

This first chapter provides an overview of Transmission Control Protocol (TCP)/Internet Protocol (IP), which is an Internet protocol stack to perform communications between two computers, or hosts, through the Internet. It is a collection of different protocols. A *protocol* is a set of rules that controls the way data is transmitted between hosts.

1.1 FUNDAMENTAL ARCHITECTURE

Figure 1.1 shows the Open Systems Interconnection (OSI) reference model and the Internet protocol stack. The International Standardization Organization (ISO) specifies a guideline called the OSI reference model. It is an abstract description for layered communications and computer network protocol design. It consists of seven layers, which are, from the bottom, physical, data link, network, transport, session, presentation, and application, as shown in Figure 1.1**a**. The application layer is the OSI layer closest to the end user, which means that both the OSI application layer and the user interact directly with the software application. This layer interacts with software applications that implement a communicating component. At the physical layer, data is recognized, or handled, as bits. At the data link layer, data is handled as frames. At the network layer, data is recognized as packets. In the transport layer, data is handled as segments or datagrams. In the remaining upper layers, users' information is recognized.

Advanced Internet Protocols, Services, and Applications, First Edition. Eiji Oki, Roberto Rojas-Cessa, Mallikarjun Tatipamula, and Christian Vogt.

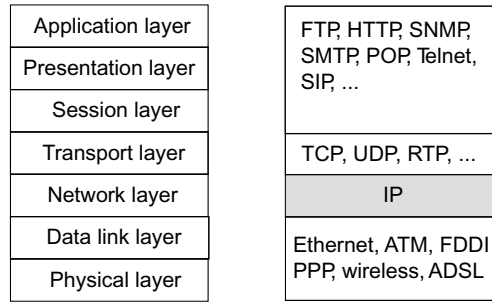

Figure 1.1 Layer model.

The Internet protocol stack is shown in Figure 1.1**b**, where we can see to which layer in the OSI reference model each protocol corresponds. For example, the Internet protocol corresponds to the network layer. TCP corresponds to the transport layer. The Internet protocol stack that includes several protocols, as shown in Figure 1.1**b**, is referred to as *TCP/IP*.

TCP/IP is being standardized by the Internet Engineering Task Force (IETF). It is widely used as the *de facto* standard protocol for building network equipment and internetworking. If the TCP/IP standard is used, computers can communicate with each other regardless of hardware and operating system.

Figure 1.2 shows a basic philosophy of TCP/IP. In the TCP/IP philosophy, the IP core network functions simply and quickly, while the functionality of the edges surrounding the core is complex. Edges can be hosts, edge routers, network boundaries, etc. The IP network based on this philosophy is scalable and flexible. This enables different complexities at the edge.

Application data and customer traffic can be transmitted on the IP layer because IP is the least common denominator, as shown in Figure 1.3. IP supports transport-layer protocols such as TCP, User Datagram Protocol (UDP), Real Time Protocol (RTP)/UDP, and Stream Control Transmission Protocol (SCTP), as shown in

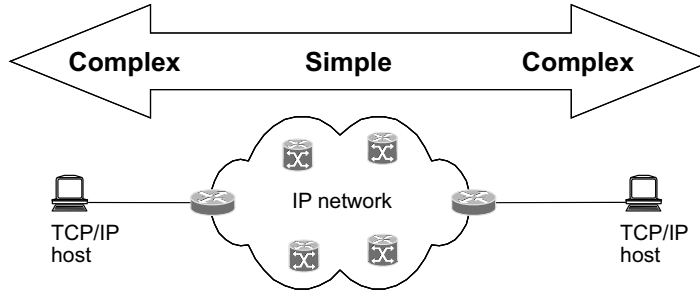

Figure 1.2 Basic philosophy of TCP/IP.

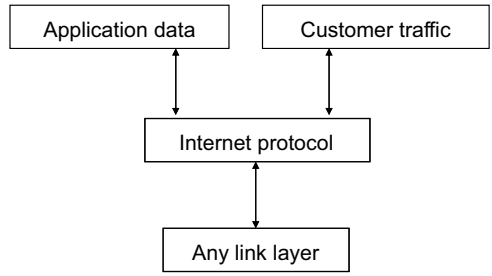

Figure 1.3 IP is the least common denominator.

Figure 1.4. IP works on link-layer protocols such as Ethernet, Point-to-Point Protocol (PPP) over Asynchronous Transer Mode (ATM), satellite, wireless, optical, and IP/Multiprotocol Label Switching (MPLS), as shown in Figure 1.5.

More than one network can be connected via IP, as shown in Figure 1.6. IP enables building a large-scale network where smaller networks, or subnetworks, are concatenated to build one large network. Inside each network, a node does not need to be connected with other nodes in the same network via IP. This is the way the Internet is built.

Figure 1.7 shows the protocol stack based on the OSI reference model and network equipment and protocols connecting layers. A repeater is a functional device of layer 1. It enhances signal level. A bridge, or switch, processes frames, for example Ethernet frames, corresponding to layer 2. Two bridges are connected via Ethernet protocol. A

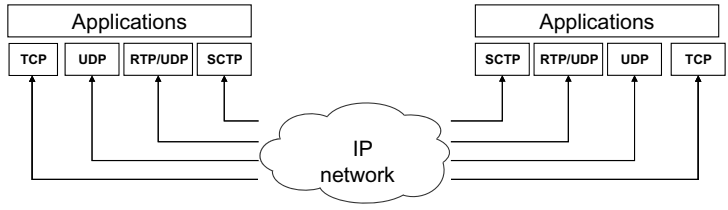

Figure 1.4 Any transport-layer protocols on IP.

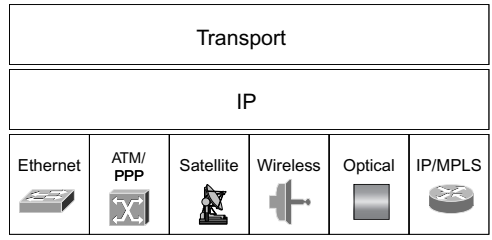

Figure 1.5 Any link-layer protocols under IP.

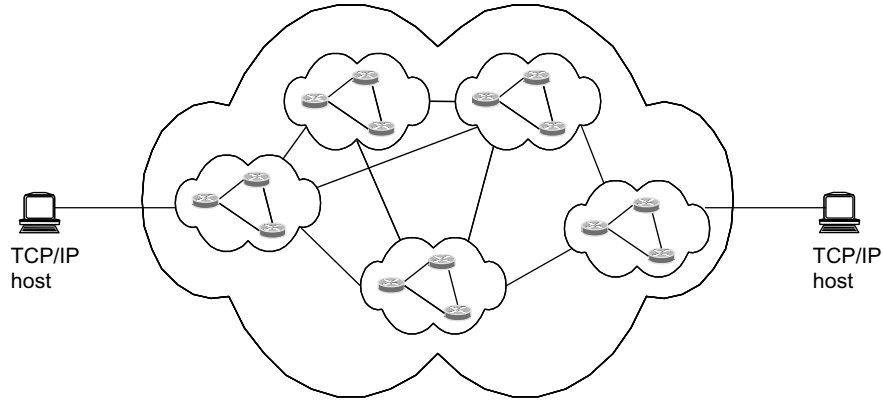

Figure 1.6 Interworking via IP.

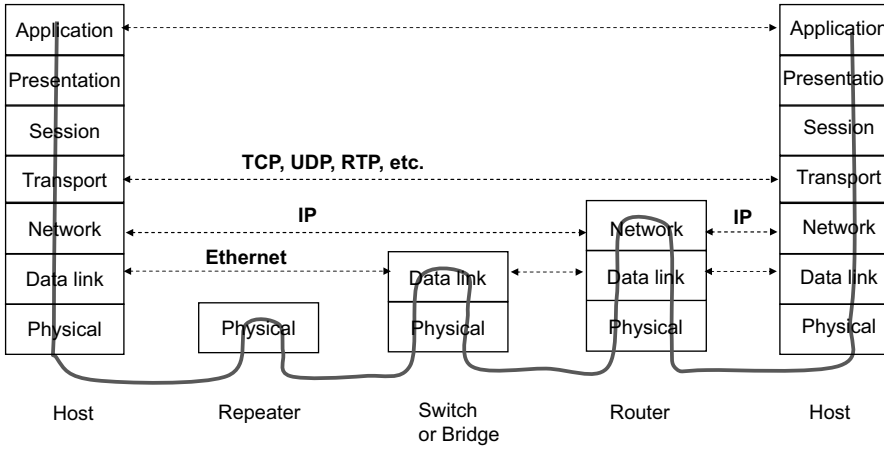

Figure 1.7 Protocol stack and connectivity.

router processes IP packets corresponding to layer 3. Two routers are connected via IP. Two hosts are connected via protocols for layers 4, 5, 6, and 7. For example, TCP, UDP, and RPT are used to connect two hosts on layer 4.

1.2 INTERNET PROTOCOL BASICS

Internet Protocol is a protocol in the network layer. It is used for sending data from a source host to its destination host where each host has a unique number. The IP header format is shown in Figure 1.8. Version specifies the IP version. Header length indicates the size of the header. Type of service is used to guide the selection of the

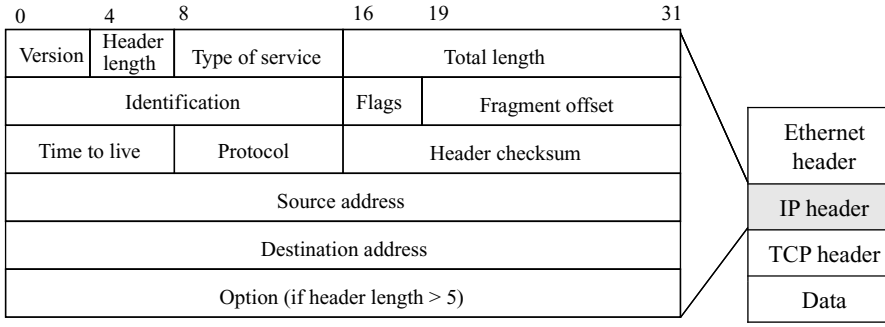

Figure 1.8 IP packet header format.

actual service parameters. Total length is the size of the IP packet including header and data. Identification is used for a particular purpose such as experimental work. Flag is used to control or identify fragments. Time-to-live (TTL) is a countdown field; every station must decrement this number by one or by the number of seconds it holds onto the packet. When the counter reaches zero, the TTL expires and the packet is dropped. Protocol specifies the protocol that is used to operate the data. There are several protocols dependent on applications. Example of well-known protocols are shown in Figure 1.9. Header checksum is used for error checking. Source address and destination address specify the IP address of the source and the destination, respectively.

1.2.1 Packet Header

When data is transmitted from a source host to a destination host in a network, it is divided into small pieces. Each piece is called a packet in an IP network. Sometimes it is called a frame, block, cell, or segment. The packet consists of a header and a payload. The header contains information such as the destination identification and length of the packet. The payload contains part of the body of the message. The packet

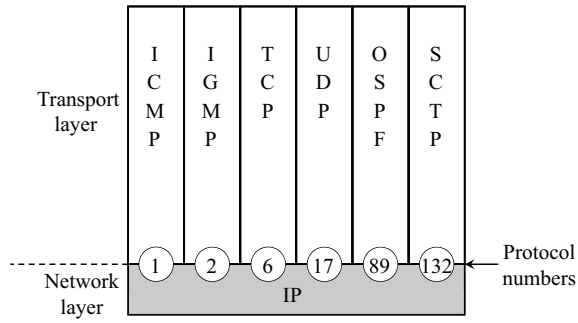

Figure 1.9 Some well-known protocols.

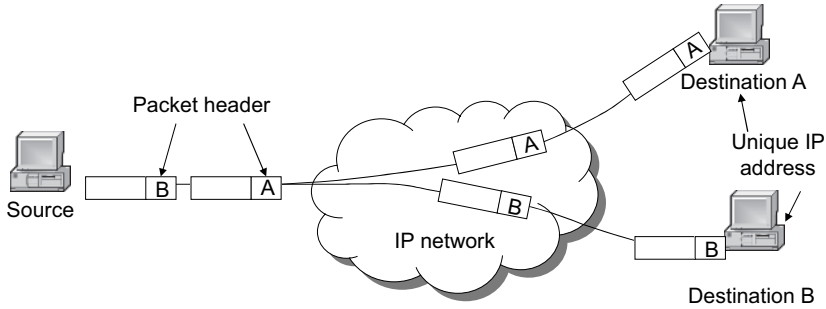

Figure 1.10 Routing using the packet header information.

is sent to the destination through an IP network along an appropriate available route. Each destination is assigned a unique identification number called an IP address. Figure 1.10 shows an example of routing using the information in the packet header. Packets are sent to destinations A and B, referring to the IP addresses.

Figure 1.11 shows how the data is wrapped when it is transferred to other layers. On the top layer, host A sends the raw data to host B. First, the raw data is wrapped as a TCP structure called the TCP packet. This raw data becomes the TCP payload, and the TCP header, which includes basic information about the TCP packet, is added. Next, the TCP packet is transferred from the transport layer to the network layer. In the network layer, the packet in the network layer is called an IP packet. Here, the TCP packet becomes the IP payload, and the IP header, which includes basic information

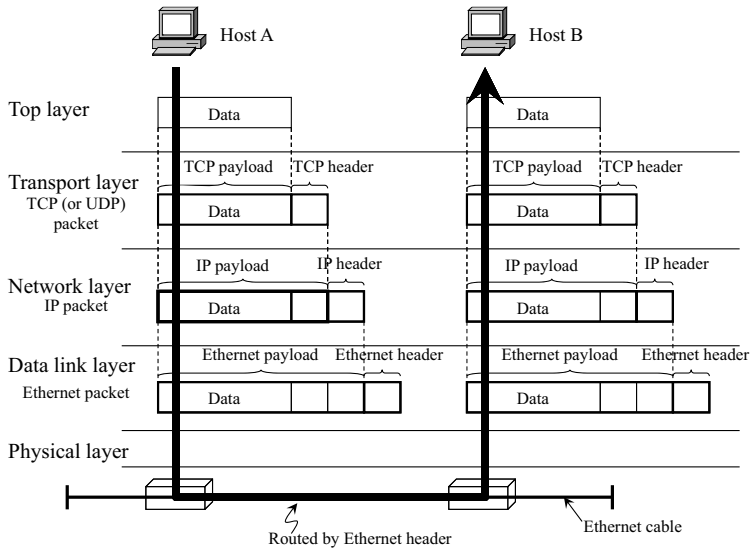

Figure 1.11 Wrapping and unwrapping the data as layers change.

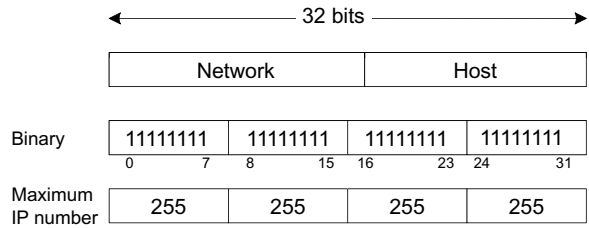

Figure 1.12 IP address structure.

about the IP packet, is added. At this point, the IP packet is transferred to the data link layer, where it is wrapped into an Ethernet packet. The IP packet becomes the Ethernet payload, and the ethernet header, which includes basic information about the Ethernet packet, is added. When the Ethernet packet is transferred back to the network layer, the Ethernet header is unwrapped and remains that way until the data reaches host B.

1.2.2 Internet Protocol Address

An IP address is a unique number assigned to each device in the computer network utilized by the Internet protocol.

In IP version 4 (IPv4), an IP address is defined by using 32 bits of binary numbers, which yields 4,294,967,296 (2^{32}) numbers. To enhance readability, an IP address is often represented as a decimal using four 8-bit sections, or octets. The number in each octet ranges between 0 and 255.

An IP address consists of the network address, which is represented by the higher order bits, and the host address, which is represented by the lower order bits, as shown in Figure 1.12. Based on control purpose, the number of bits in the network and host addresses are changeable. Figure 1.13 is an example of an IP address, 10.28.1.222. The number in each octet is converted to an 8-bit binary number.

1.2.3 Internet Protocol Classification

IP addresses are classified to enable regularity of management. They are limited to only 32 bits and are comprised of a network and host address. The number of hosts is lower if more bits are allocated for the network, and conversely, the number of networks is lower if more bits are allocated for the hosts.

Decimal	10	28	1	222
Binary	00001010	00011100	00000001	11011110

Figure 1.13 Example of IP address.

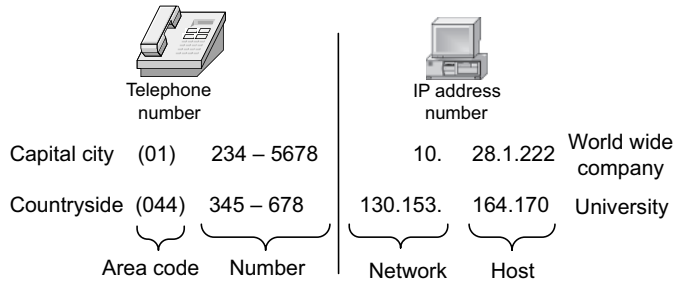

Figure 1.14 Comparison of the structure of the telephone number and IP address.

The network and host addresses in the IP address system are similar to the area code and the client number, respectively, in the telephone system of countries such as the United Kingdom, Japan, India, and Thailand. There are many users in cities, while there are fewer users in small towns. Since the number of capital cities is smaller than the number of towns, fewer digits of area code are enough to support the cities. More digits are needed in the area codes of towns.

In Figure 1.14, an example of the telephone number system and IP address system are compared. The telephone system contains nine digits. In the capital city, the first two digits specify the area code and the last seven digits specify the client number. This allows for more than 10 million numerical combinations. In the countryside, the first three digits are used for the area code and the last six digits are for the client number. This allows for approximately 1 million numerical combinations. In the IP address system, a company with branches throughout the world would have several hosts. A university located in a single country would require fewer hosts than the global company. The company would use the first octet to specify a network address and the last three octets for the host address. The network address of the university would be indicated by the first two octets, and the last two octets would indicate the host address.

The IP address can be classified into four classes: A, B, C, and D. Figure 1.15 shows IP addresses of each class.

- *Class A* is indicated the starting bit of "0." The first eight bits specify a network address. The last 24 bits specify a host address. The range of the number in the first octet is between 0 and 127. However, 126 numbers are usable since IP addresses starting with 0 and 127 are reserved. 16,777,214 hosts can be obtained for one network.
- *Class B* is indicated when the first two bits are "10." The first 16 bits specify a network address and the other 16 bits specify a host address. The range of the number in the first octet is between 128 and 191. 16,384 networks address are available. 65,532 hosts can be obtained for each network.
- *Class C* is indicated when the first three bits are "110." The first 24 bits specify a network address and the other eight bits specify host address. The range of the

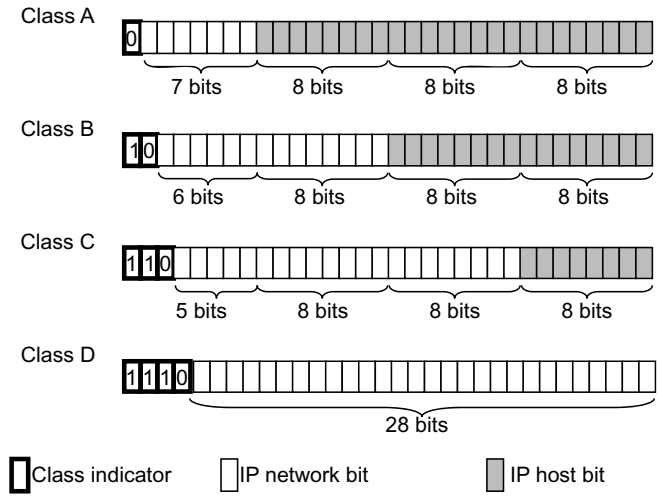

Figure 1.15 IP classification.

number in the first octet is between 192 and 223. 2,097,152 networks address are available. 254 hosts can be obtained for each network.

- *Class D* is indicated when the first four bits are "1110." This class is used for multicast purpose. The range of the number in the first octet is between 224 and 239.

1.2.4 Subnet and its Masking

Subnet is a way to group, or subnetwork, IP addresses in the same network. Subnet mask is a parameter to indicate network address, by bit "1," and host address, by bit "0." Usually, the network address and the host address are separated based on IP class as explained in the previous subsection. However, the mask can vary due to IP address management. The number of the subnet mask can be written by "/xx" after the IP address, where xx is the number of network bits. Figure 1.16 shows an example of

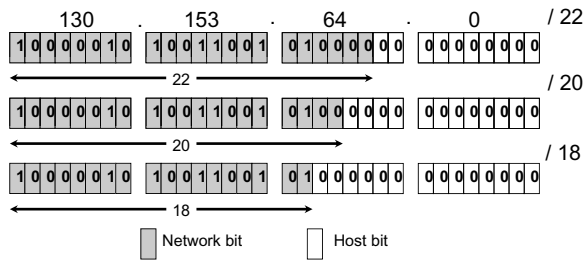

Figure 1.16 Subnetting to categorize hosts.

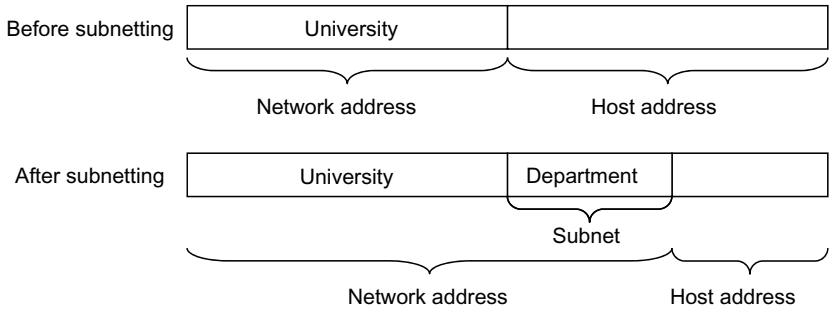

Figure 1.17 Subnet masking.

the IP address 130.153.64.0. If the first 22 bits are masked as a network address, it is notated as 130.153.64.0/22. The network address is also called a prefix.

There are two main functions of subnetting:

- *To categorize hosts into a group*. Figure 1.17 illustrates subnetting for a university network. Before subnetting, the IP address has a network address, which indicates the IP number of the university, and a host address, which is the address of the hosts in the university. If the IP address is randomly distributed to hosts, it is difficult to manage IP addresses since IP addresses are not classified. For example, if the computer engineering department allows File Transfer Protocol (FTP) service while it is blocked in the electrical engineering department, it is necessary to search the IP addresses that belong to electrical engineering and block each and every IP. By subnetting, it is possible to block the group of IP addresses that belong to electrical engineering. In Figure 1.17, subnetting is used to divide departments. After subnetting by using bits to indicate the department, the new network address includes the department subnet number.
- *To support a routing table*. A switch uses this table to forward data to a specific port. Figure 1.18 shows routing tables with and without subnetting. Hosts with IP 130.153.8.XXX are connected to the port $E0$ of the router. Hosts with IP 130.153.3.XXX are connected to the router at the $E1$ port. Based on IP classification, the network address of these IP addresses is 130.153.0.0 because these IPs are class B. In the routing table without subnetting, the data that requests a host with IP address 130.153.8.160, for which the network address is 130.153.0.0, is forwarded to both ports $E0$ and $E1$. This is incorrect because the host with IP address 130.153.8.160 is connected at the $E0$ port of the router. With subnetting, the first 24 bits are masked as a network address and the last 8 bits are a host address. In the routing table, the data that requests a host with network address 130.153.8.0 is forwarded to port $E0$ and 130.153.3.0 to port $E1$. The router can distinguish the IP addresses and forward to the correct port.

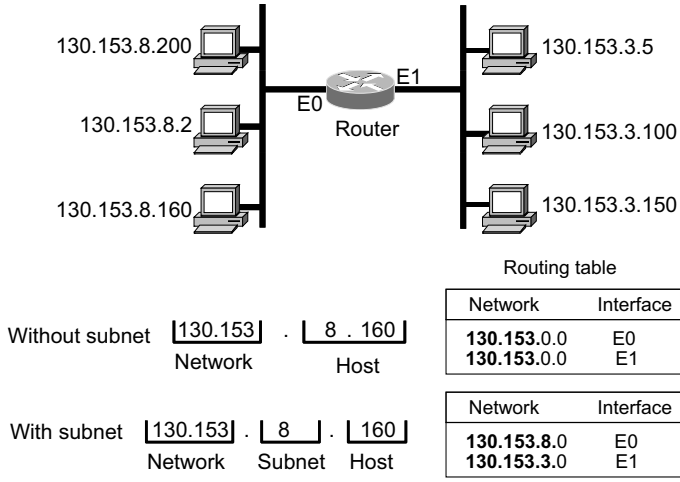

Figure 1.18 Subnetting to support a routing table.

1.2.5 Subnet Calculation

This section explains how to calculate the subnet from a specific IP address. If we know a host IP address and subnet mask, what can we obtain from the subnet calculation?

- *Subnet IP* is an address number that indicates a subnetwork within a larger network. Subnet IP = (Host IP) AND (Subnet mask).
  Trick: If the bit of a subnet mask is "1," the bit of the subnet IP is the same as the bit of the host IP. Otherwise, the bit of the subnet IP is "0."
- *Broadcast IP* is used to send information to all hosts that belong to the same network. Broadcast IP = (Inverted Subnet Mask) OR (Subnet IP).
  Trick: It is obtained by changing all of the "0" bits in the host address of the subnet IP to "1."
- *First* and *last IP* indicate the range of IP addresses that hosts on the same network can use. The first IP = (Subnet IP) + 1, and the last IP = (Broadcast IP) − 1.
- *Number of hosts* indicates the number of hosts on the subnetwork. It is counted from the first IP to the last IP, which is (Last IP) − (First IP) + 1.

Figure 1.19 is an example of subnet calculation for 130.153.2.160/26. "/26" represents the number of bits in the mask. It indicates that 26 bits are required for the network address. The subnet mask becomes 11111111.11111111.11111111.11000000 (255.255.255.192 in decimal). The subnet IP is 130.153.2.160 AND 255.255.255.192 = 130.153.2.128. To calculate broadcast IP, the inverted subnet mask is needed. The inverted subnet mask is 00000000.00000000.00000000.00111111 (0.0.0.63 in decimal). Broadcast IP is 0.0.0.63 OR 130.153.2.128 = 130.153.2.191. The first IP is 130.153.2.128 + 1 = 130.153.2.129 and the last IP is 130.153.2.191

	Network address			Host address
Host IP	10000010	10011001	00000010	10|100000
	130 .	153 .	2 .	160
Subnet mask	11111111	11111111	11111111	11|000000
	255 .	255 .	255 .	192
Subnet IP	10000010	10011001	00000010	10|000000
	130 .	153 .	2 .	128
Inverted subnet mask	00000000	00000000	00000000	00|111111
	0 .	0 .	0 .	63
Broadcast IP	10000010	10011001	00000010	10|111111
	130 .	153 .	2 .	191
First IP	10000010	10011001	00000010	10|000001
	130 .	153 .	2 .	129
Last IP	10000010	10011001	00000010	10|111111
	130 .	153 .	2 .	190

Figure 1.19 An example of subnet calculation.

$-1 = 130.153.2.190$. The number of hosts available on this subnetwork is $130.153.2.190 - 130.153.2.129 + 1 = 62$ hosts.

The subnet calculation is also used to design IP addresses for the hosts in the network. The following example provides insight into how to design the IP addresses.

How do you design the IP address to support these requirements for XYZ branch if you are a network administrator?

- ABC company has several branches throughout the world.
- XYZ branch uses IP 10.28.1.xxx.
- At XYZ branch, each department requested the following number of computers:

Information System (IS)	18 computers
Account (ACC)	16 computers
Human resource (HR)	12 computers
Production	25 computers
General affair (GA)	10 computers

The IP 10.28.1.xxx is a class A address. The network IP is 10.0.0.0. However, the IP of XYZ branch starts with 10.28.1.xxx. Production requests 25 computers, which is the highest number of any department. Using five bits as the host is enough to support 2^5, or 32, IP addresses, including subnet IP and broadcast IP, and enough for five departments. Therefore, we use five bits for the host address and the remaining 27 bits for network address. The IP address in this network is 10.28.1.xxx/27. The subnet mask becomes 11111111.11111111.11111111.11100000 (255.255.255.224

	Subnet IP	Broadcast IP	First IP	Last IP
Subnet mask	11111111 . 11111111 . 11111111 . 111 00000 255 . 255 . 255 . 224			
IS	00001010 . 00011100 . 00000001 . 000 00000 10 . 28 . 1 . 0	10.28.1.31	10.28.1.1	10.28.1.30
ACC	00001010 . 00011100 . 00000001 . 001 00000 10 . 28 . 1 . 32	10.28.1.63	10.28.1.33	10.28.1.62
HR	00001010 . 00011100 . 00000001 . 010 00000 10 . 28 . 1 . 64	10.28.1.95	10.28.1.65	10.28.1.94
Production	00001010 . 00011100 . 00000001 . 011 00000 10 . 28 . 1 . 96	10.28.1.127	10.28.1.97	10.28.1.126
GA	00001010 . 00011100 . 00000001 . 100 00000 10 . 28 . 1 . 128	10.28.1.159	10.28.1.129	10.28.1.158

Figure 1.20 IP address information for each department.

in decimal). Information about subnet IP, broadcast IP, first IP, and last IP is shown in Figure 1.20.

1.3 ROUTING

The process of forwarding a packet across the Internet from the source node to the destination node is called *routing*. Routing proceeds in a hop-by-hop manner, where each node on the packet's path, from the source node via routers to the destination node, independently determines a neighbor closer to the destination node and hands the packet on to that neighbor. The routing process completes when that neighbor happens to be the destination node itself.

Routers maintain a *routing table* that maps a packet's IP destination address onto the IP address of a neighboring router closer to the destination node, or to determine that the destination node is itself a neighbor. A routing table may be manually preconfigured into a router, but in general, routers participate in a routing protocol to exchange the topological information they need to build a routing table automatically. Given that the prefix of an IP destination address already locates the destination node, it is typically sufficient for a router to perform a routing table look-up based on only the IP address prefix. The interface identifier of the IP destination address is needed only when the packet is delivered to the destination node in the final forwarding step.

To understand the Internet routing system, it is useful to differentiate between two types of networks: networks that carry transit traffic for other networks, or so-called *providers*, and networks that do not, or so-called *edge networks*. Whereas edge networks can simply forward packets to their respective provider for further

transportation toward the destination node, providers need to know about the interconnectivity of other providers in order to make forwarding decisions. The routing across providers and within edge networks is explained in the proceding section.

1.3.1 Routing across Providers

Providers need to explore Internet topology at large in order to make routing decisions. They use the Border Gateway Protocol (BGP) to tell each other where they are located and which providers they are able to reach [1]. Providers identify themselves in BGP through their IP address prefixes. This is possible because the IP address prefixes of different providers are mutually disjointed. Figure 1.21 illustrates three providers with IP address prefixes 110.0.0.0/8, 120.0.0.0/8, and 130.0.0.0/8, respectively.

Each provider gathers the information received through BGP in a global routing table. For example, in Figure 1.21, the global routing table of provider 1 shows that provider 2 is neighboring to the right and that provider 3 can be reached via provider 2. Therefore, provider 1 sends rightward all packets destined to IP address prefixes 120.0.0.0/8 and 130.0.0.0/8. The global routing table of provider 2 points to the right for provider 3 and to the left for provider 1. Provider 2 thus forwards packets destined to IP address prefixes 110.0.0.0/8 and 130.0.0.0/8 to the left and to the right, respectively. Finally, the global routing table of provider 3 shows that providers 1 and 2 can both be reached to the left. So provider 3 forwards leftward all packets destined to IP address prefixes 110.0.0.0/8 and 120.0.0.0/8.

In addition to BGP, internal routing protocols are used by routers within the same provider's network to jointly explore their local interconnectivity. The most common internal routing protocol is known as *Open Shortest Path First* (OSPF) [2, 3]. There are three important differences between BGP and OSPF:

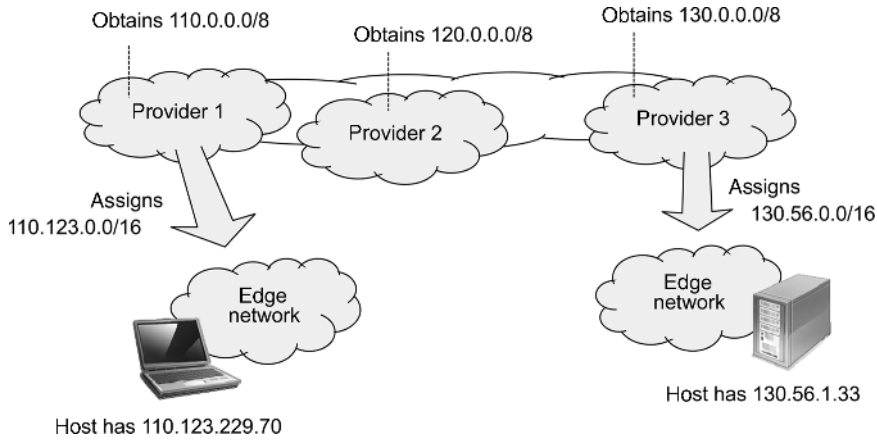

Figure 1.21 Provider-assigned addressing.

- *Scope of operation:* BGP is used to explore interconnectivity between provider networks, and hence to explore Internet topology at large, whereas OSPF is used to explore the interconnectivity of routers within the same network. BGP's scope of operation is therefore global, while OSPF's scope of operation is local.
- *Routing policy:* Since BGP operates across the borders of administratively separate domains, it becomes a tool to express routing policy. OSPF, on the other hand, is not used to express routing policy because it operates within the borders of a single administrative domain.
- *Topology resolution:* The global scope of an external routing protocol implies higher scalability requirements for external routing protocols compared with internal routing protocols. The extra scalability often comes at the cost of a coarser resolution of the explored Internet topology. BGP, for example, supplies only the direction in which a certain destination can be reached. It supplies neither the sequence of provider networks through which the destination is reached, nor the sequence of routers traversed within any given provider network. OSPF, on the other hand, can afford to supply the sequence of traversed routers for a given destination, because it operates within the smaller scope of a single network.

It is these three differences that suggest the use of different protocols within and across provider networks.

1.3.2 Routing within Edge Networks

Figure 1.22 illustrates the typical topology of an edge network. Some of the links in an edge network are *access links* to which nodes can directly attach. The access link for a stationary node may be a single physical wire, such as an Ethernet cable or a Digital Subscriber Line (DSL), or an *access point* that connects a wireless access technology to the fixed network. An access link may include more than one access point for extended geographic coverage. The first edge router for packets that the node sends to destinations off-link is an *access router*. Edge networks communicate with each other through providers.

The preferred addressing and routing in edge networks differs from the addressing and routing across providers. Edge networks are called upon to use a portion of the IP address prefix allocated to their respective providers. The left edge network in Figure 1.21 thus gets a portion of provider 1's IP address prefix, say, 110.123.0.0/16. And the right edge network gets a portion of provider 3's IP address prefix, say, 130.56.0.0/16. Accordingly, a host in the left edge network will configure an IP address that belongs to provider 1, and a host in the right edge network will configure an IP address that belongs to provider 3. As shown in Figure 1.21, these IP addresses are 110.123.229.70 and 130.56.1.33, respectively. Similar to routers within a provider, routers within an edge network use an internal routing protocol, typically OSPF, to jointly explore their interconnectivity.

The reason for reusing providers' IP address prefixes in edge networks, instead of giving edge networks provider-independent IP address prefixes, is scalability. Edge networks become implicitly reachable via their respective provider's entry in the global routing table. They do not need their own routing table entry. In specialist

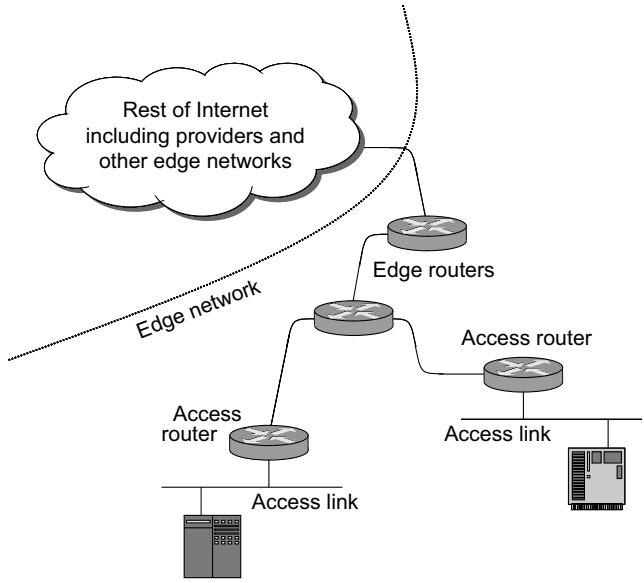

Figure 1.22 Edge network topology.

terms, IP addresses used inside an edge network are called *aggregatable* with the address space of the edge network's provider. Thus, in the example of Figure 1.21, packets destined to the host shown on the lower left find the destination edge network by following the routing table entries for provider 1, and provider 1 forwards the packets rightward.

1.3.3 Routing Scalability

Although the reuse of providers' IP address prefixes in edge networks aids the scalability of the global routing system, edge network operators frequently demand their own provider-independent IP address prefix. The reason is that the reuse of providers' IP address prefixes limits the flexibility of edge networks in two ways:

- Reuse of providers' IP address prefixes implies that, when an edge network becomes multihomed, it will obtain IP address prefixes from each of its providers, and hosts will configure IP addresses from each such prefix. Packets that are sent from or destined to either of these IP addresses will then be routed via the provider to which the IP address belongs. Rerouting for the purpose of failover or load balancing is then impossible.
- Reuse of providers' IP address prefixes causes a slight form of provider lock-in. It requires edge networks to change their IP address prefix whenever they change providers, and this requires them to undergo renumbering. *Renumbering* is an expensive and time-consuming procedure that requires substantial manual efforts.

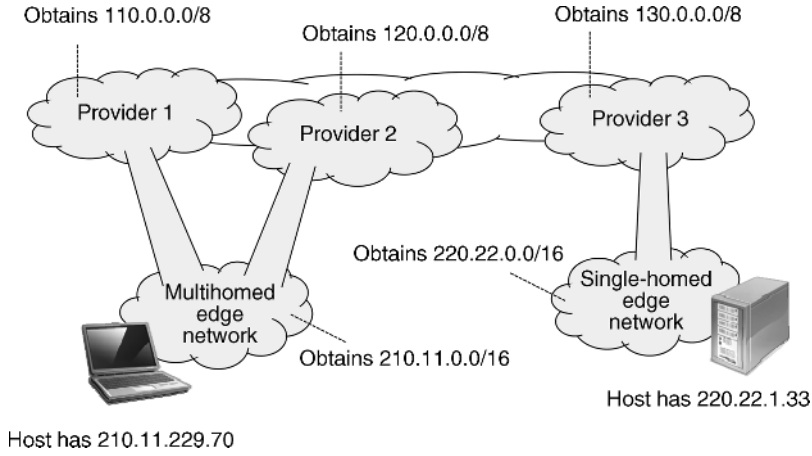

Figure 1.23 Provider-independent addressing.

It implies IP address changes to hosts and network equipment edge-network-wide. It affects routers, host, Domain Name System (DNS) and Dynamic Host Configuration Protocol (DHCP) servers, firewalls, intrusion detection systems, remote-monitoring systems, load balancers, as well as scripts and configuration files.

Consequently, it is no surprise that edge network operators are reluctant to reuse providers' IP address prefixes. They prefer to have individual IP address prefixes like providers.

To enable multihoming and eliminate renumbering, the current practice is to assign edge networks provider-independent IP address prefixes. This is illustrated in Figure 1.23 for the multihomed edge network at the lower left. Accordingly, the IP addresses of hosts in the edge network are no longer bound to provider 1 or to provider 2. Packets sent from such an IP address can be routed to either provider.

On the other hand, provider-independent addressing has an adverse impact on the scalability of the Internet routing system. In order for packets to find their way toward a provider-independent IP address, the IP address prefix of the edge network needs to be advertised among providers via BGP. This implies higher router load and, thus, less efficient packet forwarding.

- *Increased routing table size:* Global routing tables must list the provider-independent IP address prefixes of edge networks separately because these IP address prefixes can no longer be aggregated with an IP address prefix of the provider. The result is an increase in the size of global routing tables. This effect can be substantial because the number of potentially multihomed edge networks is orders of magnitude larger than the number of providers.
- *More frequent routing table updates:* For an edge network to change the provider based on which packets will reach it, IP address prefix announcements in BGP

must be updated. The result is an increase in the update frequency of the global routing table.

Even though a scalable Internet routing system is in the interest of edge network operators and providers, the selfish yet legitimate interest of each individual network operator is putting the scalability of the Internet routing system under pressure. This problem is not new. In fact, it has been a concern for almost the entire life of the Internet. Nonetheless, there has never been a major change in the Internet routing architecture. Addressing has always been done according to the Internet protocol, where version 6 of the protocol does not differ conceptually from version 4. And routing has not changed since the introduction of BGP in the 1980s. There was only one evolutionary step, Classless Inter-Domain Routing (CIDR) which is a means to improve the efficiency of IP address allocation and aggregation.

Mitigating the scalability problem of the Internet routing architecture is important to enable continued efficient functioning of the Internet and reasonable upgrade intervals for routers. A scalability problem always becomes more and more significant as the affected system grows. The Internet routing scalability problem may have been acceptable in the early days of the Internet, when the Internet was still small. It may even still be acceptable today. But the problem is becoming more and more noticeable, and it will continue in this direction.

For a few years now, therefore, people are more seriously considering improvements to the Internet routing system—not only in the engineering community, but also and foremost in the research community. In October 2006, the Internet Architecture Board (IAB) had a meeting about the routing scalability problem. The problem was brought up foremost by router vendors, who had decided to request that the engineering community do something about the problem. The IAB decided to further investigate the severity of the problem and possible solutions. It chartered a routing research group inside the Internet Research Task Force (IRTF) for these efforts. In 2010, the routing research group released a recommendation for the Internet Engineering Task Force (IETF) on possible solutions to the routing scalability problem [4]. The recommendation considers a diverse set of proposed solutions, ranging from backwards-compatible evolutionary techniques to revolutionary clean-slate approaches. Some solutions tackle the problem with more scalable router architectures, others advocate changes to IP addressing schemes and routing protocols.

REFERENCES

1. Rekhter, Y. and Li, T., “A Border Gateway Protocol 4 (BGP-4),” IETF RFC 1771, Mar. 1995.
2. Moy, J., “OSPF version 2,” IETF RFC 2328, Apr. 1998.
3. Coltun, R., Ferguson, D., Moy, J. and Lindem, A., “Open Shortest Path First for IPv6,” IETF RFC 5340, Jul. 2008.
4. Li, T. (Editor), “Recommendation for a Routing Architecture,” IRTF RFC 6115, Feb. 2011.