ELEC S212

Network Programming and Design

Unit 3

The essentials of TCP/IP

香港公開大學
THE OPEN UNIVERSITY OF HONG KONG
科技學院 School of Science and Technology

Course team

Developers: Jacky Mak, Consultant
John Wu, Consultant

Designer: Ross Vermeer, ETPU

Coordinator: Dr Philip Tsang, OUHK

Member: Dr Steven Choy, OUHK

External Course Assessor

Prof. Cheung Kwok-wai, The Chinese University of Hong Kong

Production

ETPU Publishing Team

The Open University of Hong Kong
Ho Man Tin, Kowloon
Hong Kong

This course material is printed on environmentally friendly paper.

Contents

Overview

In *Unit 1* you learned the basics of computer networks. You learned that the Internet depends on the TCP/IP suite of protocols, as do a number of network operating systems. Because of the increasing popularity of the Internet, you therefore need to learn TCP/IP to manage a network competently. In *Unit 2* you learned about some of the technologies related to the physical and data link layers of the OSI network model.

This unit explores the different Internet protocols, including the **Internet Protocol** (**IP**) from the Internet layer, and the **Transmission Control Protocol** (**TCP**) and the **User Datagram Protocol** (**UDP**) from the transport layer. The mechanisms of these protocols are analysed and illustrated. Other related protocols such as routing protocols, the **Address Resolution Protocol** (**ARP**) and the **Internet Control Message Protocol** (**ICMP**) are also discussed in this unit.

The unit begins by building on previous network concepts and examining how TCP/IP networks are managed and maintained. You will learn more about TCP/IP naming and addressing. Details of Internet naming schemes, IP addressing, subnetting, address resolution and address administration will be discussed. You will then explore the routing principles in a TCP/IP network and look at troubleshooting in a TCP/IP network.

The TCP/IP transport layer provides two dominant protocols, UDP and TCP, to facilitate process-to-process communication. UDP is a connectionless, unreliable communication service that incurs a very low overhead. TCP, on the other hand, is a connection-oriented, reliable communication service. You will also look at the concept of port and socket addresses in the TCP/IP transport layer. By using different TCP or UDP port numbers, hosts can simultaneously set up multiple connections with different destinations. You will also explore the TCP connection and control algorithms and examine the flow control and retransmission characteristics of TCP.

In short, this unit:

- describes the concept of the Internet architecture;
- discusses the different roles and features of IP, TCP and UDP in Internet communication;
- discusses IP naming, address translation mechanisms and IP routing;
- discusses the differences between connection-oriented and connectionless communication;
- outlines UDP and TCP protocol mechanisms; and
- analyses the benefits and limitations of TCP and UDP.

This unit is intended to take you five weeks (or approximately 40 hours) to complete.

Introduction

The origin of the Internet protocol suite

The beginning of the Internet can be traced back to almost half a century ago. In the 1960s, computers were really expensive, and the increasing importance of computers made it natural that people would begin looking for more effective ways to connect them so that computers could be shared among geographically distant users.

At the time, the telephone network was the world's dominant communication network. It was based on circuit switching in which a fixed connection must be established between two hosts before communication can begin. The connection is dedicated to the two communicating hosts for the whole communication session. However, data exchange between computers is usually 'bursty'. What does this mean? For example, when a user enters a command via a terminal, the command is sent via the network to a central mainframe computer. The computer receives the input command, performs some kind of processing, and then sends the result back to the user's terminal. Then the computer waits for the next command — indefinitely. Obviously, there are long gaps between each burst of data transmission. Using circuit switching for this kind of bursty data transmission is obviously inefficient since the network is idle most of the time.

In the early 1960s, some visionary researchers such as Robert Taylor and Joseph Licklider pioneered calls for a global network to address interoperability issues, such that computers manufactured by different vendors could communicate with each other without regard to their physical locations. Concurrently, several notable researchers including Donald Davies (NPL), Paul Baran (RAND Corporation), and Leonard Kleinrock (MIT) began to research principles of networking between separate physical networks. This led to the development of **packet switching**.

The research work began to materialize as the Advanced Research Projects Agency (ARPA) of the US Department of Defense, in collaboration with US universities and other research organizations, built the first packet-switching network — the **ARPANET**. The first two nodes of the ARPANET — between Leonard Kleinrock's lab at UCLA and Douglas Engelbart's lab at SRI in Menlo Park, California — were connected on October 29, 1969. The ARPANET is the direct ancestor of today's public Internet.

Today, the Internet is a global system of interconnected networks of computer networks that interchange data by packet switching using the standardized Internet Protocol Suite (or simply, TCP/IP). The **Transmission Control Protocol** (**TCP**) and **Internet Protocol** (**IP**) were developed in 1974 for all hosts in the ARPANET. In 1984, the ARPANET was split into two parts. One part represented the beginnings of what was to become the Internet. The other part was called MILNET to represent the military nature of the network.

The Internet architecture

A key feature of the ARPANET (and the Internet that subsequently evolved from it) was its ability to accommodate many different kinds of machines. As long as individual machines could 'speak' the packet-switching protocols of the new, anarchic network, they could be incorporated into the network.

The Internet's architecture has similarly open standards. It allows the integration of systems of all sizes, from different system vendors, with different operating systems. Consequently, it has become enormously popular and is now a common structure for internetwork communication.

The **Internet Reference Model**, as depicted in figure 3.1, is also known as the **TCP/IP Model**, the **DoD Model**, or the **ARPANET Reference Model**. The Internet architecture, as described in RFC 1122, is divided into four layers:

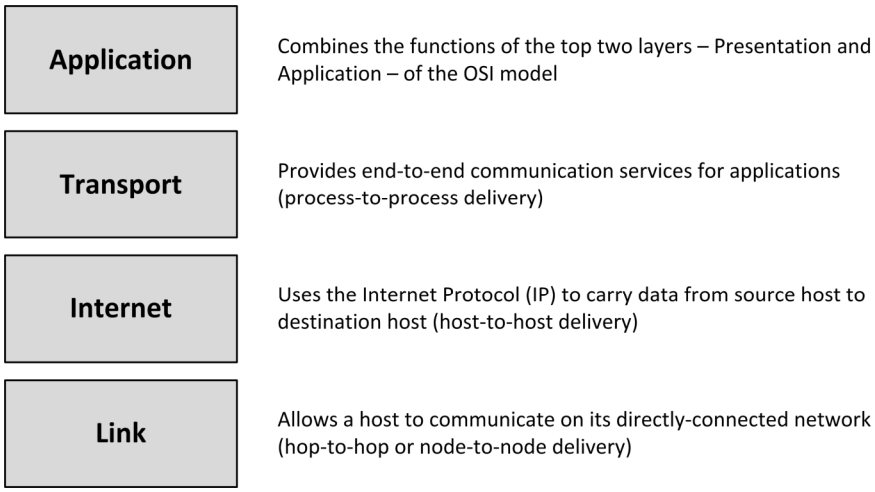


Figure 3.1 The 4-layer Internet model as specified in RFC 1122

The Internet Reference Model is one of the **Request for Comments** (**RFC**s) which are memoranda published by the **Internet Engineering Task Force** (**IETF**) describing methods, behaviors, research or innovations applicable to the working of the Internet and Internet-connected systems. The RFC format is the official publication channel for the IETF, and some of the RFCs are adopted by the IETF as Internet standards.

Each RFC is assigned a unique number. Some of the most important RFCs include:

- RFC 768 (1980) — 'User Datagram Protocol'
- RFC 791 (1981) — 'Internet Protocol'
- RFC 793 (1981) — 'Transmission Control Protocol'
- RFC 826 (1982) — 'An Ethernet Address Resolution Protocol'
- RFC 854 (1983) — 'Telnet Protocol Specification'

- RFC 920 (1984) — 'Domain Requirements'
- RFC 959 (1985) — 'File Transfer Protocol'
- RFC 1122 (1989) — 'Communication Layers'
- RFC 1939 (1996) — 'Post Office Protocol — Version 3'
- RFC 2131 (1997) — 'Dynamic Host Configuration Protocol'
- RFC 2616 (1999) — 'Hypertext Transfer Protocol — HTTP/1.1'
- RFC 2821 (2001) — 'Simple Mail Transfer Protocol'

There are many others. You can retrieve the RFCs from IETF's RFC page at the following URL:

http://www.ietf.org/rfc.html.

You may have encountered the 5-layer Internet model in textbooks or webpages on computer networking. Different authors have interpreted the RFCs differently, particularly as to whether the link layer covers physical layer issues, i.e. as to whether or not a 'hardware layer' is assumed below the link layer. Some authors have tried to use other names for the link layer, such as 'network interface layer' or 'network access layer', in an effort to avoid confusion with the data link layer of the 7-layer OSI model. Others have attempted to map the Internet model onto the 7-layer OSI Model. The mapping often results in a 5-layer Internet model, in which the link layer is split into a data link layer on top of a physical layer.

The following table shows the layer names used in RFC 1122, along with the names used in some widespread computer networking textbooks today.

Table 3.1 Layer names used in RFC 1122 and widespread computer networking textbooks

<table>
<tr><th></th><th>RFC 1122</th><th>Tanenbaum (2002)</th><th>Kurose and Ross (2007)</th><th>Dye, McDonald and Rufi (2007)</th><th>Forouzan (2006)</th><th>Stallings (2006)</th></tr>
<tr><td></td><td>4 layers</td><td>4 layers</td><td>4 layers</td><td>4 layers</td><td>5 layers</td><td>5 layers</td></tr>
<tr><td>L5</td><td>Application</td><td>Application</td><td>Application</td><td>Application</td><td>Application</td><td>Application</td></tr>
<tr><td>L4</td><td>Transport</td><td>Transport</td><td>Transport</td><td>Transport</td><td>Transport</td><td>Host-to-host or Transport</td></tr>
<tr><td>L3</td><td>Internet</td><td>Internet</td><td>Internet</td><td>Internetwork</td><td>Network</td><td>Internet</td></tr>
<tr><td>L2</td><td rowspan="2">Link</td><td rowspan="2">Host-to-network</td><td rowspan="2">Link</td><td rowspan="2">Network interface or Network access</td><td>Data link</td><td>Network access</td></tr>
<tr><td>L1</td><td>Physical</td><td>Physical</td></tr>
</table>

In this course we will adhere to the 4-layer model described in RFC 1122. Keep in mind, however, that the 5-layer model is also valid, so you should be aware of the different layer names that are also in widespread use in the literature.

The TCP/IP Model is often compared with the 7-layer OSI Reference Model. The following figure shows where the predominant network protocols reside in these two models respectively:

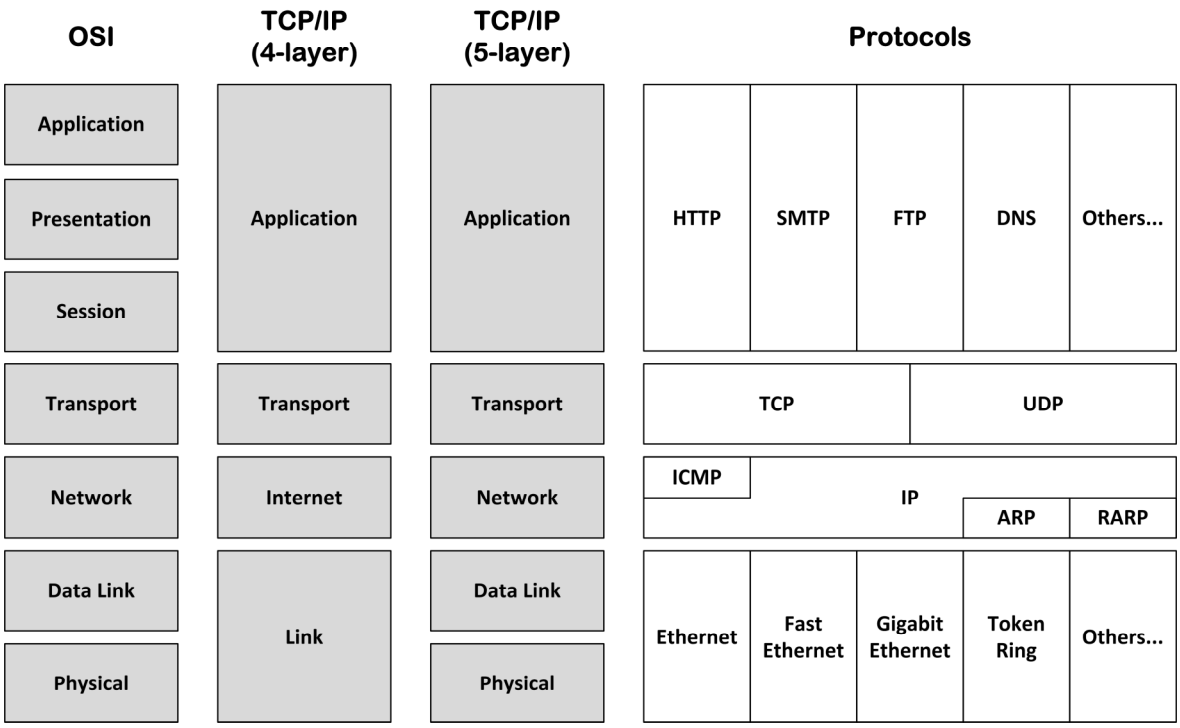


Figure 3.2 Predominant Internet protocols and their locations in the TCP/IP and OSI models

The Internet and transport layers

This unit focuses on the middle two layers of the Internet architecture, the Internet layer and the transport layer. Let's briefly look at what these layers do before we go on to examine their protocols in the rest of the unit.

The Internet layer is responsible for data transmission from the source host to the destination host (also known as **host-to-host delivery** or **end-to-end delivery**). The two communicating hosts may or may not reside on the same network. In fact, the Internet layer is intended to make their physical locations transparent to the upper layers. This means that as long as two hosts know each other's logical addresses, they can communicate with each other. As such, every host on the Internet must have a unique Internet address (IP address) so that it can be uniquely located.

The Internet Protocol (IP) is the predominant protocol for the Internet layer in the Internet architecture. IP functions such as naming, address administration and packet routing are explained in this unit. Then we'll discuss supporting protocols such as **Address Resolution Protocol** (**ARP**), **Reverse Address Resolution Protocol** (**RARP**) and **Internet Control Message Protocol** (**ICMP**).

The transport layer is responsible for **process-to-process delivery** of the entire data message. A process is an application program, such as a Web server program, running on a host. The transport layer ensures that the entire data message is delivered to the appropriate process on the destination host. This means that the transport layer must know how to address the appropriate process at the destination host, perform necessary fragmentation of outgoing messages before transmission and their reassembly after receiving them at the destination host, and deliver the entire message up to the application layer for further processing.

In contrast with the OSI transport layer, the TCP/IP transport layer does not necessarily enforce reliable communication. The TCP/IP transport layer uses the **Transmission Control Protocol** (**TCP**) for reliable communication. In applications where reliable communication is unnecessary, however, the **User Datagram Protocol** (**UDP**) is used to provide better efficiency. This unit discusses both of these protocols in detail, and explains how they break up a message into packets or datagrams and reassemble them at the other end.

The roles of IP, TCP and UDP

IP, TCP and UDP are the fundamental protocols for the whole Internet protocol suite. These three protocols can be used to communicate across any set of interconnected networks. They are equally well suited for local area networks (LANs) as they are for wide area networks (WANs).

The responsibility of the IP is to provide services in the network layer. TCP or UDP send data blocks to IP and tell IP the Internet address of the computer at the other end. Afterwards, it is the duty of IP to deliver the data to the receiving ends. You can imagine the operation of the IP as similar to sending a letter through the post office. When you send a letter, you have to write only the destination address; the rest of the delivery task is carried out by the post office. The IP performs a similar task to the post office — the TCP or UDP 'address' the message and the IP 'delivers' it.

Both TCP and UDP are responsible for breaking up the message into packets or datagrams, and reassembling them at the other end. TCP is a more intelligent protocol, however. It can resend any data that gets lost, and put packets back in the right order. In a real situation, a single station on a network can perform different tasks such as sending email and retrieving files from remote sites via the **File Transfer Protocol** (**FTP**). Hence, one of the functions of the transport layer protocols is to provide

multiplexing communication abilities. Details of TCP and UDP are discussed in later sections of the unit.

You may have noticed in Figure 3.2 that the structure of the Internet protocol suite is very interesting. In the application layer, we have a variety of application protocols such as SMTP, HTTP, DNS, FTP and so on. Beneath the application layer, the transport layer consists of the TCP and UDP protocols which support the application layer protocols. The link layer supports different network technologies such as Ethernet, Token Ring and FDDI. The Internet layer, however, is dominated solely by the IP. The importance of the IP in the TCP/IP protocol suite means that we must explore it first.

The Internet protocol

The Internet protocol is the Internet layer protocol that controls the routes of data across an internetwork. In addition to the internetwork routing function, IP provides fragmentation and reassembly of datagrams, and error reporting. Fragmentation is the process of dividing a large datagram into several smaller datagrams. In a large network, an originating host may not know all of the size limits that a datagram will come across along its path to its destination. So a datagram that is too large for some intermediate devices (routers) will be fragmented by the intermediate devices. It is then up to the destination host to gather up incoming fragments and to reassemble the original datagram.

The current version of IP in widespread use is IP version 4 (IPv4), which is described in RFC 791 (1981). The next generation of IP, Internet Protocol version 6 (IPv6), is specified by RFC 2460 (1998). We will briefly describe IPv6 later in this unit. For now, let's start with the IPv4 packet format.

The IPv4 packet format

The IPv4 packet format is shown in Figure 3.3 below.

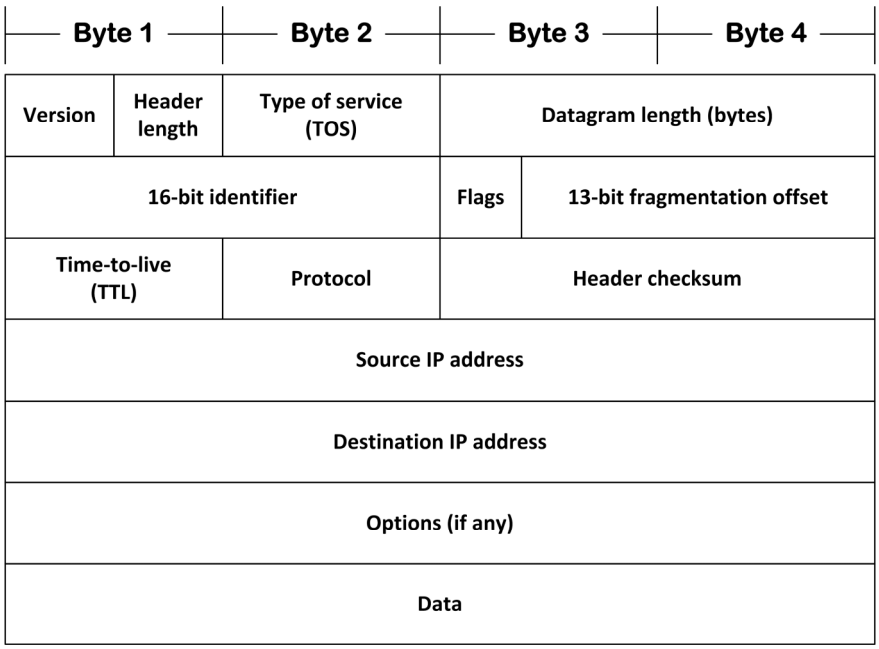


Figure 3.3 The IPv4 packet format

The fields of the IP packet are as follows:

- *Version* — indicates the version of IP currently used. The current version of IP is version 4, or IPv4 in short form.
- *Header length* — indicates the datagram header length in 32-bit words. If options are set, the header length is 20 **octets** (bytes).

- *Type of service* (*TOS*) — specifies how a particular upper layer protocol would like the current datagram to be handled. The TOS for different applications may be different.
- *Datagram length* — the length of the entire IP packet measured in octets, including data and header. The maximum packet length is 64K octets.
- *Identifier, flags, fragmentation offset* — indicate the packets' sequence in fragmentation. Receiving hosts reassemble the fragmented packets to the sending sequence according to the information in this field.
- *Time-to-live* (*TTL*) — indicates the maximum number of seconds/hops that a datagram will be allowed to remain in the network. The TTL is a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.
- *Protocol* — identifies which upper layer protocol receives incoming packets after IP processing is completed.
- *Header checksum* — a 16-bit field that helps to ensure IP header integrity.
- *Source IP address* — specifies the sending IP address.
- *Destination IP address* — specifies the receiving IP address.
- *Options* — provide additional information on features of the IP header. Options such as dedicated routing, security or time stamp operation can be available.
- *Data* — contains upper layer information, including TCP or UDP data.

The following Wikipedia article gives you a very good, concise overview of the Internet Protocol. After reading this Wikipedia article, read page 162–166 of Dean for more details on the fields in the IP header.

Reading 3.1

http://en.wikipedia.org/wiki/Internet_Protocol

Reading

Dean (2012) 148–52.

IPv4 addressing

You may have noticed in Figure 3.3 that, in IPv4, IP addresses are limited to a 32-bit length. This means the address space of the IPv4 is 2^{32} or 4,294,967,296. When IPv4 was standardized in the early 1980s, there were only about 200 hosts on the Internet (https://www.isc.org/solutions/survey/history). As such, 4,294,967,296 (i.e. over 4 billion) logical addresses were considered to be enough. But the dramatic growth of the Internet was beyond everyone's expectation. The 32-bit IPv4 address space is expected to be used up in a few years. This limitation has helped stimulate the push towards IPv6, which is currently in the early stages of deployment.

Address notations

At its simplest, the IPv4 address is just 32-bit binary number. For most people, a string of 32 zeros and ones is very difficult to remember, write, and verbally communicate. Consider the following 32-bit number in **binary notation**:

11001010001010001101110000000011

I bet most of you would find this string of 32 bits rather difficult to remember. In practice, IPv4 addresses are usually written in **dotted decimal notation**. To convert a 32-bit binary number into its corresponding dotted decimal notation, it is first split into four eight-bit **octets** (or bytes):

11001010 – 00101000 – 11011100 – 00000011

Each octet is then converted to a decimal number and the octets separated by a single period (dot):

202.40.220.3

Since each octet is a single eight bit octet, it can take on any value from 0 to 255. Thus, the lowest value of an IP address is theoretically 0.0.0.0, and the highest is 255.255.255.255. The following figure shows the decimal, binary and dotted decimal representations of the above IP address:

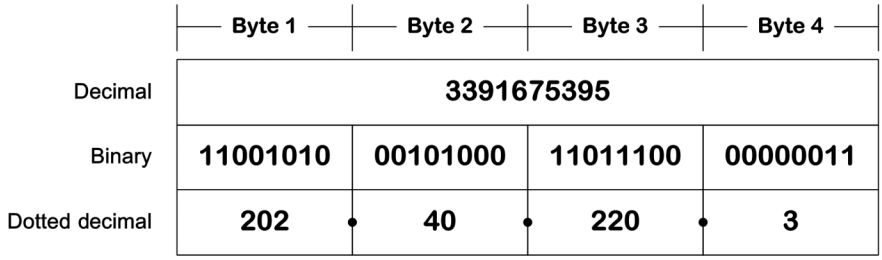


Figure 3.4 An IP address in decimal, binary and dotted decimal representations

Self-test 3.1

Convert the following numbers into their corresponding dotted decimal forms:

1 2130706433

2 11010010000100011001110010110011

3 167837953

Classful addressing

In the early 1980s, IPv4 addressing used the concept of classes. This addressing scheme, called **classful addressing**, is now becoming obsolete. Modern routers forward IP packets based on Classless InterDomain Routing (CIDR) (which we discuss in a later section) instead of classful addressing. However, classful addressing was historically important, and the address class terminology is still in widespread use today. Hence, we briefly discuss it here first to show the rationale behind it.

In classful addressing, the address space is divided into five different network classes: A, B, C, D and E. The network classes are designed to apply for different scales of network. The far left 4-bits indicate the network class. The patterns are shown in the following table:

Table 3.2 IP address class bit patterns, first-octet ranges and address ranges

Address class	First octet of IP address	Range of first octet values (decimal)	Theoretical IP address range
Class A	0xxx xxxx	0–127	0.0.0.0 to 127.255.255.255
Class B	10xx xxxx	128–191	128.0.0.0 to 191.255.255.255
Class C	110x xxxx	192–223	192.0.0.0 to 223.255.255.255
Class D	1110 xxxx	224–239	224.0.0.0 to 239.255.255.255
Class E	1111 xxxx	240–255	240.0.0.0 to 255.255.255.255

You should note the following points about the classes:

- *Class A networks* use the left-most 8 bits for the network address field and start with a number between 0 and 127. The host address field has 24 bits. Because Class A networks can support a very large number of host addresses, they are mainly for use with a few very large networks.

- *Class B networks* use the left-most 16 bits for the network address field and start with a number between 128 and 191. The host address field has 16 bits. This network class provides a good compromise between network and host address space.

- *Class C networks* use the left-most 24 bits for the network address field and start with a number between 192 and 223. The host address field has 8 bits. Eight bits implies a maximum number of 254 hosts in a Class C network. This class is the most popular and suitable for small-scale networks.

- *Class D networks* start with a number between 224 and 239. Class D addresses are reserved for multicast application. A multicast is a restricted form of broadcast in which data sent from a host with a Class D IP address is delivered to a group of systems that belong to the same multicast group. One of the major advantages of using multicasting is to reduce the redundancy of sending the same information from an application host to multiple receivers. To consider a video broadcast application with a server and two users (assuming this application does not support the multicast feature), the server has to send data separately to user1 and user2. This greatly affects network performance as the number of users increases.

  The situation is totally different if the application supports the multicast feature. When all users belong to the same multicast group, the server needs to send the data only once for all users. The server is not required to send the data one by one to each recipient. Obviously, network loading increases for this type of network application.

- *Class E networks* are reserved for experimental use. Class E addresses start with a number between 240 and 255.

Example 3.1

Determine the class of each address:

1 18.72.0.3

The first octet, 18, is equivalent to 00010010 in binary. Since the first bit of this octet is 0, the address is in class A. Alternatively, the first octet, 18, is within the range 0–127, so the address is in class A.

2 160.210.1.10

The first octet, 160, is equivalent to 10100000 in binary. Since the first two bits of this octet are 10, the address is in class B. Alternatively, the first octet, 160, is within the range 128–191, so this address is in class B.

3 192.180.5.1

The first octet, 192, is equivalent to 11000000 in binary. Since the first three bits of this octet are 110, the address is in class C. Alternatively, the first octet, 192, is within the range 192–223, so this address is in class C.

4 230.2.3.4

The first octet, 230, is equivalent to 11100110 in binary. Since the first four bits of this octet are 1110, the address is in class D. Alternatively, the first octet, 230, is within the range 224–239, so this address is in class D.

5 250.1.2.3

The first octet, 250, is equivalent to 11111010 in binary. Since the first four bits of this octet are 1111, the address is in class E. Alternatively, the first octet, 250, is within the range 240–255, so this address is in class E.

Network address and host address

In classful addressing, an IP address in class A, B and C is divided into two components:

- **Network address** (also known as **network ID** or simply **netid**) — A certain number of bits, starting from the left-most bit of an IP address, identify the network on which the host is located. This is also called the **network prefix** or simply the **prefix**.
- **Host address** (also known as **host ID** or simply **hostid**) — The remaining bits in the IP address are used to identify the host on the network.

The number of bits that defines the network address varies, depending on the class of the IP address. In class A, the left-most 8 bits defines the network address and the remaining 24 bits define the host address. In class B, the left-most 16 bits define the network address and the remaining 16 bits define the host address. In class C, the left-most 24 bits define the network address and the remaining 8 bits define the host address. For example, an IP address 192.173.15.23 is a class C network. The network address is 192.173.15.0 and the host address 23. Similarly, an IP address 10.122.12.34 is a class A network. The network address is 10.0.0.0, and the host number 122.12.34.

The following figure illustrates how three addresses, each of class A, B and C, are divided into their respective network and host addresses:

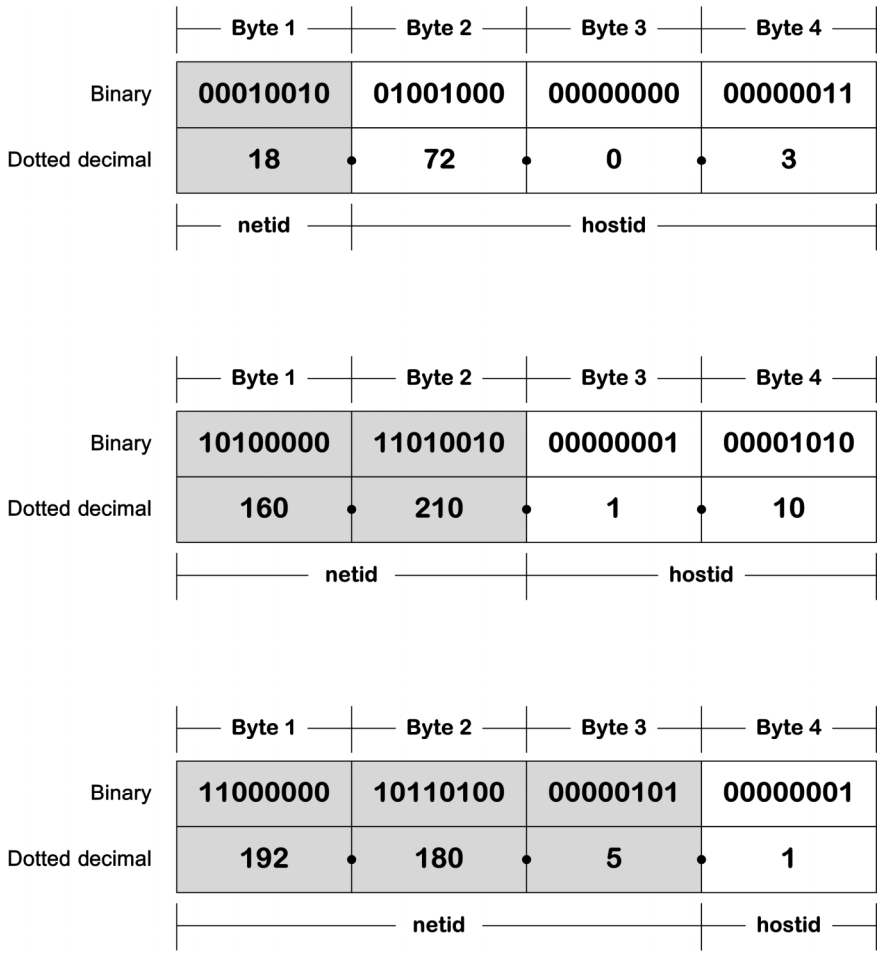


Figure 3.5 Basic IP address division: network address and host address

Splitting the IP address into a network address portion and a host address portion facilitates the routing of IP datagrams when the address is known. Routers look at the network portion of the destination IP address to determine how to route (select the most suitable route to delivery) the datagram to the destination.

We will explain IP routing in a later section in this unit.

Self-test 3.2

Determine the network address and host address for each of the following IP addresses:

1 61.10.128.100

2 128.133.5.254

3 202.40.220.3

Reserved, private and loopback addresses

There are generally three categories of addresses that are not available for normal address assignment:

- **Reserved addresses** are addresses set aside for future experimentation or for internal use in managing the Internet.
- **Private addresses** are addresses that can be used on a private network, but are not routable through the Internet. If a host on a private network needs to connect to the Internet, its IP address must be converted a public and routable address (real IP) by a mechanism called **Network Address Translation** (**NAT**). As such, many hosts on the private network can use the same real IP to connect to the Internet.
- **Loopback addresses** are addresses that a host can use to test itself without going into the network. They are used mostly for troubleshooting purposes. The loopback address is almost always cited as 127.0.0.1, although in fact, transmitting to any IP addresses whose first octet is 127 will always 'loop back' to the originating device.

The following table shows all of the reserved, private and loopback address blocks set aside from the normal IP address space in numerical order:

Table 3.3 Reserved, private and loopback IP addresses

Start address	End address	Address range in CIDR notation	Description
0.0.0.0	0.255.255.255	0.0.0.0/8	Reserved
10.0.0.0	10.255.255.255	10.0.0.0/8	Class A private address block
127.0.0.0	127.255.255.255	127.0.0.0/8	Loopback address block
128.0.0.0	128.0.255.255	128.0.0.0/16	Reserved
169.254.0.0	169.254.255.255	169.254.0.0/16	Class B private address block reserved for Automatic Private IP Address Allocation (APIPA)
172.16.0.0	172.31.255.255	172.16.0.0/12	Class B private address block
191.255.0.0	191.255.255.255	191.255.0.0/16	Reserved
192.0.0.0	192.0.0.255	192.0.0.0/24	Reserved
192.168.0.0	192.168.255.255	192.168.0.0/16	Class C private address block
223.255.255.0	223.255.255.255	223.255.255.0/24	Reserved

The third column in Table 3.3 shows the address blocks in CIDR notation. We discuss the CIDR notation in a later section in this unit.

IP addressing guidelines

For the assignment of a network address, some guidelines must be followed in order to select the correct IP address. The following criteria are important for network addressing. You should always bear these rules in mind:

- The IP address of each host must be unique. Duplicated IP addresses on different hosts will cause the corresponding hosts to be unable to communicate.
- Each octet must contain a value from 0 to 255, inclusive.
- The network address cannot begin with the number 127. The number 127 in Class A is reserved for internal loopback operations.
- The first octet in a network address cannot be 255. The octet 255 is used for the network's broadcast mode.
- The first octet in a network address cannot be 0. The 0 is used to indicate that the address is the local network.
- The host address cannot be all 1s because such an address is reserved for broadcasting to all hosts on the local network. An IP address with all 1s in the host portion is called a **broadcast address**. A broadcast address cannot be assigned to a host.
- The host address cannot be all 0s because such an address is the address of the network that the host belongs to.

The following reading gives you more information about IP addressing. It presents a clear explanation of IP addressing and examples of network classes.

> ***Reading***
>
> Dean (2012) 153–58.

Self-test 3.3

1 Identify which of the following IP addresses cannot be assigned to a host. Identify the IP addresses that would be invalid if they were assigned to a host. Explain why they are invalid.

131.107.256.80

222.222.255.222

231.200.1.1

0.127.4.100

127.1.1.1

2 What are the numbers of valid addresses that can be assigned to hosts in class A, B and C networks respectively?

Subnetting

An organization with a Class A or Class B network can support a large number of hosts in the same network. It does not make sense, however, to attach several thousand hosts in a single network address. Broadcast messages generating from thousands of hosts in the network may degrade the network performance to an unacceptable level. In addition to this performance problem, the use of the IP address is a concern for Class A and Class B. It's therefore reasonable to divide Class A, B or even C network addresses into multiple smaller subnetwork addresses. This addressing procedure, called **subnet addressing** or **subnetting**, is defined in RFC 950 (1985).

Subnet addressing adds an additional hierarchical level to the way IP addresses are interpreted — instead of just hosts, the network has subnets and hosts. Each subnet is a subnetwork, and functions much the way a full network does in conventional classful addressing. A three-level hierarchy is thus created — networks, which contain subnets, each of which then has a number of hosts.

The format of a subnetted IP address is shown in the following figure.

Figure 3.6 A subnetted IP address

The first part of the subnetted address designates the network address, the second part the subnet address, and the final part the host address. Subnets provide extra flexibility for network administrators.

For example, let's assume that an organization has been assigned a network 135.15.0.0 Class B. This implies that the network of the organization can support at most $2^{16} - 2 = 65{,}534$ hosts. However, the network administrator of the organization can use a **subnet mask** to subdivide the 135.15.0.0 into a group of subnetworks. This is done by borrowing bits from the host portion of the address and using them as a subnet field.

Suppose the network administrator assigns 8 bits of **subnetting**; the third octet of a class B IP address provides the subnet number. We can consider 135.15.0.0 to be the subnet 1 of the network; 135.15.1.0 to be the subnet 2; 135.15.2.0 to be the subnet 3; and so on. Logically, the network administrator can now increase the number of network addresses to more than 200 networks for the organization. This approach seems to give the administrator more flexibility to design the IP network with different segments.

The number of bits borrowed for the subnet address is variable, so it is possible to determine the number of subnets by choosing different subnet masks. Subnet masks make use of the format and representation of IP addresses. Subnet masks have 1s in all bits except those that specify the host field.

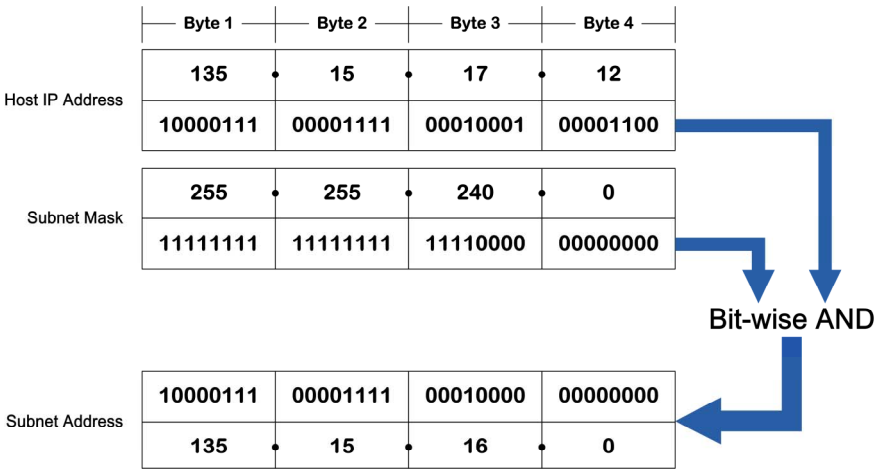
Let's think back to our example. The subnet mask is 255.255.255.0 for the Class B that specifies 8 bits of subnetting. The subnet mask that specifies 8 bits of subnetting for class B address 135.15.0.0 is 255.255.255.0. Hence, the network address of IP addresses 135.15.1.15 and 135.15.1.35 are both 135.15.1.0. We use the following example to illustrate the variable length of subnet masks.

Example 3.2

A host has the IP address 135.15.17.12. Given the subnet mask 255.255.240.0, find the corresponding network, subnet and host addresses.

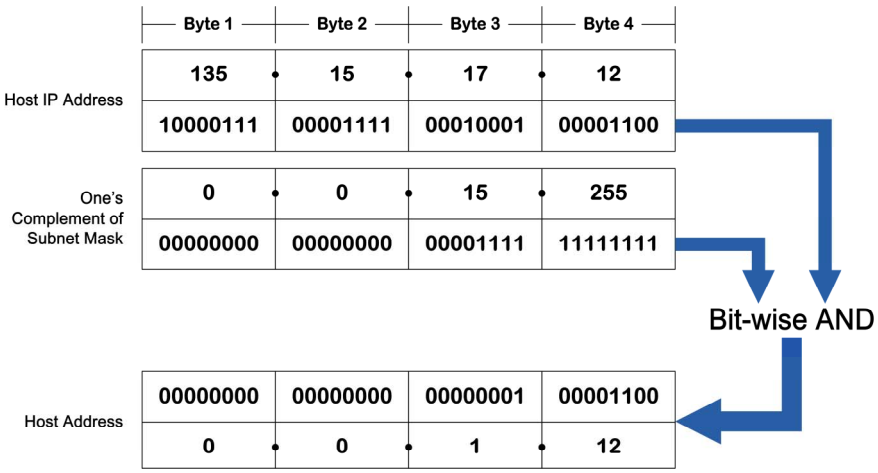
The address 135.15.17.12 is a class B address because the first octet is within the range 128–191. Hence the network address is 135.15.0.0.

To get the subnet address, we do a bit-wise AND of the IP address and the subnet mask:



The result is the subnet address, which is 135.15.16.0.

To get the host address, we do a bit-wise AND of the IP address and the 1's complement of the subnet mask:

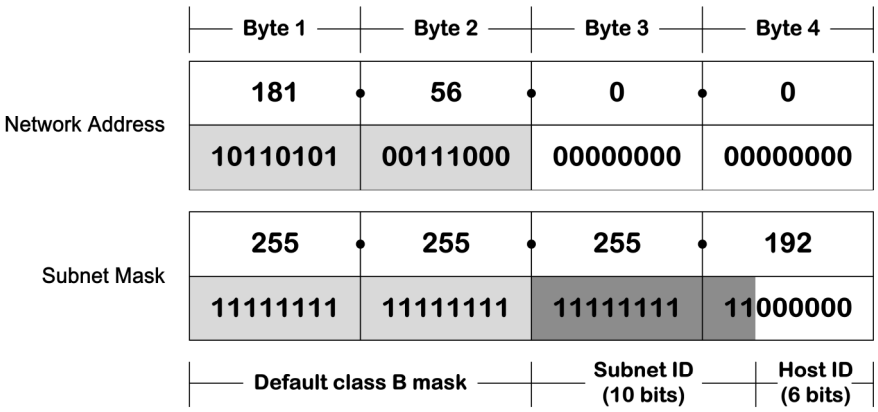


The result is 0.0.1.12. However, by convention, octets of 0s are omitted in the representation of a host address. Thus, the resulting host address is therefore simply written as 1.12.

Example 3.3

Your company is granted a class B network with the network address 181.56.0.0. Find the subnet mask and the range of addresses in the first subnet if you need to create 1,000 subnets for your company.

To accomplish this, we need to borrow 10 bits from the host portion of the address, since $2^9 < 1{,}000 < 2^{10}$. The default mask for class B has 16 ones, and hence the required subnet mask will have 16 + 10 = 26 ones. The resulting subnet mask is 255.255.255.192 in dotted decimal, as illustrated in the following diagram:



Since we have only 6 bits remaining for the host address portion, each subnet can have $2^6 = 64$ addresses (but only 62 of which can be assigned to hosts since the first and last one cannot be assigned). Consequently, the first subnet starts at 181.56.0.0 and ends at 181.56.0.63.

The benefits of subnetting

Using the subnet technique, network administrators can divide a network into multiple subnetworks and connect subnetworks with routers. This:

- reduces network congestion by redirecting traffic and reducing broadcasts; it can significantly improve the performance of Ethernet network;
- makes better use of the IP address; and
- makes it easier to control the network segment by dividing hosts into different subnetworks.

The next reading from Dean explains IP subnetting. The correlation between an IP address and its subnet mask is clearly illustrated.

Reading

Dean (2012) 399–410.

Self-test 3.4

1 Your company is granted a class B address 172.16.0.0.

   a By using a subnet mask of 255.255.255.0, how many subnets and how many addresses per subnets will there be? How many hosts can there be in each subnet?

b You have just assigned an interface address 172.16.10.50 with a subnet mask of 255.255.255.0. What subnet is it in? What is the host range in this subnet?

c You have a need for 2,046 subnets in your company. Based on your obtained class B address (172.16.0.0), recommend an appropriate subnet mask that you would assign. By using your assigned subnet mask, what subnet is in the following address: 172.16.10.170?

2 Another company has five departments: Personnel, Finance, Marketing, Research & Development (R&D) and Production. The company has requested a Class C IP address, with the network address 202.45.2.0. Each department has fewer than 25 hosts and connects to a centralized router. You have to suggest the network address and subnet masks for each department. What is the host range for each department?

Activity 3.1

There are many online IP calculators available on the Web. You can find relevant results by searching with keywords such as 'IP calculator', 'IP subnet calculator' or 'network calculator'. Use these tools to verify the results of Examples 3.2 and 3.3 using the software.

Classless Interdomain Routing (CIDR)

As the Internet began to grow dramatically in the early 1990s, it became apparent that the classful system of allocating IP addresses could be very wasteful: that is, anyone who could reasonably show a need for more than 254 host addresses was given a Class B address block of 65534 host addresses. Even more wasteful were companies and organizations that were allocated Class A address blocks, which contain over 16 million host addresses!

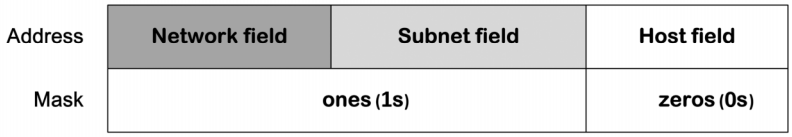
Classless InterDomain Routing (**CIDR**) is an addressing scheme that provides more efficient allocation of IP addresses than the old classful addressing scheme. CIDR abolishes the idea of traditional classes A, B and C networks and the notion of a subnet field. Subnets and network classes do not exist in a classless world: instead, there is only a network prefix and a host field.

Using CIDR, address blocks are no longer restricted to 16,777,216, 65,536 or 256 (corresponding to classes A, B and C address blocks respectively) in size. Instead, address blocks of various sizes (see Table 3.4) can be assigned to companies and organizations depending on the actual sizes needed. The companies and organizations that have already been granted class A and B address blocks can still hold on to theirs, but new companies and organizations are now assigned address blocks under

the CIDR address scheme. As such, the IPv4 address space can be used in a much more efficient way.

Figure 3.7 describes the difference between classful and classless addressing.

Classful addressing:



Classless addressing:

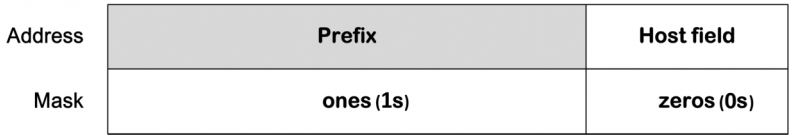


Figure 3.7 Classful versus classless addressing

The length of the network prefix is determined by a prefix mask, which is a contiguous series of 1s that starts with the left-most bit (the most significant bit). Although the prefix mask looks like a subnet mask, it's important to realize that there is no subnet field.

Since there are no address classes in CIDR, you cannot tell the size of the network ID of an address from the address alone. In CIDR, the length of the prefix (network ID) is indicated by placing it following a slash after the address. This is called **CIDR notation** or **slash notation**.

For example, consider a network specified by 163.54.192.0/22. The /22 following 163.54.192.0 means this network has 22 bits for the network ID and 10 bits for the host ID. This is equivalent to specifying a network with an address of 163.54.192.0 and a subnet mask of 255.255.252.0. This sample network allows a total of $2^{10} - 2 = 1{,}022$ hosts.

Since CIDR allows you to divide IP addresses into network IDs and host IDs along any bit boundary, it allows for the creation of dozens of different sizes of networks. The following table shows each of the possible theoretical ways to divide the 32 bits of an IPv4 address into network ID and host ID bits under CIDR:

Table 3.4 CIDR address blocks

Number of bits for network ID	Number of bits for host ID	Number of hosts per network	Prefix length in slash notation	Equivalent subnet mask
1	31	2,147,483,646 ($2^{31} - 2$)	/1	128.0.0.0
2	30	1,073,741,822 ($2^{30} - 2$)	/2	192.0.0.0
3	29	536,870,910 ($2^{29} - 2$)	/3	224.0.0.0
4	28	268,435,454 ($2^{28} - 2$)	/4	240.0.0.0
5	27	134,217,726 ($2^{27} - 2$)	/5	248.0.0.0
6	26	67,108,862 ($2^{26} - 2$)	/6	252.0.0.0
7	25	33,554,430 ($2^{25} - 2$)	/7	254.0.0.0
8	24	16,777,214 ($2^{24} - 2$)	/8	255.0.0.0
9	23	8,388,606 ($2^{23} - 2$)	/9	255.128.0.0
10	22	4,194,302 ($2^{22} - 2$)	/10	255.192.0.0
11	21	2,097,150 ($2^{21} - 2$)	/11	255.224.0.0
12	20	1,048,574 ($2^{20} - 2$)	/12	255.240.0.0
13	19	524,286 ($2^{19} - 2$)	/13	255.248.0.0
14	18	262,142 ($2^{18} - 2$)	/14	255.252.0.0
15	17	131,070 ($2^{17} - 2$)	/15	255.254.0.0
16	16	65,534 ($2^{16} - 2$)	/16	255.255.0.0
17	15	32,766 ($2^{15} - 2$)	/17	255.255.128.0
18	14	16,382 ($2^{14} - 2$)	/18	255.255.192.0
19	13	8,190 ($2^{13} - 2$)	/19	255.255.224.0
20	12	4,094 ($2^{12} - 2$)	/20	255.255.240.0
21	11	2,046 ($2^{11} - 2$)	/21	255.255.248.0
22	10	1,022 ($2^{10} - 2$)	/22	255.255.252.0
23	9	510 ($2^{9} - 2$)	/23	255.255.254.0
24	8	254 ($2^{8} - 2$)	/24	255.255.255.0
25	7	126 ($2^{7} - 2$)	/25	255.255.255.128
26	6	62 ($2^{6} - 2$)	/26	255.255.255.192
27	5	30 ($2^{5} - 2$)	/27	255.255.255.224
28	4	14 ($2^{4} - 2$)	/28	255.255.255.240
29	3	6 ($2^{3} - 2$)	/29	255.255.255.248
30	2	2 ($2^{2} - 2$)	/30	255.255.255.252
31	1	0 ($2^{1} - 2$)	/31	255.255.255.254

In CIDR-based networks, IP addresses are aggregated into CIDR blocks, which are identified by their respective network prefixes. All IP addresses in the same CIDR block belong to the same network, and they all have the same network prefix. A CIDR block has the following characteristics:

- The addresses in a block must be contiguous.
- All addresses in a block have the same network prefix.
- The number of addresses in a block must be a power of 2.
- The first address must be evenly divisible by the number of addresses in the block.

An advantage of classless addressing is the capability to combine what multiple originally class C addresses into a **supernet**. For example, if you need about 1,000 host addresses, you can supernet four class C networks together:

`192.60.128.0`	Class C network address
`192.60.129.0`	Class C network address
`192.60.130.0`	Class C network address
`192.60.131.0`	Class C network address
`192.60.128.0`	Supernetted network address
`255.255.252.0`	Subnet mask
`192.60.131.255`	Broadcast address

In this example, the supernetted network 192.60.128.0 includes all addresses from 192.60.128.0 to 192.60.131.255. Using the slash notation, the network address can be written simply as 192.60.128.0/22.

Reading

Dean (2012) 408–9.

The next reading describes CIDR in more detail and explains why it is needed. In addition to examining the restructuring of IP address assignments, the reading describes how CIDR enables ‘route aggregation’ to minimize routing table entries. Finally, it highlights the user effects of the CIDR addressing scheme and route aggregation.

Reading 3.2

'What is CIDR',
http://www.cryer.co.uk/glossary/c/cidr/what_is_cidr.htm

Self-test 3.5

1 Determine the address range covered by the network: 192.168.4.0/22.

2 For each of the following addresses, determine whether it is a network address, host address or broadcast address:

10.0.8.0/22

172.17.16.255/23

192.168.37.192/25.

3 How does CIDR overcome the limitations of classful addressing?

Physical and logical address resolution: ARP and RARP

As you learned in the previous units, nodes on a network have both physical and logical addresses. The physical address is a hardware address — often called a **Media Access Control** (**MAC**) address. Each network interface has a built-in 48-bit MAC address to identify itself. This physical address is fixed, so you do not need to configure the setting of this address. Theoretically, we can have 2^{48} different physical addresses in the world without duplication, so we are still not facing the problem of running out of physical addresses for network interfaces. In contrast, the logical address depends on the networking protocol used for data transmission in the OSI network layer. A network administrator must manage logical addresses to ensure that every node on a network can communicate with other nodes, a process known as IP addressing.

In the Internet architecture, the physical address works at the link layer. The IP address works at the Internet layer, which provides a network address similar to the requirements of the network layer in the OSI model. In most applications, we address the destination host with its Internet layer address scheme, which is the IP address in the TCP/IP protocol suite. The transmission of packets in Ethernet is in broadcast mode. The broadcast mode implies that, whenever a packet is sent from a host, the packet will be passing through the whole network of the particular host. Each host in the network has to keep an eye on the network media in order to extract data sent to it. Whenever a host detects

a packet that is sent to its address, the host absorbs the packet and passes it to the upper layers.

You may ask how a network interface recognizes that the packet is addressed to it. The answer is that the network interface checks the destination address in the network access layer, which is the physical address, to determine whether to absorb the packet or not.

The mapping between the physical address and the IP address is implemented by the **Address Resolution Protocol** (**ARP**) and **Reverse Address Resolution Protocol** (**RARP**), so there is no problem for a user to send data to the receiving host via its IP address. ARP can resolve the corresponding physical address of the receiving host. The resolving process can be done the other way round: hosts with physical addresses can perform RARP to determine the corresponding IP addresses.

Mapping logical to physical address: ARP

ARP is described in one of the earliest of the Internet RFCs still in common use: RFC 826 (1982). ARP uses broadcast messages to determine the physical address of a particular IP address. The system on the local network applies ARP to automatically look up the physical addresses.

The following figure shows how ARP operates between hosts on the same network:

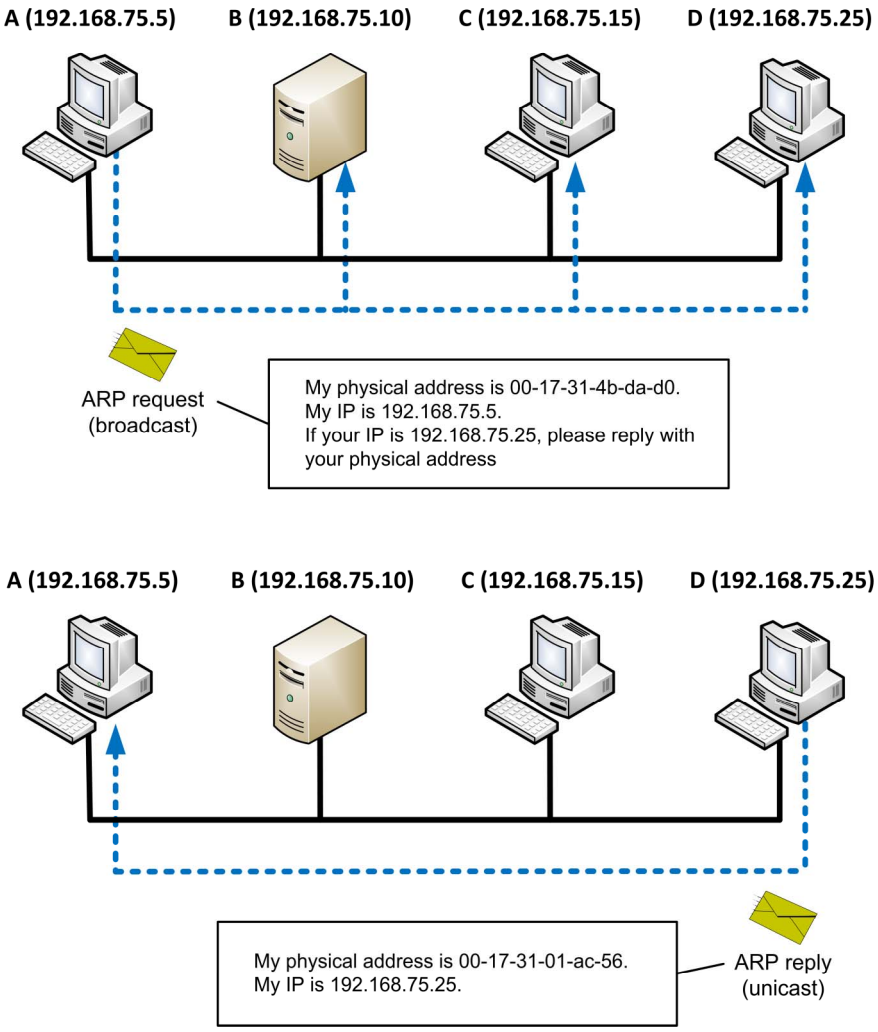


Figure 3.8 Operation of ARP between hosts on the same network

As Figure 3.8 shows, a user at host A with the IP address 192.168.75.5 wants to send data to the receiving host D with the IP address 192.168.75.25. However, other hosts are attached to the network. In order to identify the physical address of host D, host A sends a broadcast message to discover the physical address of host D. When host D receives the ARP broadcast message and knows that host A is looking for its physical address, host B sends its physical address information to host A via a unicast reply message. After receiving the physical address of host D, host A updates its ARP cache. The ARP cache maintains the recent mapping from IP addresses to physical addresses.

The physical address of host D is recorded in host A's ARP cache for further use. You can use the command **`arp -a`** to examine the ARP cache of host A:

```
host B (192.168.75.25)      at 00:17:31:01:ac:56
```

The host names, IP addresses and the corresponding physical addresses are stored in the cache. Similarly, as host A communicates with host C, the ARP cache will store another entry for host C.

Mapping physical to logical address: RARP

The reverse process of ARP, Reverse Address Resolution Protocol, is known as Reverse ARP or RARP. RARP was formalized in RFC 903 in 1984. A station uses RARP to broadcast messages to determine the Internet address associated with its own hardware address. A server on the network is dedicated to handling RARP queries and to respond to the sending station. The services of RARP are particularly important to diskless systems. A diskless system has no local disk resources so it may not know its internetwork address when the system is up. A diskless system then must initially use RARP to obtain its IP address. The mechanism of RARP is shown in the following figure:

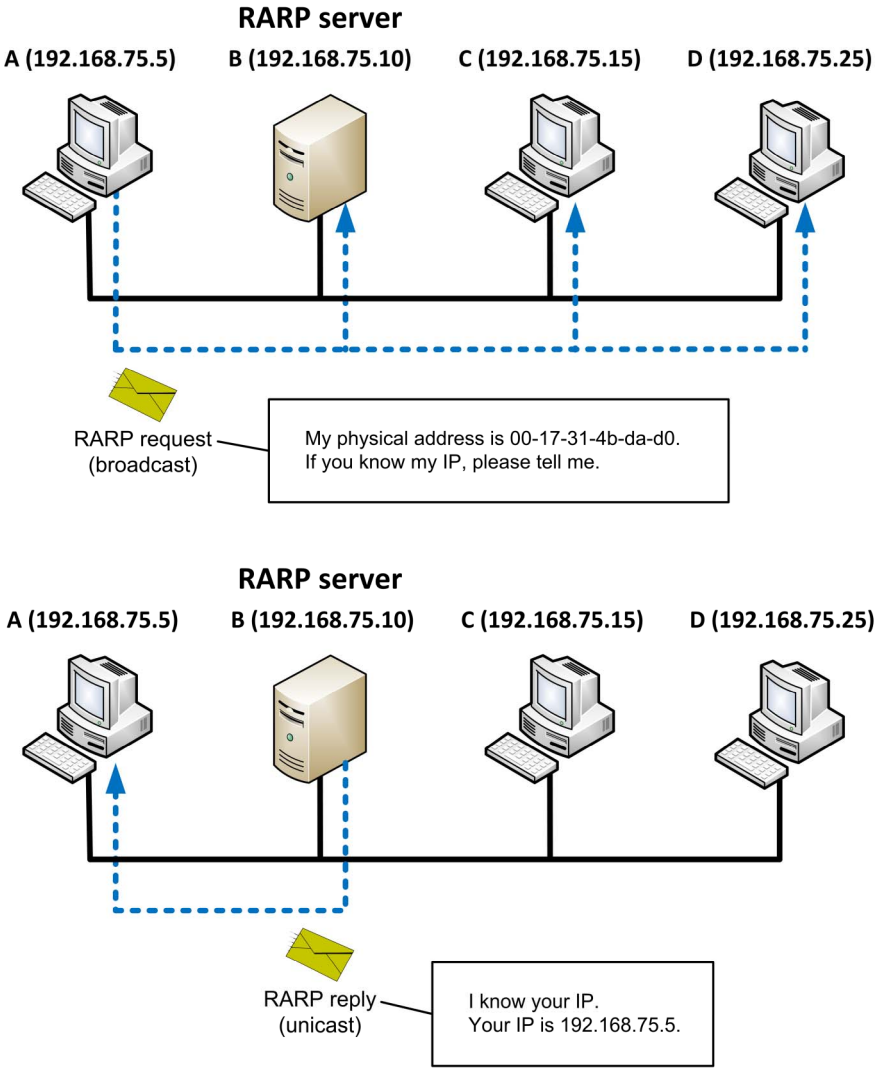


Figure 3.9 Operation of RARP

The packet format of RARP is similar to that of ARP. The field of frame type in the packet indicates whether it is a RARP request or reply. The transfer modes of the RARP request and reply are different, however. The RARP request is broadcast and the RARP reply is normally unicast. The RARP reply is issued by the RARP server, which maintains a list of available IP addresses. The assignment of the IP address to the request

host can be fixed or dynamic; it depends on the implementation of the server. RARP is one of the protocols for IP assignment; other protocols such as the Bootstrap Protocol (BOOTP) and Dynamic Host Configuration Protocol (DHCP) provide similar services.

Self-test 3.6

Assume that you are working in a TCP/IP network with five hosts interconnected (you can name these H1 to H5). How can you determine the corresponding hardware address for each host?

Address administration: BOOTP and DHCP

As you learned in the previous section, a host without an IP address can determine its own IP address by using RARP. However, a major limitation of RARP is that routers cannot forward RARP requests. Network administrators therefore can't design RARP clients in different network segments with a single RARP server.

In this section, we discuss two other methods that are used to overcome the limitations of RARP:

- the **Bootstrap Protocol (BOOTP)**, and
- the **Dynamic Host Configuration Protocol (DHCP)**.

The Bootstrap Protocol (BOOTP)

The Bootstrap Protocol (BOOTP) was developed in the mid-1980s. It uses a central list of IP addresses and their associated device's MAC addresses to dynamically assign IP addresses to clients. When a client that relies on BOOTP first connects to the network, it sends a broadcast message to the network asking to be assigned an IP address. This broadcast message includes the MAC address of the client's NIC. The BOOTP server recognizes a BOOTP client's request, looks up the client's MAC address in its BOOTP tables, and responds to the client with information of the client's IP address, server's IP address, server's host name and the IP address of a default router.

The Dynamic Host Configuration Protocol (DHCP)

Unlike BOOTP, the Dynamic Host Configuration Protocol provides a framework for passing configuration information to hosts on TCP/IP networks. DHCP servers deliver host-specific configuration parameters — including the IP address, subnet mask and Domain Name Server (DNS) address — to the request hosts. DHCP adds the capability of

automatic allocation of reusable network addresses and additional configuration options.

The next reading outlines the process of assigning IP addresses by using BOOTP and DHCP. DHCP was developed by the IETF as a replacement for BOOTP. You should note the advantages of DHCP compared with BOOTP. This reading also explains IP address leasing and terminating processes.

Reading

Dean (2012) 159–64.

Reading 3.3 (optional)

http://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

Self-test 3.7

Describe the primary differences between DHCP and BOOTP.

Domain name system (DNS)

Most people find it is easier to remember the name of a host than to remember its IP address. Most people also find it easier to remember a hostname such as 'www.ouhk.edu.hk' than a 32-bit number such as 3391675395, or an IP address such as 11001010.00101000.11011100.00000011 or 202.40.220.3.

Since this is the case, we need a host naming convention to assign names to hosts in a meaningful and organized way. In addition, we also need an efficient and reliable system to map hostnames to IP addresses and vice versa (similar to a white-pages phonebook that maps names to telephone numbers).

Domain names

The naming scheme in the Internet is based on the concept of domains. Host names within a domain are delegated by the domain administrator. A domain can be divided into different subdomains. Each subdomain can consist of hundreds or thousands of hosts in the naming system.

The **Internet Corporation for Assigned Names and Numbers (ICANN)** is a non-profit corporation that was created in 1998 to manage a number of Internet-related tasks, including the assignment of domain names and IP addresses. Nowadays you can register a domain name via a **domain name registrar** that is accredited by the ICANN. The domain naming system is structured in a hierarchical model. There are several common naming conventions:

- *com* — commercial organizations
- *edu* — educational institutions
- *gov* — governments
- *mil* — military organizations
- *net* — network services (such as Internet service providers)
- *org* — non-profit organizations
- *hk, cn, uk, jp* and so on — top-level domains based on countries.

The structure of host names may consist of two to five labels. For example, the host name 'www.ouhk.edu.hk' consists of the name of the host, 'www', and the domain name, 'ouhk.edu.hk'. The domain name is relative. 'ouhk.edu.hk' is a subdomain name of 'edu.hk' and 'edu.hk' is a subdomain of 'hk'. 'hk' is the root of the name structure.

Reading 3.4 (optional)

The following website provides a list of frequently asked questions about the domain name registration process and the management of the Internet domain system. If you would like to learn more about the naming scheme in the Internet, you can visit this site:

http://www.internic.net/faqs/domain-names.html

Activity 3.2

Go to the following site:

http://www.register.com/.

Then type in your first and last name (e.g. www.andylau.com if your name is Andy Lau) and select your desired extension. Is your name available as a domain name?

Domain name system

A straightforward way to implement host naming to IP address mapping is to create a file listing hostnames and their corresponding IP addresses line by line. The file is called **`hosts`** under the directory '**`/etc`**' in the UNIX system. But since it is impossible to include all hostnames in this file, the Domain Name Server is used by TCP/IP applications to map between hostnames and IP addresses. The network administrator defines the hostname. Hence, it is possible to assign a meaningful hostname to the corresponding hosts. They help users and administrators locate network resources more easily.

The DNS is a distributed database holding the alphanumeric names and IP addresses of every registered system on the Internet. The databases are created by the network administrators. Configuration files in local systems store the mapping between hosts and IP addresses.

The **daemon** process, which you may find interesting, is a background process running on an operating system that generates the database according to the configuration files. The DNS works as a client-server model. If a client would like to make communication with a host called 'www.ouhk.edu.hk', the client requires the corresponding IP address of 'www.ouhk.edu.hk'. The Internet layer works with the IP address only, and not the hostname address, so the client needs to send a request to a predefined DNS server in order to look up the IP address of the 'www.ouhk.edu.hk'. The IP address is then sent back to the client. If the database of the DNS server does not contain information about the targeted host, the query is redirected to next level DNS server, which contains other zone information. If this DNS server contains the required information, the query is completed and the IP address is sent back to the client. But if the result is again negative, the query may be further redirected to other DNS servers until it reaches the root DNS server.

Figure 4.14 on page 185 of Dean (2004) gives a very good illustration of the domain name resolution process.

Root name servers

A domain name is defined in a hierarchy model. At the top of the hierarchy is the root domain. Information on this domain resides on a selected number of root name servers around the Internet. The **root name servers** contain a database for all of the top-level domains.

For example, the '.com' DNS root server contains all subdomain servers relating to hostnames ending with '.com'. These root name servers make it possible for every host on the Internet to have access to the complete DNS database.

Authoritative name servers

Authoritative name servers store information for the hostnames of subdomains. Each authoritative name server holds in its database the

name-to-address mappings for the group of hosts it administers. Basically, each domain should have its own authoritative name server. The local hostnames are assigned by the network administrators. In order to let other DNSs refer to a particular domain, the local DNS of that domain has to be referred by the other DNSs. The authoritative name servers for every domain are officially registered with IANA.

The next reading provides a description of host names and domain names. It illustrates the hierarchy of Domain Name System (DNS). It also demonstrates the configuration of DNS on a Windows XP machine. Finally, it discusses how name servers keep track of IP addresses and their associated names. In this reading, you should note the purpose of DNS and host file and try to understand the hierarchical nature of DNS.

Reading

Dean (2012) 166–73.

Reading 3.5 (optional)

http://en.wikipedia.org/wiki/Domain_name_system

The following activity introduces a very useful DNS utility program, '**`nslookup`**', which is available on both Windows and Unix systems. **`nslookup`** is an interactive program for searching Domain Name Servers. The user can contact DNS servers to request information about a specific host, or print a list of hosts in the domain. Besides the basic IP and hostname information, you can use this command to search the DNS server, look for aliases for a single IP, and so on. On Windows, you need to open the command prompt window and enter '**`nslookup`**' to launch the program. On UNIX, you just enter the command at the console prompt. You will need to refer to the help manual of the '**`nslookup`**' command in order to answer the following questions.

Activity 3.3

1 What is the IP address of the 'www.ouhk.edu.hk' host?

2 What is the hostname of the IP address, '207.68.137.59'?

3 What is/are the DNS server(s) for the 'ouhk.edu.hk' domain?

4 How do you find out the root name server in your working domain?

The Internet Protocol version 6 (IPv6)

The **Internet Protocol Version 6** (**IPv6**) is the next generation of the Internet Protocol and is now included as part of IP support in many products, including the major computer operating systems. IPv6 has also been called IP Next Generation or 'IPng'. It is defined in RFC 2460 (1998). IPv6 was designed to improve the existing IP version 4. To ensure compatibility, network hosts and intermediate nodes with either IPv4 or IPv6 can handle packets formatted for both levels of the Internet Protocol.

The development of IPv6 has several purposes. To solve the problem of a shortage of 32-bit IP addresses, a significant improvement in IPv6 over the IPv4 is that IP addresses are lengthened to 128 bits. This extension of the IP addresses ensures a supply of addresses for the future growth of the Internet. IPv6 also provides several new features:

- To speed up overall network performance, options are specified in an extension to the header that is examined only at the destination.
- The anycast address allows messages to be sent to any of several possible service providers or to any of a related group of remote destinations, with the idea that any one of them can manage the forwarding of the packet to others.
- Real-time services are available for multimedia applications. The IPv6 header now includes extensions that allow a packet to specify a mechanism for authenticating its origin, and for ensuring data integrity and privacy.

Your next reading from Dean also addresses IPv6.

> ***Reading***
>
> Dean (2012) 158–59.

A good overview of IPv6 is given in the following Wikipedia article. You are not required to study all materials in this article but only focus on sections 1 to 4.

> ***Reading 3.6***
>
> http://en.wikipedia.org/wiki/IPv6

If want to learn more about IPv6 (and you should, because it is expected to replace IPv4 in a few years), you can start by exploring the following websites:

- http://www.ipv6.com/
- http://www.ipv6.org/
- http://www.ipv6forum.com/.

IP routing

When a computer wants to send a packet to a destination that is not on its local subnetwork, it sends that packet to its default gateway (usually a router). Then the default gateway forwards the packet to its true destination, or to another router that knows how to reach that destination. This process is called **routing**. In other words, we use the term **routing** for the process of taking a packet from one device and sending it through the network to another device on a *different* network.

In this section, we explain in detail the processes and the protocols that routers use to make forwarding decisions by using routing protocols (e.g. RIP and OSPF), and how they handle routing and delivery problems or failures by using ICMP.

Routing principles

In an internetwork system, different network segments are connected via a multitude of network devices, including hubs, bridges, switches and routers. Hubs, bridges and switches all work within a single network segment, but routers are able to support different segments.

Let's consider a simple example, illustrated below: A router with four interfaces connects to four networks — NET1, NET2, NET3 and NET4. A packet from a source host in segment NET1 can reach the destination host in segment NET2 because the router is able to route packets from NET1 to NET2. But which function lets routers know how to pass packets from NET1 to another network interface that connects to NET2?

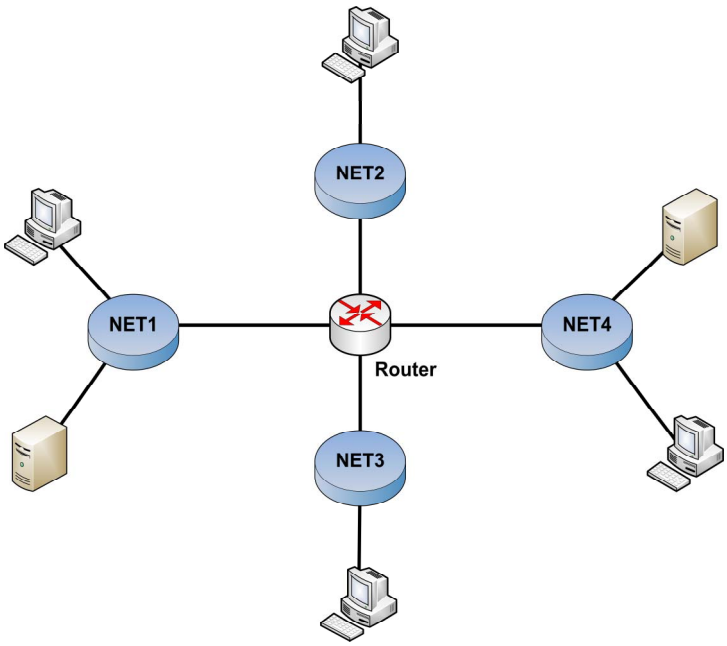


Figure 3.10 Example of a router connection

This example is actually far simpler than most real environments. In a corporate network with hundreds of routers, the situation becomes much more complicated.

In order to provide the routing information for data to different networks, many routing protocols have been developed to provide a partial or full network map between interconnected routers of networks. The main purpose of routing protocols is to construct **routing tables**, i.e., lists of routing entries, in routers. Each entry in a routing table consists of a remote or local network and the corresponding network interface that can get closer to the desired network. With the propagation of these entries to each router in the network, it is possible to establish the full connectivity of the network.

Routing tables

A minimal IP routing table consists of destination address/next hop pairs. A sample entry, shown in Table 3.5 below, is interpreted as meaning 'to get to network 135.125.0.0, the next stop is the node at address 173.21.23.1'. Similarly, the next hop for the network 201.4.12.0 is at the node of 173.21.23.1.

Table 3.5 An IP routing table

Network/host	Next hop
135.125.0.0	173.21.23.1
201.4.12.0	173.21.23.1
…	…
215.93.3.0	212.34.20.254
134.122.0.0	212.34.20.254
…	…

IP routing specifies that IP datagrams travel through internetworks one hop at a time. As the next hop may or may not be the final destination, each intermediate device needs to match the destination address of the datagram with an entry in the current node's routing table. Each node involved in the routing process tries its best to forward packets based on internal information, regardless of whether the packets get to their final destination. Whenever any error occurs in the forward packets, or a network is unreachable, it is not the node's responsibility to provide error reporting to the source.

Static and dynamic routing

In general, the two ways through which routers can learn routing information are **static routing** and **dynamic routing**.

In static routing, routing information is manually configured on the router, creating what is known as a **static route**. Static routing requires a network administrator for initial setup and for any subsequent changes to routes. Obviously, static routing does not work well except in the smallest and simplest networks such as home and small office networks, where the internetworking structures are simple and seldom change. In large networks, it would take too much administrative effort to manually configure and maintain routing tables in a manner that ensures efficient routing.

In contrast, dynamic routing, also known as **adaptive routing**, is the use of routing protocols to dynamically update routing tables. In dynamic routing, routers periodically exchange routing information and update each other according to a routing protocol in such a way that any change in the internetworking structure is efficiently propagated to all routers in the internetwork. These dynamic routing information updates occur automatically and require no administrative effort.

In the following sections we brief discuss two important dynamic routing protocols, the Routing Information Protocol (RIP) and Open Shortest Path First (OSPF).

The Routing Information Protocol (RIP)

The **Routing Information Protocol** (**RIP**) is a kind of **distance-vector routing** protocol, in which each router maintains a vector (table) of minimum distances to every other router in the internetwork. Each entry in an RIP routing table provides information including the ultimate destination, the next hop on the way to that destination, and a metric. The metric is equal to the distance in number of hops to the destination. Other information can also be presented in the routing table, including various timers associated with the route.

Table 3.6 An RIP routing table

Destination	Next hop	Metric	Timer
135.23.0.0	Router 1	3	11
198.23.43.0	Router 4	5	21
202.168.2.0	Router 4	3	15
212.15.23.0	Router 3	4	45

RIP dynamically adjusts the routing table onto the best route to a destination. When new information provides a better route, this information replaces old route information. The network links may vary from time to time. The network environment will be changed if a link is broken, a host is down, and so on. When network topology changes occur, they are reflected in routing update messages. Each router receiving a routing update message that includes a change updates its tables and propagates the change.

Like other routing protocols, RIP uses timers to improve its performance. The routing table in each router is updated regularly. Each router sends a complete copy of its routing table to all neighbors every 30 seconds. The route invalid timer determines how much time must expire without a router having heard about a particular route before that route is considered invalid. When the route flush timer expires, the route is removed from the routing table. Typical initial values for these timers are 90 seconds for the route invalid timer, and 270 seconds for the route flush timer. The removal of unused entries is necessary to keep the routing table records to a reasonable size. If the routing table grows without the removal of expired entries, the updating of routing traffic may significantly affect the network traffic.

Stability features

RIP specifies a number of features designed to make its operation more stable in the face of rapid network topology changes. These include:

- hop count limit; and
- split horizon.

Hop count limit

RIP has to limit the maximum hop count to 15. Any destination greater than 15 hops away is regarded as unreachable. The limitation of the hop count in RIP is able to ensure network routing table convergence in a shorter time, but this limits its application in large internetworks.

Split horizon

'Split horizon' is a strategy in which each router sends only part of its routing table through each interface, instead of flooding the table through each interface. It is based on the fact that it is never useful to send information about a route back in the direction from which it came. For example, consider the following figure:

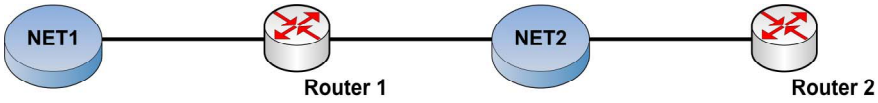


Figure 3.11 Split horizons

It is clear that NET1 is directly connected to Router 1 (R1). There is no reason for Router 2 (R2) to include this route in its update back to R1, because R1 is closer to NET1. The split-horizon rule says that R2 should delete this route from any updates it sends to R1. The split-horizon rule helps prevent two-node routing loops. For example, consider what happens if R1's interface to NET1 fails. Without split horizons, R2 keeps on informing R1 that it can get to NET1 through R1. If R1 does not have sufficient intelligence, it might actually pick up R2's route as an alternative to its failed direct connection, causing a routing loop.

Self-test 3.8

What information is required in a routing table?

Open Shortest Path First (OSPF)

Since a router with RIP will send routing information to its neighbor routers every 30 seconds, it may take a relatively long period to propagate its information across the whole network. The delay of the update in routing information may affect a router's ability to choose the right route for the data. The convergence of routing information implies stabilizing after something changes, such as a router or a link going down.

Open Shortest Path First (**OSPF**) is a newer technology than RIP, and it functions as an interior gateway protocol. It overcomes the major problem of RIP, which is slow convergence. Unlike the concept of hop count applied in RIP, OSPF is a **link state routing** protocol. In a link state routing protocol, a router does not exchange distances with its neighbors. Each router frequently checks the status of its links to each of its neighbors, sends this information to its other neighbors, which then propagate the link status throughout the autonomous system.

An 'autonomous system' in this context means a system in which a group of routers are under the control of an organization. A corporate network, for example, can be considered as an autonomous system. The link status of each connection between routers in the autonomous system is updated frequently and build a complete routing table.

In a network running the OSPF routing protocol, routers send 'Hello' messages to other routers to check if these neighbors are awake. A designated router in the autonomous system is selected to be responsible for updating its adjacent neighbors with the latest network topology information. When the status of a link is changed, the designated router is responsible for making sure that all routers in the network know the update status of the link.

Self-test 3.9

1 Why is RIP typically not used in large internetworks?

2 Explain the differences between RIP and OSPF.

The Internet Control Message Protocol

The Internet Control Message Protocol (ICMP) can be regarded as a network helper. It performs a number of tasks within an IP internetwork. ICMP can report routing failures to the source. In addition, ICMP provides helpful messages such as the following:

- testing node reachability across an internetwork with echo and reply
- stimulating more efficient routing with redirects
- 'time exceeded' messages to inform sources that a datagram has exceeded its allocated time to exist within the internetwork
- router advertisement and router solicitation messages to determine the addresses of routers on directly attached subnetworks.

ICMP error messages

ICMP error messages are useful services for network administrators. ICMP generates error messages to let administrators trace the causes of the problems. The following different situations may induce ICMP error messages:

- a datagram cannot reach its destination
- a packet's time-to-live expires
- an incorrect IP header
- router or destination host congestion.

The format of the ICMP message starts with the same three fields: a message type, a code of the type, and a checksum for message error recovery. In a 'destination unreachable' message, the message type is 3 to indicate the destination is not reachable. The causes of the destination being unreachable are various — it can be induced by a broken link, the failure of a router, an unknown destination network, and so on — so the code field provides a more specific description of the error type. For example, if a host is unreachable because it is unknown, the corresponding error type is 3 and the code is 7. This message helps the administrator isolate the connectivity problem.

In addition to the 'destination unreachable' message are other useful message types supported by ICMP. Some general types are listed for your reference below.

- ICMP echo reply type = 0
- Source Quench type = 4
- ICMP redirect type = 5
- ICMP echo request type = 8
- Time exceeded message type = 11

- IP parameter problem type = 12

Before we end this section, we want to familiarize you with the very useful '**`ping`**' command for testing network connectivity. '**`ping`**' commands are built on the ICMP echo message (i.e. Type 0 and 8). If you want to check whether a host is alive or not, you can issue a '**`ping`**' command to the desired host with different options.

Usage

```
ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS] [-r count]
     [-s count] [-w timeout] destination-list
```

Frequently used options

`-t`	Ping **`destination`** until interrupted
`-a`	Resolve addresses to hostnames
`-n count`	Number of echo requests to send
`-l size`	Send buffer size
`-f`	Set 'Don't fragment' flag in packet
`-i TTL`	Time to live
`-v TOS`	Type of service
`-r count`	Record route for counting hops
`-s count`	Timestamp for counting hops
`-w timeout`	Timeout in milliseconds to wait for each reply.

The following section in your textbook provides further information on the '**`ping`**' command.

Reading

Dean (2012) 176–78.

Self-test 3.10

1 What is the ICMP error type and code if the datagram has timed out because the TTL is equal to zero?

2 How do you send a sequence of ICMP echo messages to www.ouhk.edu.hk with 15 messages, each containing 128 octets?

The Transmission Control Protocol

The Internet transport layer is implemented by the **Transmission Control Protocol** (**TCP**) and the **User Datagram Protocol** (**UDP**). TCP provides connection-oriented data transport, whereas UDP is connectionless. The connection-oriented data transport of the TCP connection is a reliable application-to-application (process-to-process) transmission.

The facilities that TCP provides include:

- **Stream data transfer —** TCP transfers a continuous stream of bytes through the network. The application does not have to bother with chopping the data into basic blocks or datagrams. TCP does this by grouping the bytes into TCP segments, which are passed to the IP layer for transmission to the destination. Also, TCP itself decides how to segment the data, and it can forward the data at its own convenience.

- **Reliability —** TCP assigns a sequence number to each byte transmitted, and expects a positive acknowledgment (ACK) from the receiving TCP. If the ACK is not received within a timeout interval, the data is retransmitted. Because the data is transmitted in blocks (TCP segments), only the sequence number of the first data byte in the segment is sent to the destination host. The receiving TCP uses the sequence numbers to rearrange the segments when they arrive out of order, and to eliminate duplicate segments.

- **Flow control —** The receiving TCP, when sending an ACK back to the sender, also indicates to the sender the number of bytes it can receive (beyond the last received TCP segment) without causing overrun and overflow in its internal buffers. This is sent in the ACK in the form of the highest sequence number it can receive without problems. This mechanism is referred to as a window-mechanism.

- **Multiplexing** and **demultiplexing —** A typical host runs multiple applications that need to send data to and receive data from the network simultaneously. These applications are differentiated by their assigned port numbers. At the sending host, data packets from these applications are multiplexed to transmit through the single outgoing TCP. On the other hand, at the receiving host, TCP must be able to demultiplex the data packets such that they are delivered to the appropriate applications.

- **Logical connections —** The reliability and flow control mechanisms require that TCP initializes and maintains certain status information for each data stream. The combination of this status, including sockets, sequence numbers, and window sizes, is called a logical connection. Each connection is uniquely identified by the pair of sockets used by the sending and receiving processes.

- **Full duplex —** TCP provides for concurrent data streams in both directions.

We discuss these features in the following sections.

Ports and socket addresses

The IP address of a corresponding host is not sufficient to handle multiple connections and to address a connection for each application. Each application has to provide its TCP or UDP port number, with the host address, to specify a particular connection. The range of port numbers is from 0 to $2^{16} - 1 = 65535$.

The **Internet Assigned Numbers Authority** (**IANA**) is responsible for maintaining the official assignments of port numbers for specific uses. The IANA suggests dividing the port numbers into three ranges:

1 The **Well Known Ports** are those in the range 0–1023. For example, the port numbers of FTP, Telnet and the Simple Mail Transfer Protocol (SMTP) services are 21, 23 and 25 respectively. The services of these ports are standardized in Internet applications. On Unix-like operating systems, opening a port in this range to receive connections usually requires administrative privileges.

2 The **Registered Ports** are those in the range 1024–49151. These ports are not assigned and controlled by IANA, but they can be registered with IANA to prevent duplication.

3 The **Dynamic** and/or **Private Ports** are those in the range 49152–65535. Randomly chosen port numbers out of this range are called ephemeral ports. These ports are not permanently assigned to any publicly defined application. A client application usually uses an ephemeral port to a make connection to a well known or registered port on a server.

The combination of the IP address and the port used for communication is called a **socket address**. The socket address is unique for a session of communication between hosts.

Let's use FTP services as an example. A client makes two sessions of FTP connections to an FTP server with IP address 153.34.2.56. In order to correctly separate the packets for both sessions, the packet headers contain two different socket addresses for the client 153.34.2.56, port 4233, and 153.34.2.56, port 3245. The first FTP session uses the socket 153.34.2.56, port 4233. The next session uses 153.34.2.56, port 3245. As you can see, the server has no problem reply with the correct data to appropriate applications.

Socket addresses are specified using the notation: <IP Address>:<Port Number>. For example, for a website running on IP address 123.45.67.89, the socket address corresponding to the HTTP server would be 123.45.67.89:80.

The TCP segment structure

The following figure introduces the TCP segment structure.

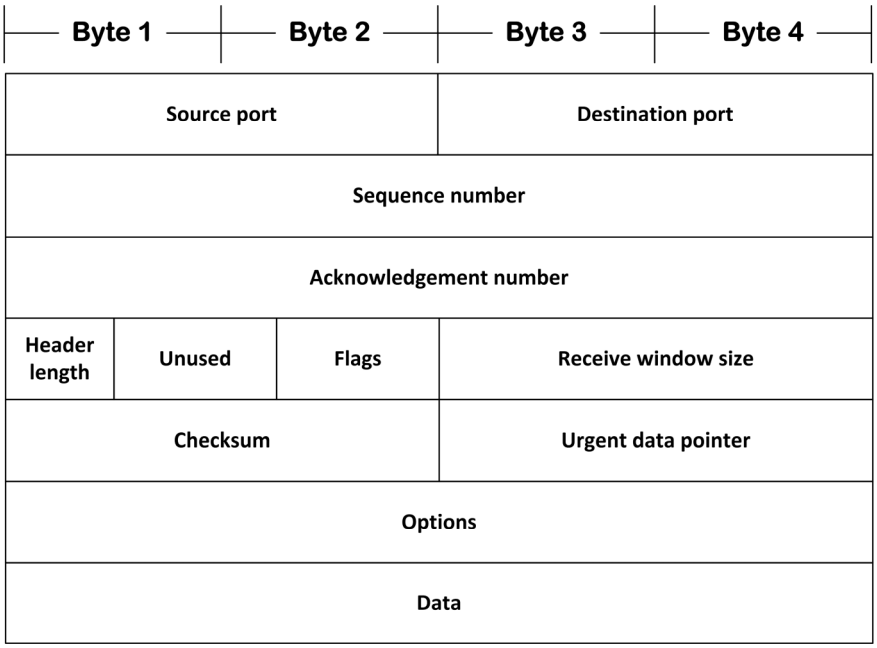


Figure 3.12 The TCP segment format

The fields of the TCP segment are as follows:

- *Source port* and *destination port* identify the port numbers of the communication.
- *Sequence number* usually specifies the sequence number of the first octet of data in the current segment.
- *Acknowledgement number* contains the sequence number of the next octet of data the sender of the packet expects to receive.
- *Header length* indicates the number of 32-bit words in the TCP header.
- *Flags* carry a variety of control information.
- *Receive window size* specifies the size of the sender's receiving window (i.e. buffer space available for incoming data).
- *Checksum* indicates whether the header was damaged in transit.
- *Urgent data pointer* points to the first urgent data octet in the packet.
- *Options* specify various TCP options.
- *Data* contains upper layer information.

The following reading in your set textbook lists some commonly used TCP/IP port numbers. You do *not* need to memorize every port number.

However, it will help you in configuring and troubleshooting TCP/IP network services.

> ***Reading***
>
> Dean (2012) 164–66.

The following Wikipedia article also lists the TCP and UDP port numbers in a nicely formatted table. You are recommended to bookmark this web page and refer to it from time to time.

http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers

Self-test 3.11

1 Explain the meaning of well known ports.

2 Explain the two objectives of well known ports.

3 Derive the socket address needed to connect to a POP3 server on the IP address 192.207.91.2.

TCP connection establishment: three-way handshaking

Before starting a communication session, TCP uses **three-way handshaking** to establish a connection. In a three-way handshake, each communicating host must send a **synchronizing (SYN) message** and then wait for an **acknowledgment (ACK) message** for it from the other host.

Thus, conceptually, four control messages must pass between the hosts to establish a connection. However, it is inefficient to send a SYN and an ACK in separate messages when it is possible to communicate both simultaneously. They can be sent together by setting both of the relevant bits in a single SYN+ACK message. This makes a total of three messages, and for this reason the connection process is called a **three-way handshake**.

The three-way handshake process is illustrated in the following figure:

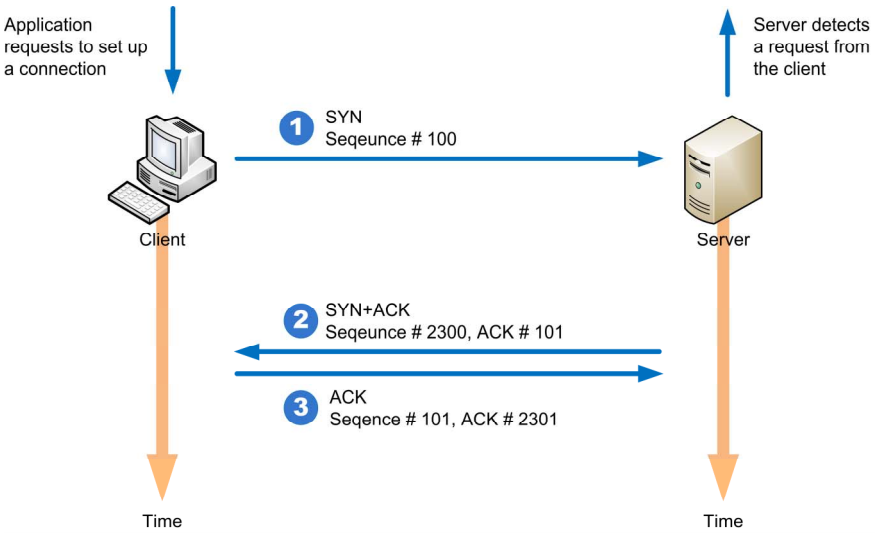


Figure 3.13 TCP three-way handshake connection establishment process

In Figure 3.13 you can see there are six steps of the connection process:

1 A server host opens a TCP port for services, e.g. 80 for a Web server. The server listens to the port and accepts connection if there are any request from clients.

2 A client initiates a request to start a connection to the server at the given port and IP address.

3 The client sends a SYN message with an **initial sequence number** (**ISN**), let's say, 100.

4 If the server receives the SYN message, and replies by sending a SYN+ACK message to the client with an ISN, let's say 2300, and the acknowledgment number 101. This implies the first sequence number from the server to the client is 2301 and the first TCP data segment sent by the client should have the sequence number 101.

5 At the other end, the client performs a similar process. An ACK message is sent with the sequence number 101 and the acknowledgment number 2301.

6 At this point, the connection between the client and server has been established. The connection is now ready to start data transmission between the client and server.

Interactive data flow

The data transfer is ready after the establishment of a connection between the source and destination. The mode of communication is full duplex. This means source and destination hosts are able to send data to each other at the same time. Each segment header includes an acknowledgement to specify the sequence number of the next byte expected from the other host.

To illustrate the data transfer mechanism in TCP, we describe the flow of data using TCP. Assuming the connection procedure has been established between the client and the server, with 100 octets in each segment, the first segment sent by the client has the sequence number 201 and the acknowledgement number 2401.

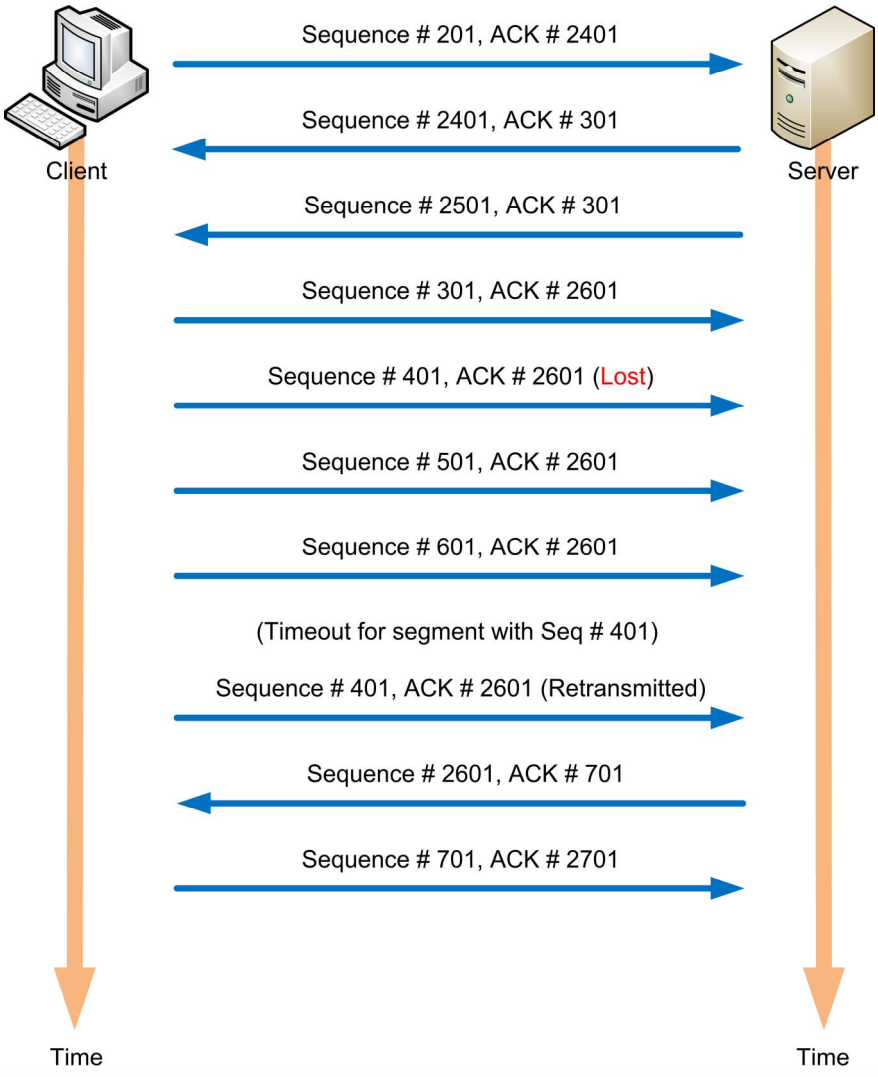


Figure 3.14 TCP data transfer and retransmission

Figure 3.14 shows a simple flow control. After the client sends a segment with the sequence number 201, the server delivers two segments back to the client. Note that a sender does not have to wait for acknowledgement of each packet before sending more data. This approach can improve the data throughput of the communication.

As you learned earlier, TCP is a reliable transport protocol that provides for the retransmission of corrupted or lost segments. Looking back again at Figure 3.14, you can see that the segment identified by sequence number 401 is lost in the transmission. Since the server does not receive the segment identified by sequence number 401, no further acknowledgement or reply is made to the client. After a predefined timeout period, the client determines that the segment identified by sequence number 401 is lost, and starts the retransmission of the

segment. The communication can proceed properly if the acknowledgement segment is sent to the client.

You should note that the last two segments have the ACK # 701, because the segments identified by sequence numbers 501 and 601 have already been received by the server.

After finishing the data transfer, it is necessary to close the connection session. The closing process involves the following request/reply procedures:

1 An application has finished its work and tells TCP to close the connection.

2 The host sends a close segment to inform its partner that it will send no more data.

3 The partner replies to the close request segment and stops its application.

4 The partner also sends a close segment to the host to confirm the end of communication.

5 After receiving the close segment for the partner, the host replies with the acknowledgement segment and stops its application.

The flow control algorithm

The incoming flow of data is controlled by the receiving host. The receiver has a buffer that determines how much data it is willing to accept before sending an acknowledgement to the sender. The buffer size is usually an integer multiple of the maximum segment size. Buffer space is used up as data arrives. When the receiving application removes data, space is cleared for more incoming data.

The sender's perspective

A sliding window is used to determine the buffer size of the TCP communication. For example, a sliding window with length 7 is applied to the data flow. The shaded block shown in Figure 3.15 below is the current segment to be sent or received. From the sender's perspective, it keeps a check on how much data has been sent and acknowledged. When the sender receives acknowledgement number 701 from the receiver, the sender's window starts the data sequence number 701. This implies the sender can transmit segments to the receiver up to data sequence number 1401 without acknowledgement from the receiver. The window shrinks from the trailing edge as it sends segments out. The benefit of sending data without waiting for the acknowledgement is that it can improve the throughput of the transmission. Furthermore, the receiver can reply with an acknowledgement sequence number such as 1301, to imply the data number from 701 to 1201 reached the destination.

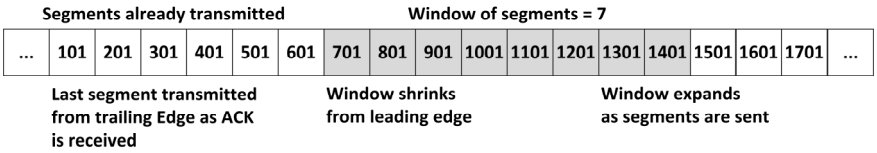
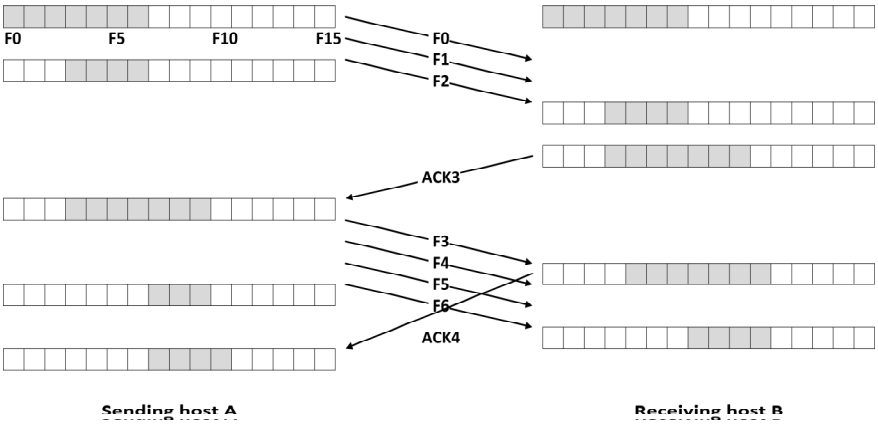


Figure 3.15 A sender window with length equal to 7

The receiver's perspective

The receiver window is a buffer to track unacknowledged segments. Data remain in the window until the receiver's application accepts them. The window extends from the last acknowledged byte to the end of the buffer. When the buffer is full, the receiver will no longer accept any segments, and it discards the incoming data. Until the receiving application absorbs the data in its buffer, the receivers will start to store further incoming segments. The discarded data will also be retransmitted from the senders. The situation of the sliding window between sender and receiver is shown in Figure 3.16.



where Fx = the x^{th} segment

Figure 3.16 Example of a sliding window protocol

Retransmission timeout

After sending a segment, TCP sets a timer and listens for an acknowledgement (ACK). If there is no ACK reply, the sender assumes the segment is lost in transmission, and it retransmits the segment to the receiver.

This leads, however, to a direct question of how long the timeout should be. If the retransmission is too short, duplicated segments may be retransmitted frequently to the network. This kind of unnecessary burden on network traffic degrades the network's performance. But on the other hand, timeouts that are too long will prevent prompt recovery when a segment really has been destroyed, and will decrease the network response time.

In order to design a suitable timeout parameter for TCP communication, a suitable algorithm with adaptive features must be developed according

to the network status. In practice, therefore, TCP cannot use a single number for this value. It must determine the value dynamically using a process called **adaptive retransmission**. This topic is beyond the scope of this course, however, so we will not discuss it further.

Self-test 3.12

In a TCP communication session, the quality of the data channel is not reliable and causes multiple retransmissions of data segments. The scenario is shown in Figure 3.17. Fill in all the missing entries (i.e. those with '????') in the diagram. The segment length is 50 octets.

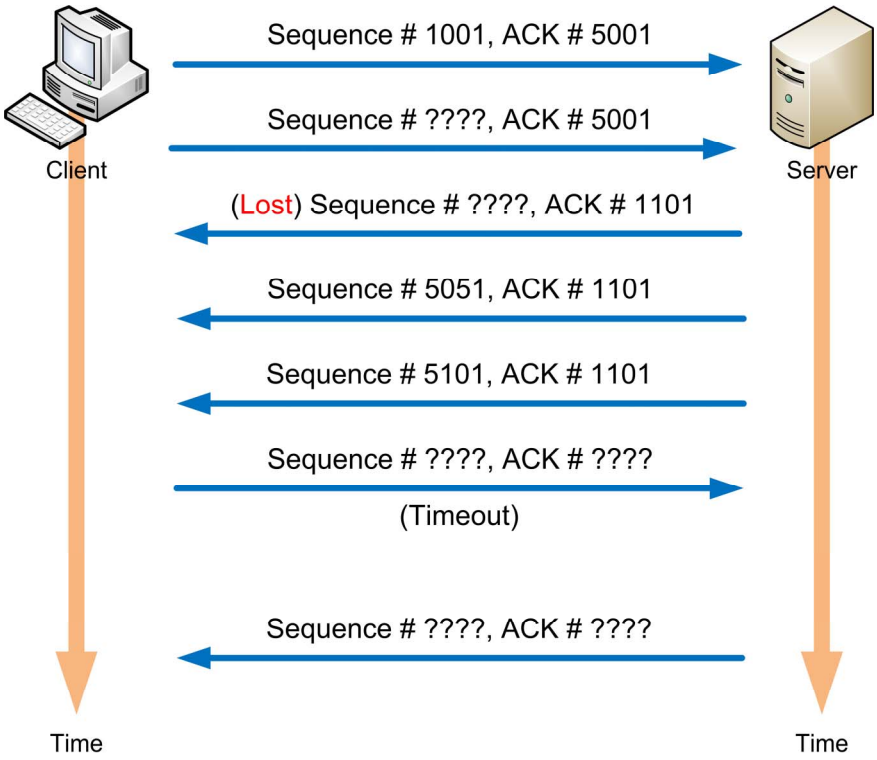


Figure 3.17 Typical TCP communication interactions between client and server

The User Datagram Protocol

In the transport layer of the Internet architecture, the TCP provides a significant services role for many applications. But TCP's major weakness is its relatively large overhead in data transmission. In order to cater for some applications with small data units or those that can afford to lose a little data (such as multimedia streaming), the **User Datagram Protocol** (**UDP**) has been developed to reduce the overheads of TCP data transmission.

UDP is a much simpler protocol than TCP. The UDP header has only four fields: the source port, destination port, length and UDP checksum. The source and destination port fields serve the same functions as they do in the TCP header. The length field specifies the length of the UDP header and data, and the checksum field allows packet integrity checking, but is optional with UDP.

Each data unit in UDP is called a 'datagram'. As its name implies, the User Datagram Protocol uses a connectionless approach. Unlike TCP, there is no need to establish a connection before sending data in UDP. Instead, UDP simply passes the individual datagrams to the IP layer for transmission. If a query in a UDP datagram is transmitted and a response does not reach the sender before timeout, the application itself has to handle the retransmission instead of UDP.

Given the limitations of UDP, you might wonder why it's used at all. But UDP has the advantage over TCP in two critical areas: speed and packet overheads. Because TCP is a reliable protocol, it spends a great deal of effort ensuring that data arrive at the destination and, as a result, it exchanges a fairly high number of packets over the network. UDP doesn't have this overhead so it is considerably faster than TCP. UDP is therefore the solution for situations in which speed is paramount, or in which the number of packets sent over the network has to been kept to a minimum.

Format of the UDP user datagram

The UDP user datagram is illustrated in the figure below.

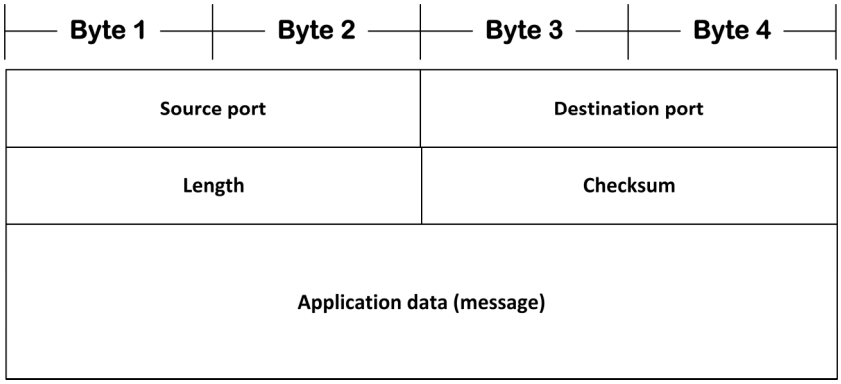


Figure 3.18 The user datagram format

The fields of the user datagram are as follows:

- *Source port* — a 16-bit port number
- *Destination port* — a 16-bit port number
- *Length* (UDP header + data) — a 16-bit count of octets in the UDP segment
- *Checksum* — a 16-bit field; if zero, then there is no checksum or else it is a checksum over a pseudo-header + UDP data area.

UDP uses a pseudo-header to verify that the UDP datagram has arrived at the correct host address with correct port number. The UDP pseudo-header consists of the source and destination IP addresses. Note that the source address, destination address and protocol field are taken from the IP packet header. The purpose of the UDP checksum is to check the correctness of a datagram. The UDP pseudo-header is shown in Figure 3.19.

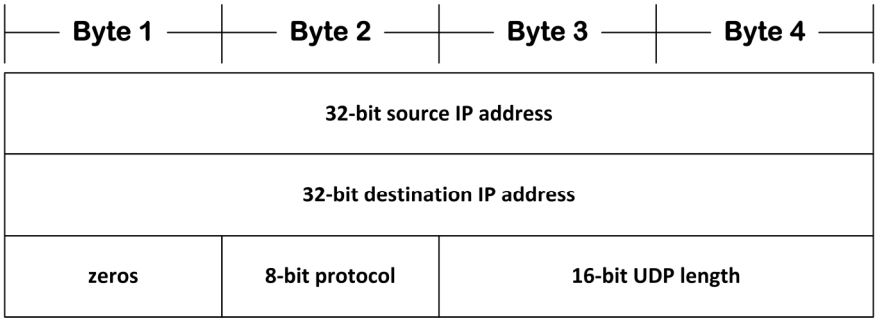


Figure 3.19 The UDP pseudo-header, used in computing the UDP checksum

How UDP works

The information needed to start a UDP communication is similar to that of its companion, TCP. Every UDP datagram is assigned a port number to identify a targeted application. Port numbers from 0 to 1023 are reserved for standard services. Referring to the Boostrap Protocol discussed previously, the BOOTP protocol works with UDP at ports 67 and 68 for server and client respectively. A partial list of well-known UDP services is shown in Table 3.7 below.

Table 3.7 A partial list of well-known UDP service ports

Port	Protocol	Description
7	Echo	Echoes a received datagram back to the sender
13	Daytime	Returns the data and time
17	Quote	Returns a quote of the day
53	DNS	Domain Name Server (Domain Name System)
123	NNTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
520	RIP	Routing Information Protocol (RIP-1 and RIP-2)

The communication procedure with UDP is simple. Instead of needing a connection establishment, a UDP client simply sends datagrams to the defined port and starts a communication with a server. There is no acknowledgement signal between the client and server. It is not UDP's responsibility to guarantee the delivery of datagrams. The application has to determine the correctness of the communication by itself.

Apart from the predefined service ports, the remaining port numbers are assigned by the corresponding application with an on-demand basis. The steps below indicate how this happens:

1 A sender starts its network program.

2 The sender requests the receiver to assign a port for communication.

3 The receiver selects an unused port from the pool of available ports and sends it back to the sender.

The source and destination port numbers are stored in each UDP header. In addition to the IP address stored in the IP header, the combination of the IP address and the port number forms a socket address for data transmission. Since both UDP and TCP data are encapsulated within the IP format, you may recall that the protocol field in the IP header is used to classify which protocol is running on the top IP layer. UDP has been assigned its unique protocol identifier, 17.

Self-test 3.13

1 When is the User Datagram Protocol used?

2 Explain why RIP for IP uses the UDP protocol for transport.

Summary

This unit explored the details of the Internet protocol suite, including the Internet Protocol, Transmission Control Protocol and User Datagram Protocol.

The operation of the Internet Protocol is the core of the Internet protocol suite. We began our study of the Internet Protocol by looking at the naming and addressing scheme for an IP host for the five classes of the IP address. To illustrate the relationship between the IP address and the physical address, we examined and applied the Address Resolution Protocol and its opposite, Reverse ARP. You saw how IP routing is closely related to ARP services. Routing processes determine paths of packets from sources to destinations. The Routing Information Protocol was applied to illustrate the concept of routing processes.

You then learned why the Internet Control Message Protocol is a useful service for checking network connectivity. ICMP works in a request-reply mode. In addition to being a diagnostic tool, ICMP can improve routing performance by using the ICMP redirect feature.

The widely-applied transport layer protocols in the Internet are TCP and UDP. TCP is a connection-oriented protocol with reliable communication. The combination of port number and IP address forms the socket for transmission. TCP can provide reliable data transmission using a send/acknowledge approach. If segment loss occurs in transmission, TCP automatically handles retransmission of the lost segment.

UDP is another protocol in the transport layer. Unlike the connection-oriented TCP model, the UDP datagram is a connectionless approach. The UDP format is much simpler TCP's. The simplicity of UDP enables it to be applied on applications such as ICMP, the Bootstrap Protocol and Simple Network Management Protocol. You also studied the details of TCP and UDP by analysing their data formats.

In this unit, we introduced you to only the very essential elements of the IP, TCP and UDP protocols. If you are interested in learning about these protocols in depth, we strongly encourage you to spend some time reading one or more of the reference textbooks listed in the References section at the end of this unit.

Suggested answers to self-tests and activities

Self-test 3.1

1 127.0.0.1

2 210.17.156.179

3 10.1.1.1

Self-test 3.2

1 Network address: 61.0.0.0

Host address: 10.128.100

2 Network address: 128.133.0.0

Host address: 5.254

3 Network address: 202.40.220.0

Host address: 3

Self-test 3.3

1 131.107.256.80 — this is invalid because the highest possible value in an octet is 255.

222.222.255.222 — this is a valid address.

231.200.1.1 — this is invalid because 231 is a class D address, and is not supported as a host address.

0.127.4.100 — this is invalid because the first octet cannot be 0.

127.1.1.1 — this is invalid because 127 addresses are reserved for diagnostics.

2 Class A address — the host address field is 24-bit in length, so the number of valid host addresses is $2^{24} – 2 = 16{,}777{,}214$ (Remember the two host addresses of all 0s and all 1s cannot be used, so you need to deduct these two).

Class B address — the host address field is 16-bit in length, so the number of valid host addresses is $2^{16} – 2 = 65{,}534$.

Class C address — the host address field is 8-bit in length, so the number of valid host addresses is $2^8 – 2 = 254$.

In general, for a host address field of length n, the number of valid host addresses is $2^n – 1$.

Self-test 3.4

1 Your company is granted a class B address 172.16.0.0.

a The default class B mask is 255.255.0.0. By using a subnet mask of 255.255.255.0, the subnet portion of the address will consist of 8 bits. This will provide $2^8 = 256$ subnets. The remaining host portion will consist of 32 – 16 – 8 = 8 bits. The number of addresses in each subnet will therefore be $2^8 = 256$, and the maximum number of hosts in each subnet will be 256 – 2 = 254.

b First write down the address and the subnet mask in binary notation and do a bitwise-AND operation:

```
IP address      172      .16      .10      .50  dotted decimal
                10101100.00010000.00001010.00000000
Subnet mask     11111111.11111111.11111111.00000000
Result of AND   10101100.00010000.00001010.00000000
                172      .16      .10      .0   dotted decimal
```

This gives you a subnet address of 172.16.10.0. Since each subnet consists of 256 addresses, the maximum number of hosts is 254, and the host range is from 172.16.10.1 to 172.16.10.254.

c Since $2^{10} < 2046 < 2^{11}$, you need an 11-bit subnet, and the subnet mask is 11111111.11111111.11111111.11100000 or 255.255.255.224.

```
IP address      172      .16      .10      .170 dotted decimal
                10101100.00010000.00001010.10101010
Subnet mask     11111111.11111111.11111111.11100000
Result of AND   10101100.00010000.00001010.10100000
                172      .16      .10      .160 dotted decimal
```

You can see that the IP address is in subnet 172.16.10.160. Since you have only 5 bits for the hosts, each subnet can have up to $2^5 – 2 = 30$ hosts, and the host range is from 172.16.10.161 to 172.16.10.190.

2

Personnel:	202.45.2.0	255.255.255.224	202.45.2.1 to 30
Financial:	202.45.2.32	255.255.255.224	202.45.2.33 to 62
Marketing:	202.45.2.64	255.255.255.224	202.45.2.65 to 94
R&D:	202.45.2.96	255.255.255.224	202.45.2.97 to 126
Production:	202.45.2.128	255.255.255.224	202.45.2.129 to 158

Self-test 3.5

1 Network: 192.168.4.0/22

```
Prefix: 192.168.4.0   192.168.000001 00.00000000
Mask: /22             255.255.111111 00.00000000
Addresses:            192.168.000001 00.00000000 through
                      192.168.000001 11.11111111
                      or
                      192.168.4.0 through 192.168.7.255
```

2 Address: 10.0.8.0/22

```
Address: 10.0.8.0          00001010.00000000.000010 00.00000000
Mask: /22                  11111111.11111111.111111 00.00000000
```

The host ID is all 0s, so 10.0.8.0/22 is a network address.

Address: 172.17.16.255/23

```
Address: 172.17.16.255     10101100.00010001.0001000 0.11111111
Mask: /23                  11111111.11111111.1111111 0.00000000
```

The host ID contains both 0s and 1s, so 172.17.16.255/23 is a host address.

Address: 192.168.37.127/25

```
Address: 192.168.37.127    11000000.10101000.00100101.0 1111111
Mask: /25                  11111111.11111111.11111111.1 0000000
```

The host ID is all 1s, so 192.168.37.127/25 is a broadcast address.

3 CIDR overcomes the limitations of classful addressing by eliminating the concept of address classes entirely. The first three bits of the IP address are no longer tied to specifying the class of the address. Instead of classes, CIDR-based networks are addressed in sections called CIDR blocks, which are identified by prefixes. The prefixes can have variable length, as opposed to the fixed 8, 16 and 24 bits network prefixes in classful addressing. Hence, network blocks in various sizes can be flexibly assigned.

Self-test 3.6

First, on host A, enter the commands '**`ping H2`**', '**`ping H3`**', '**`ping H4`**' and '**`ping H5`**' such that the ARP cache in H1 stores the information of the H2 to H5 hosts. You can show the ARP cache entries by using the '**`arp -a`**' command at host H1. Similarly, you can ping H1 from H2 and then check the ARP cache in H2 to look for the hardware address of H1. Repeat the operations on hosts H3 to H5.

Self-test 3.7

The two primary differences between DHCP and BOOTP are:

- DHCP defines mechanisms through which clients can be assigned a network address for a certain lease period, allowing for subsequent reassignment of network addresses to different clients.
- DHCP provides the mechanism for a client to acquire all of the IP configuration parameters that it needs in order to operate.

Self-test 3.8

Destination network address with subnet mask, the address of the next hop router used to reach the destination network, and a metric.

Self-test 3.9

1 It creates too much broadcast traffic. It can take a long time for RIP information to propagate among all routers.

2 RIP is based on a hop-count metric and uses a distance-vector algorithm. The maximum number of hops supported by RIP is 15. RIP is therefore a good dynamic protocol for smaller networks, but can cause congestion in larger environments.

   OSPF uses the Dijkstra algorithm to calculate paths through the network. It calculates routes based on several variables including the number of hops, line speed and cost. OSPF is therefore the protocol of choice in large inter-networked TCP/IP environments.

Self-test 3.10

1 Type = 11 and Code = 0

2 On Linux, assuming you are root:

```
ping -l 128 -s 15 www.ouhk.edu.hk
```

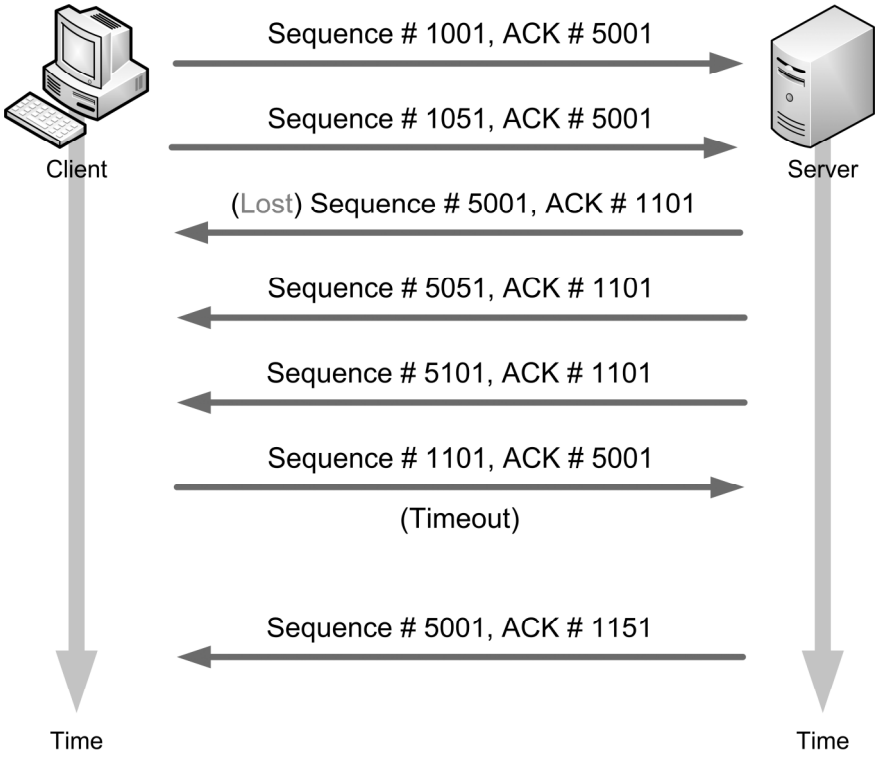
Self-test 3.11

1 A well known port is simply a default communication channel used for a particular service or protocol. It is a TCP or UDP port with a port number between 1 and 1023.

2 The first objective is to provide standard process across systems to communicate with TCP/IP processes, services and protocols. The second is to provide easy nomenclature for an end user to connect to a standard TCP/IP resource.

3 192.207.91.2:110

Self-test 3.12



Self-test 3.13

1 UDP is used when an application needs to send connectionless traffic, typically sending a messaging to multiple receiving stations.

2 UDP is a broadcast-based transport mechanism offering nonguaranteed transmissions. It asks all of the other routers for changes. If one router doesn't respond, it doesn't matter at all, because it sends a broadcast every 30 seconds. RIP uses UDP port 520 for all of its communications.

Activity 3.3

1 Use the command: **`nslookup www.ouhk.edu.hk`**

The IP address of www.ouhk.edu.hk is 202.40.220.3.

2 Use the command: **`nslookup 202.40.157.167`**

The host name of IP address 202.40.157.167 is sun20.ouhk.edu.hk.

3 Use the 'nslookup' command and follow the steps below. The result shown below is taken from a Windows XP computer. Your result should be similar to this:

```
C:\>nslookup
Default Server:  hk1dns02.iadvantage.net.hk
Address:  202.85.128.33

> set type=ns
```

```
> ouhk.edu.hk
Server:  hk1dns02.iadvantage.net.hk
Address:  202.85.128.33

Non-authoritative answer:
ouhk.edu.hk     nameserver = ns2.ouhk.edu.hk
ouhk.edu.hk     nameserver = ns2.cuhk.edu.hk
ouhk.edu.hk     nameserver = ns1.ouhk.edu.hk

ns1.ouhk.edu.hk internet address = 202.40.157.167
ns2.cuhk.edu.hk internet address = 137.189.6.21
ns2.cuhk.edu.hk AAAA IPv6 address = 2405:3000:3:60::21
ns2.ouhk.edu.hk internet address = 202.40.157.162
> exit
```

As my result shows, there are three DNS servers for the 'ouhk.edu.hk' domain: 'ns1.ouhk.edu.hk', 'ns2.ouhk.edu.hk' and 'ns2.cuhk.edu.hk'. One of them should be the primary DNS server for the 'ouhk.edu.hk' domain, and the other two are secondary DNS servers are redundancy purposes.

4 Again, use the 'nslookup' command and follow the steps below:

```
C:\>nslookup
Default Server:  hk1dns02.iadvantage.net.hk
Address:  202.85.128.33

> root
Default Server:  A.ROOT-SERVERS.NET
Address:  198.41.0.4
> exit
```

In the above example, the root name server is 'a.root-servers.net'. If you do not understand the commands supported by 'nslookup', you can enter 'help' at the nslookup prompt. The help manual for 'nslookup' will be displayed.

Note that the above IP addresses and host names might change, so don't be surprised if your lookups give you different results.

Glossary

adaptive routing — See *dynamic routing*.

Address Resolution Protocol (ARP) — a core protocol in the TCP/IP suite that belongs in the Network layer of the OSI Model. ARP obtains the MAC (physical) address of a host, or node, and then creates a local database that maps the MAC address to the host's IP (logical) address.

authoritative — an authoritative **DNS** for a section of the Domain Name Server name tree, for which one name server is the authority.

Bootstrap Protocol (BOOTP) — a UDP network protocol used by a network client to obtain its IP address automatically. It has been superseded by the Dynamic Host Configuration Protocol (DHCP).

buffer — an area of memory used for storing messages.

cache — a small, fast memory holding recently accessed data, designed to speed up subsequent access to the same data; most often applied to processor memory access, but also used for a local copy of data accessible over a network.

checksum — a computed value that depends on the contents of a block of data and which is transmitted or stored along with the data in order to detect corruption of the data.

daemon — a program that runs in the background as a background process, and that lies dormant waiting for some condition(s) to occur and then begins processing.

Domain Name System (DNS) — is a hierarchical naming system for computers, services, or any resource participating in the Internet. It associates various types of information with domain names assigned to such participants. Most importantly, it translates humanly meaningful domain names to the numerical (binary) identifiers associated with networking equipment for the purpose of locating and addressing these devices world-wide. An often used analogy to explain the Domain Name System is that it serves as the 'phone book' for the Internet by translating human-friendly computer hostnames into IP addresses.

Dynamic Host Configuration Protocol (DHCP) — a protocol used by networked devices (clients) to obtain the parameters necessary for operation in an Internet Protocol network. This protocol reduces system administration workload, allowing devices to be added to the network with little or no manual configuration.

dynamic routing — the use of routing protocols to dynamically update routing tables. In dynamic routing, routers periodically exchange routing information and update each other according to some routing protocol such that any change in the internetworking structure is efficiently propagated to all routers in the internetwork. Such dynamic routing

information updates occur automatically and no administrative effort is required.

File Transfer Protocol (FTP) — a protocol that enables users to copy files between systems and perform file-management functions.

full duplex — a term used to describe a communications channel down which data can travel in both directions simultaneously.

hop — a term used to describe the passage of a data packet between two network nodes (e.g. between two routers).

IEEE 802 — a series of standards to guide many vendor implementations; these standards have been recognized by the International Organization for Standardization (ISO).

Internet Control Message Protocol (ICMP) — a helper protocol to the Internet Protocol that allows for the generation of error messages, test packets and informational messages related to IP.

Internet Protocol (IP) — the predominant network layer protocol in the TCP/IP stack that offers a connectionless and best-effort internetwork service. IP provides features for addressing, type of service specification, fragmentation and reassembly, and routing.

Internet Protocol Version 6 (IPv6) — the new generation of IP; also known as IP Next Generation (IPng).

Karn and Jacobson's Algorithm — an algorithm for determining the timeout parameter in TCP/IP transmissions.

link state — the state of routing information based on the number of hops, link speeds, traffic congestion and other factors determined by the network designer.

loopback address — an IP address reserved for communicating from a node to itself (used mostly for troubleshooting purposes). The loopback address is almost always cited as 127.0.0.1, although in fact, transmitting to any IP addresses whose first octet is 127 will always 'loop back' to the originating device.

Media Access Control (MAC) — the interface between a node's Logical Link Control and the network's physical layer.

multiplexing — combining several signals for transmission on one shared medium.

Network Operating System (NOS) — an operating system designed to pass information and communicate between more than one computer. Examples include Unix, Windows Server 2000/2003/2008, Apple OS X and Novell NetWare.

octet — an 8-bit quantity; byte and octet are often used interchangeably.

Open Shortest Path First (OSPF) — a link state routing protocol.

ping — a program used to test whether a particular host is reachable across an IP network. It works by sending ICMP 'echo request' packets to the target host and listening for ICMP 'echo response'.

private address — an address that can be used on a private network, but that is not routable through the Internet. If a host on a private network needs to connect to the Internet, its IP address must be converted a public and routable address (real IP) by a mechanism called Network Address Translation (NAT). As such, many hosts on the private network can use the same real IP to connect to the Internet.

pseudo-header — the data portion in a datagram which contains information working as a header of the datagram.

Request for comment (RFC) — one of a series, begun in 1969, of numbered Internet informational documents and standards widely followed by commercial software and freeware in the Internet and UNIX communities.

reserved address — an address in one of the reserved address blocks set aside for future experimentation or for internal use in managing the Internet.

Reverse Address Resolution Protocol (ARP) — a protocol to look up the IP address for the physical address.

Root Name Server — the name server for a domain at the top of the domain name hierarchy.

round-trip time (RTT) — the transmission time for a packet to be sent to a destination and then returned to the sender.

route flush time — the expired time for routing information in a router.

Routing Information Protocol (RIP) — a dynamic routing protocol used in local area networks. An RIP is an interior gateway protocol (IGP) using the distance-vector routing algorithm.

Simple Mail Transfer Protocol (SMTP) — the *de facto* standard for email transmission across the Internet. The original SMTP was first defined in RFC 821 and was then amended by RFC 1123. The protocol in widespread use today is also known as **extended SMTP** (**ESMTP**); it was defined in RFC 5321.

sliding window — a window to be applied in the flow control technique of the data transmission.

Smoothed round trip time (SRTT) — the average round trip time of the last few send and acknowledgement packet timestamps.

Source Quench — a technique to avoid the retransmission of datagrams because of routers running out of buffer space.

static routing — a routing method in which routing information is manually configured on the router, creating what is known as a static route. Static routing requires a network administrator for initial setup and for any subsequent changes to routes.

subnet — a portion of a network, which may be a physically independent network segment that shares a network address with other portions of the network and is distinguished by a subnet number.

subnet mask — a bit mask used to identify which bits in an IP address correspond to the network address and the subnet portions of the address.

subnetting — the process of dividing a network into subnets.

telnet — the TCP/IP protocol that enables a terminal attached to one host to log into other hosts and interact with their applications.

Time to live (TTL) — the field in the Internet Protocol header that indicates how many more hops the packet should be allowed to make before being discarded or returned.

timeout — the period after which an error condition is raised if some event has not occurred.

Transmission Control Protocol (TCP) — a connection-oriented transport layer protocol in the TCP/IP protocol suite that provides reliable full-duplex data transmission.

Transmission Control Protocol over Internet Protocol (TCP/IP) — the *de facto* standard Ethernet protocols incorporated into 4.25BSD UNIX. TCP/IP was developed by the Defense Advanced Research Agency (DARPA) for internetworking, encompassing both network layer and transport layer protocols. TCP and IP specify two protocols at specific layers, but TCP/IP is often used to refer to the entire US Department of Defense (DoD) protocol suite based on these, including Telnet, FTP and UDP.

User Datagram Protocol (UDP) — a connectionless transport layer protocol in the TCP/IP protocol suite. UDP is more efficient than TCP due to the lack of connection establishment and management overhead.

References

Dye, M, McDonald, R and Rufi, A (2007) *Network Fundamentals: CCNA Exploration Companion Guide*, Cisco Press.

Forouzan, B A (2006) *Data Communications and Networking*, 4th edn, McGraw Hill.

Halsall, F (2005) *Computer Networking and the Internet*, 5th edn, Addison Wesley.

Kozierok, C M (2005) *The TCP/IP Guide*, No Starch Press.

Kurose, J and Ross, K W (2008) *Computer Networking: A Top-Down Approach*, 4th edn, Addison Wesley.

Stallings, W (2006) *Data and Computer Communications*, 8th edn, Prentrice Hall.

Tanenbaum, A S (2002) *Computer Networks*, 4th edn, Prentice Hall.

Online materials

https://www.iplocation.net/subnet-calculator

RFC 791 — http://tools.ietf.org/html/rfc791

RFC 826 — http://tools.ietf.org/html/rfc826

RFC 903 — http://tools.ietf.org/html/rfc903

RFC 950 — http://tools.ietf.org/html/rfc950

RFC 1122 — http://tools.ietf.org/html/rfc1122

RFC 2460 — http://tools.ietf.org/html/rfc2460

http://www.ietf.org/rfc.html

http://www.internic.net/faqs/domain-names.html

http://www.ipv6.com/

http://www.ipv6.org/

http://www.ipv6forum.com/

http://www.register.com/

http://en.wikipedia.org/wiki/Domain_name_system

http://en.wikipedia.org/wiki/Dynamic_Host_Configuration_Protocol

http://en.wikipedia.org/wiki/Internet_Protocol

http://en.wikipedia.org/wiki/IPv6

http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers