DATA COMMUNICATION AND NETWORKS

DSE 2258

What is DCN?

A system designed to transfer data from one network access point to one other or more network access points via data switching, transmission lines, and system controls.

Why study DCN?

Source: https://www.statista.com/statistics/617136/digital-population-worldwide/

60% of the world population are connected to the Internet on a daily basis

Where can we apply knowledge of DCN?

- Resource sharing such as printers and storage devices
- Exchange of information by means of e-Mails and FTP
- Information sharing by using Web or Internet
- Interaction with other users using dynamic web pages
- Video conferences
- Instant messaging

Role of Data Science in DCN

- Compares incoming data and makes appropriate decisions.
- Intelligence network analysis is used for spotting bottlenecks, evaluating the health of devices, issue remediation, identifying connected endpoints, and probing for potential security lapses.
- Network analytics may recommend corrective actions for identified issues in the network.
- For identifying an endpoint, correlate the data from other sources and build a profile for the endpoint.
- For detecting potential security issues, by monitoring the endpoint behavior and traffic (even encrypted) to detect anomalies.

Source: https://www.cisco.com/c/en_in/solutions/analytics/what-is-network-analytics.html#~benefits

Course Structure

TCP/IP Model

Course Structure (contd..)

Network Protocols

Routing Protocols

Course Structure (contd..)

UDP

USER DATAGRAM PROTOCOL

SIMPLE CONNECTIONLESS PROTOCOL

WHAT KIND OF APPLICATIONS USE UDP?

QUERY-RESPONSE PROTOCOLS

DNS NTP

TIME-SENSITIVE APPLICATIONS

IPTV VOIP ONLINE GAMES

BROADCASTING SERVICES

MULTICAST

SERVICE DISCOVERY SYSTEM ROUTING PROTOCOL

TCP

TRANSMISSION CONTROL PROTOCOL

COMPLEX CONNECTION-ORIENTED PROTOCOL

WHERE IS TCP USED?

ONE OF THE MOST COMMONLY USED PROTOCOLS

ON THE INTERNET

HTTP FTP SSH SMTP

Course Structure (contd..)

Course Structure (contd..)

Text Books To Refer

1. Behrouz A. Forouzan, TCP/IP Protocol Suite, 4th Edition, Tata McGraw Hill, 2010.

2. Tannenbaum, A.S, Computer Networks, 5th Edition, Prentice Hall of India EE Edition, 2011.

3. Behrouz A. Forouzan, Data Communications and Networking, 5th Edition, Tata McGraw Hill, 2013.

4. Leon Garcia and Widjala, Communication Networks, 5th Edition, Tata McGraw Hill, 2017.

Basic concepts of computer networks

DCN

Network definition, classification and topologies

What is DCN?

The term telecommunication means communication at a distance. The word data refers to information presented in whatever form is agreed upon by the parties creating and using the data. The data communication network is the exchange of data between two devices via some form of transmission medium such as a wire cable.

Figure 1.1 ***Components of a data communication system***

Data flow between devices

Figure 1.2 ***Data flow (simplex, half-duplex, and full-duplex)***

What is a Network?

A network is a set of devices (often referred to as nodes) connected by communication links.

A node can be a computer, printer, or any other device capable of sending and/or receiving data generated by other nodes on the network.

A link can be a cable, air, optical fiber, or any medium which can transport a signal carrying information.

Network Criteria

- Performance
  - Depends on Network Elements
  - Measured in terms of Delay and Throughput
- Reliability
  - Failure rate of network components
  - Measured in terms of availability/robustness
- Security
  - Data protection against corruption/loss of data due to:
    - Errors
    - Malicious users

Physical Structures

- Type of Connection
  - Point to Point - single transmitter and receiver
  - Multipoint - multiple recipients of single transmission

Figure 1.3 *Types of connections: point-to-point and multipoint*

Classification of networks

- The networks can be classified based on one of the two dimensions:
  - Transmission Technology
  - Scale

Classification of networks (contd..)

- Based on Transmission Technology

Point to point networks

- Point-to-point links connect individual pairs of machines.
- Point-to-point transmission with exactly one sender and exactly one receiver is called **unicasting**.

Note: Some broadcast systems also support transmission to a subset of the machines, which known as **multicasting**

Broadcast network

- In a broadcast network, the communication channel is shared by all the machines on the network;
- Packets sent by any machine is received by all the others.
- An address field within each packet specifies the intended recipient. Upon receiving a packet, a machine checks the address field.
- If the packet is intended for the receiving machine, that machine processes the packet; if the packet is intended for some other machine, it is just ignored.

Classification of networks (contd..)

- Based on Scale

Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	Wide area network
1000 km	Continent	
10,000 km	Planet	The Internet

(WAN) Wide Area Network

Metropolitan Area Network

Campus Area Network(CAN)

Home Area Network(HAN)

Table: Classification of networks based on the area of coverage

Personal Area Networks (PAN)

- **PANs** (**Personal Area Networks**) let devices communicate over the range of a person.
- A common example is a wireless network that connects a computer with its peripherals using Bluetooth.
- It is often used to connect a headset to a mobile phone without cords and it can allow your digital music player.
- PANs can also be built with other technologies that communicate over short ranges.

Local Area Networks (LANs)

- A LAN is a privately owned network that operates within and nearby a single building like a home, office or factory.
- LANs are widely used to connect personal computers and consumer electronics to let them share resources (e.g., printers) and exchange information.
- When LANs are used by companies, they are called **enterprise networks**.
- In these systems, every computer has a radio modem and an antenna that it uses to communicate with other computers via a device called an access point or wireless router.

Wired and Wireless LANs

- This device, called an **AP** (**Access Point**), **wireless router**, or **base station**, relays packets between the wireless computers and also between them and the Internet.

Wired and Wireless LANs (contd..)

- The standard for wireless LANs is called **IEEE 802.11**, popularly known as **WiFi**, which has become very widespread.
- It runs at speeds anywhere from 11Mbps to 100 Gbps
- Wired LANs use a range of different transmission technologies. Most of them use copper wires, but some use optical fiber
- The topology of many wired LANs is built from point-to-point links. IEEE 802.3, popularly called **Ethernet**.
- Each computer speaks the Ethernet protocol and connects to a box called a **switch** with a point-to-point link.
- A switch has **multiple ports**, each of which can connect to one computer. The switch is to relay packets between computers that are attached to it, using the address in each packet to determine which computer to send it to.

Wired and Wireless LANs (contd..)

- The wired and wireless networks can be either static or dynamic based on the channel allocated.
- A typical static allocation would be to divide time into discrete intervals and use a round-robin algorithm, allowing each machine to broadcast only when its time slot comes up.
- Static allocation wastes channel capacity when a machine has nothing to say during its allocated slot, so most systems attempt to allocate the channel dynamically (i.e., on demand).
- Dynamic allocation methods for a common channel are either centralized or decentralized.
- In the centralized channel allocation method, there is a single entity, which determines who goes next. It might do this by accepting multiple packets and prioritizing them according to some internal algorithm.
- In the decentralized channel allocation method, there is no central entity; each machine must decide for itself whether to transmit.

Metropolitan Area Networks (MANs)

- MANs generally cover a city.
- The best-known examples of MANs are the cable television networks available in many cities.
- These systems grew from earlier community antenna systems used in areas with poor over-the-air television reception.
- Cables were being laid to facilitate this approach.
- When the Internet began attracting a mass audience, the cable TV network operators began to realize that with some changes to the system, they could provide Internet service in unused parts of the spectrum.

Wide Area Networks (WANs)

- A **WAN** (**Wide Area Network**) spans a large geographical area, often a country or continent.
- It contains a collection of machines called hosts intended for running application programs.
- These hosts are connected by a communication subnet.
- The subnet consists of two components the transmission lines and switching elements (routers).

Wide Area Networks (WANs) contd..

- When two routers that are not connected directly need to communicate, then the routers communicate through the intermediate routers.
- Each of the intermediate routers stores the entire data and forwards it once the line is free. This mechanism is called as **Store and Forward** or **Packet Switched**

Wireless Networks

- The wireless networks can be divided into three main categories:
  - System Interconnections
  - Wireless LANs
  - Wireless WANs

  Based on distance
- **System interconnection** refers **interconnecting of devices using a short radio** range. Ex: Bluetooth mouse keyboard, headphones, etc., connected to the computer
- **Wireless LANs:** These are the systems in which the computer has a modem and antenna with which it can communicate with other systems
- The best example for **Wireless WANs** is the cellular telephones, where the communication distances involved are much higher.

Home Networks

- In these networks the fundamental idea is that every device in home is capable of communicating with every other device, and all of them are accessible over the internet.
- Some of the devices that are capable of being networked are:
  - Computers (desktop PC, PDA, shared peripherals
  - Entertainment (TV, DVD, VCR, camera, stereo, MP3)
  - Telecomm (telephone, cell phone, intercom, fax)
  - Appliances (microwave, fridge, clock, furnace, airco)
  - Telemetry (utility meter, burglar alarm, baby cam).

Home Networks (contd..)

A home network should have the following properties:

- The network and the devices have to be easy to install
- The network and devices have to be foolproof in operation
- Low price is essential for success
- Since most of the applications include multimedia, the network needs sufficient capacity to handle.
- Security and reliability
- The devices should be compatible with the introduction of new technologies.

The home networks devices can be wired or wireless.

Wireless networking is preferable over wired network, due to low cost but the problem with wireless is that the radio waves may lead to ill effects and wireless facilities can be used by our neighbors if security provisions are absent.

Internetworks

- Many networks exist in the world, often with different hardware and software.
- People connected to one network often want to communicate with people attached to a different one.
- The fulfillment of this desire requires that different, and frequently incompatible, networks be connected.
- A collection of interconnected networks is called an **internetwork or internet**.
- Ex: connecting a LAN and a WAN or connecting two LANs is the usual way to form an internetwork.
- The general name for a machine that makes a connection between two or more networks and provides the necessary translation, both in terms of hardware and software, is a **gateway**.

Network Topology

- A link is a communications pathway that transfers data from one device to another.
- For communication to occur, two devices must be connected in some way to the same link at the same time.
- There are two possible types of connections: point-to-point and multipoint.
- **Point-to-Point** A point-to-point connection provides a dedicated link between two devices. The entire capacity of the link is reserved for transmission between those two devices.
- **Multipoint:** A multipoint connection is one in which more than two specific devices share a single link. In a multipoint environment, the capacity of the channel is shared.

Fig: Point to Point and broadcast Connection

Categories of topology

Bus

Star

Ring

Mesh

Tree

Point-to-Point

Mesh Topology

- Every node has a dedicated point to point link with every other node
- The total number of links in a mesh topology with ‘n’ nodes is n(n-1).
- However, if each physical link allows communication in duplex mode, we can divide the number of links by 2. In other words, we can say that in a mesh topology, we need n(n-1)/2 duplex mode links.

Figure 1.5 *A fully connected mesh topology (five devices)*

Mesh Topology (contd..)

Problem 1

- If an organization has 6 nodes, and wants to link the nodes using topology. Find the total number of links required for the construction of the entire network for simplex and duplex communication, and the number of ports required by each node

Mesh Topology (contd..)

Solution:

Given number of nodes in the organization ‘n’ = 6

As each node is connected to the remaining five (n-1) nodes in simplex mode of communication, thus the total number of links required is n(n-1) = 6 * 5 = 30.

In case of duplex mode the total number of links required is

n(n-1)/2 = 6 * 5 = 30/2 = 15.

Since each node has to connect to the remaining n-1 nodes thus the number of ports required for each node is 6-1 = 5.

Mesh Topology (contd..)

Advantages

- First, the use of dedicated links guarantees that each connection can carry its own data load, thus eliminating the traffic problems that can occur when links must be shared by multiple devices.
- Second, a mesh topology is robust. If one link becomes unusable, it does not alter the entire system.
- Third, there is the advantage of privacy or security. When every message travels along a dedicated line, only the intended recipient sees it.
- Finally, point-to-point links make fault identification and fault isolation easy.

Disadvantages

- The main disadvantages of a mesh are related to the amount of cabling and the number of I/O ports required.
- First, because every device must be connected to every other device, installation and reconnection are difficult.
- Second, the sheer bulk of the wiring can be greater than the available space (in walls, ceilings, or floors) can accommodate.
- Finally, the hardware required to connect each link (I/O ports and cable) can be prohibitively expensive.
- For these reasons a mesh topology is usually implemented in a limited fashion.
- Ex. Telephone cables

Star topology

- In a star topology, each device has a dedicated point-to-point link only to a central controller, usually called a hub.
- The devices are not directly linked to one another.
- Unlike a mesh topology, a star topology does not allow direct traffic between devices.
- If one device wants to send data to another, it sends the data to the controller, which then relays the data to the other connected device

Figure 1.6 *A star topology connecting four stations*

Star topology (contd..)

Problem 2

- If an organization has 6 nodes, and wants to link the nodes using star topology. Find the total number of links required for the construction of the entire network for simplex and duplex communication and the number of ports required by each node

Star topology (contd..)

Solution:

Given the number of nodes in the organization ‘n’ = 6

As each node is connected only to the hub, whether it is simplex or duplex the total number of links will be equal to the number of nodes = 6

Since each node has to connect only to the hub thus the number of ports required for each node is 1.

Star Topology (contd..)

Advantages

- A star topology is less expensive than a mesh topology.
- In a star, each device needs only one link and one I/O port to connect it to any number of others.
- Easy to install and reconfigure.
- If one link fails, only that link is affected. Hence the topology supports robustness.
- Provides easy fault identification and fault isolation.

Disadvantages

- Although a star requires far less cable than a mesh, each node must be linked to a central hub. For this reason, often more cabling is required in a star than in some other topologies (such as ring or bus).
- The star topology is used in local-area networks (LANs).
- High Speed LAN use star topology.

Bus topology

- Bus topology is a multipoint link where one long cable acts as a backbone to link all the devices in the network.
- Eg : Ethernet LAN

Figure 1.7 ***A bus topology connecting three stations***

Bus Topology (contd..)

Advantages

- Advantages of a bus topology include ease of installation.
- Backbone cable can be laid along the most efficient path, then connected to the nodes by drop lines of various lengths.
- In this way, a bus uses less cabling than mesh or star topologies.

Disadvantages

- Adding new devices may therefore require modification or replacement of the backbone.
- In addition, a fault or break in the bus cable stops all transmission, even between devices on the same side.

Ring topology

- In a ring topology, each device has a dedicated point-to-point connection with only two devices on either side of it.
- A signal is passed along the ring in one direction, from device to device, until it reaches its destination.
- Each device in the ring incorporates a repeater. When a device receives a signal intended for another device, its repeater regenerates the bits and passes them along.

Figure 1.8 *A ring topology connecting six stations*

Ring Topology (contd..)

Advantages

- This topology is relatively easy to install and reconfigure as each device is linked to only its immediate neighbors (either physically or logically).
- To add or delete a device requires changing only two connections.
- Generally in a ring, a signal is circulating at all times. If one device does not receive a signal within a specified period, it can issue an alarm. The alarm alerts the network operator to the problem and its location. Thus fault isolation is simplified.

Disadvantages

- However, unidirectional traffic can be a disadvantage.
- In a simple ring, a break in the ring (such as a disabled station) can disable the entire network.
- This weakness can be solved by using a dual ring or a switch capable of closing off the break.

Hybrid topology

A combination of any of the above technologies

Fi: ***A hybrid topology: a star backbone with three bus networks***

Tree topology

A tree topology connects multiple star networks to a linear bus backbone cable

Fig: Tree Topology

Topography

- Topography – is the way in which the cables, which provide channels between stations, are positioned.
- For the purposes of cable planning, Topography is more important than Topology.

Fig. A Ring Topology

Fig. A Ring topology wired in a star Topography

Fig. A Ring topology wired in a star Topography

Network models and layered architecture

DCN

Network Models, Layered Architecture, Layer to Layer Communication

OSI and TCP/IP

Example 2.1

Assume Maria and Ann are neighbors with a lot of common ideas. However, Maria speaks only Spanish, and Ann speaks only English. Since both have learned the sign language in their childhood, they enjoy meeting in a cafe a couple of days per week and exchange their ideas using signs. Occasionally, they also use a bilingual dictionary. Communication is face to face and Happens in one layer as shown in Figure 2.1.

Example 2.2

Now assume that Ann has to move to another town because of her job. Before she moves, the two meet for the last time in the same cafe. Although both are sad, Maria surprises Ann when she opens a packet that contains two small machines. The first machine can scan and transform a letter in English to a secret code or vice versa. The other machine can scan and translate a letter in Spanish to the same secret code or vice versa. Ann takes the first machine; Maria keeps the second one. The two friends can still communicate using the secret code, as shown in Figure 2.2.

THE OSI MODEL

Established in 1947, the *International Standards Organization (ISO)* is a multinational body dedicated to worldwide agreement on international standards. Almost three-fourths of countries in the world are represented in the ISO. An ISO standard that covers all aspects of network communications is the *Open Systems Interconnection (OSI)* model. It was first introduced in the late 1970s.

Topics Discussed in the Section

- ✓ Layered Architecture
- ✓ Layer-to-layer Communication
- ✓ Encapsulation
- ✓ Layers in the OSI Model
- ✓ Summary of OSI Layers

NOTE: *ISO is the organization;*
OSI is the model.

7 Layers of OSI Model

Figure: *The OSI model*

OSI Layered communication

Figure 2.4 *OSI layers*

How data is exchanged?

Figure 2.5 *An exchange using the OSI model*

Encapsulation

Note: The physical layer is responsible for moving individual bits from one (node) to the next.

Layers of OSI Model-Functions

Layer 1
Physical Layer

- Physical medium
- Representation of bits.
- Data rate and Transmission rate
- Synchronization of bits.
- Physical topology and Transmission mode.

Layer 2
Data Link Layer

- Framing.
- Flow control.
- Physical addressing.
- Error control.
- Access control.

Layer 3
Network Layer

- Logical addressing.
- Routing.

Layers of OSI Model-Functions (contd..)

Layer 4 Transport Layer

- Service-point addressing.
- Segmentation and reassembly.
- Connection control.
- Flow control.
- Error control.

Layer 5 Session Layer

- Dialog control.
- Synchronization.

Layer 6 Presentation Layer

- Translation.
- Encryption.
- Compression.

Layer 7 Application Layer

- Network virtual terminal.
- File transfer, access, and management (FTAM).
- E-mail services.
- Directory services.

Summary

Application	To allow access to network resources	7
Presentation	To translate, encrypt, and compress data	6
Session	To establish, manage, and terminate sessions	5
Transport	To provide reliable process-to-process message delivery and error recovery	4
Network	To move packets from source to destination; to provide internetworking	3
Data link	To organize bits into frames; to provide hop-to-hop delivery	2
Physical	To transmit bits over a medium; to provide mechanical and electrical specifications	1

Figure 2.6 *Summary of OSI Layers*

TCP/IP PROTOCOL SUITE

The TCP/IP protocol suite was developed prior to the OSI model. Therefore, the layers in the TCP/IP protocol suite do not match exactly with those in the OSI model. The original TCP/IP protocol suite was defined as four software layers built upon the hardware. Today, however, TCP/IP is thought of as a five-layer model with the layers named similarly to the ones in the OSI model. Figure 2.7 shows both configurations.

TCP/IP Layers

Figure *Layers in the TCP/IP Protocol Suite*

TCP/IP PROTOCOL SUITE

Comparison between OSI and TCP/IP

Figure: *TCP/IP and OSI model*

Layers of TCP/IP Protocol Suite

Figure: *A private internet*

TCP/IP-Physical Layer

The unit of communication is a bit.

Figure: *Communication at the physical layer*

TCP/IP-Data Link Layer

The unit of communication is a frame.

Figure: *Communication at the data link layer*

TCP/IP-Network Layer

Figure: *Communication at the network layer*

The unit of communication is a datagram.

TCP/IP-Transport Layer

Figure: *Communication at the transport layer*

The unit of communication is a segment, user datagram, or a packet, depending on the specific protocol used in this layer.

TCP/IP-Application Layer

The unit of communication is a message.

Figure 2.14 *Communication at application layer*

Ethernet Technologies, Data Link layer Essentials

DCN

Wired and Wireless Technologies

WIRED LOCAL AREA NETWORKS

- A local area network (LAN) is a computer network that is designed for a limited geographic area such as a building or a campus.
- Although a LAN can be used as an isolated network to connect computers in an organization for the sole purpose of sharing resources, most LANs today are also linked to a wide area network (WAN) or the Internet.
- The LAN market has seen several technologies such as Ethernet, token ring, token bus, FDDI, and ATM LAN, but Ethernet is by far the dominant technology.

Topics Discussed in the Section

- ✓ IEEE Standards
- ✓ Frame Format
- ✓ Ethernet Addressing
- ✓ Ethernet Evolution
- ✓ Standard Ethernet
- ✓ Fast Ethernet
- ✓ Gigabit Ethernet
- ✓ Ten-Gigabit Ethernet

IEEE Standards-Ethernet

- **Project 802** – set standard for intercommunication among the various manufacturers.
- Adopted by American National Standard Institute (ANSI)
- No replacement to OSI or TCP/IP model just defines functions of the physical layer and the data link layer of LAN protocol

Figure 3.1 *IEEE standard for LANs*

Ethernet Frame

Fields:

Preamble: The first field of the 802.3 frame contains 7 bytes (56 bits) of alternating 0s and 1s that alerts the receiving system to the coming frame and enables it to synchronize its input timing.

SFD: The second field (1 byte: 10101011) signals the beginning of the frame.

Destination Address: contains the physical address of the destination station or stations to receive the packet.

Source Address: contains the physical address of the sender of the packet.

Length/type: Length of the data sent

Data/Padding: This field carries data encapsulated from the upper-layer protocols. It is a minimum of 46 and a maximum of 1500 bytes, as we will see later.

CRC: The last field contains error detection information.

Note: In SFD, if the last 2 bits are 11 and alert the receiver that the next field is the destination address.

Figure 3.2 *Ethernet Frame*

Ethernet Frame Length

Figure 3.3 *Maximum and minimum lengths*

6	6	2	4	=	18

Min length 64-18= 46 bytes
if found less than 46 bytes
PADDING

Ethernet is unreliable as there are no acknowledgment frames received.

Ethernet Addressing

- Each station on an Ethernet network (such as a PC, workstation, or printer) has its own **network interface card (NIC)**
- Referred to as the data link address, physical address, or MAC address.

d: Hexadecimal digit

$$d_1d_2 : d_3d_4 : d_5d_6 : d_7d_8 : d_9d_{10} : d_{11}d_{12}$$

6 bytes = 12 hexadecimal digits = 48 bits

Figure 3.4 *Ethernet address in hexadecimal notation*

4A:30:10:21:10:1A

Number	0	1	2	3	4	5	6	7
Binary	0000	0001	0010	0011	0100	0101	0110	0111
Hexadecimal	0	1	2	3	4	5	6	7

Number	8	9	10	11	12	13	14	15
Binary	1000	1001	1010	1011	1100	1101	1110	1111
Hexadecimal	8	9	A	B	C	D	E	F

Circuit Globe

Ethernet Unicast, Multicast, and Broadcast Addresses

Figure 3.5 *Unicast and multicast addresses*

- A source address is always a unicast address—cause the frame comes from only one station.
- The destination address, however, can be unicast, multicast, or broadcast.

The least significant bit (LSB) of the first byte defines the type of address.
If the bit is 0, the address is unicast; otherwise, it is multicast.

Example 3.1

Identify the type of destination addresses for the following:

a. 4A:30:10:21:10:1A
b. 47:20:1B:2E:08:EE
c. FF:FF:FF:FF:FF:FF

Solution

- To find the type of address, we need to look at the second hexadecimal digit from the left(LSB).
- If it is even, the address is unicast. If it is odd, the address is multicast.
- If all digits are F's, the address is broadcast. Therefore, we have the following:

a. A in binary is 1010 (even) ----- Unicast.
b. 7 in binary is 0111 (odd) ---- Multicast.
c. This is a broadcast address because all digits are F's.

Example 3.2

Show how the address **47:20:1B:2E:08:EE** is sent out on the Ethernet cable

Solution

The address is sent left-to-right, byte by byte; for each byte, it is sent right-to-left, bit by bit, as shown below:

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Example 3.2 (contd..)

Solution

The address is sent left-to-right, byte by byte; for each byte, it is sent right-to-left, bit by bit, as shown below:

47:20:1B:2E:08:EE

Digit	Actual Hex Value	Sent Hex Value
47	0100 0111	11100010
20	0010 0000	00000100
1B	0001 1011	11011000
2E	0010 1110	01110100
08	0000 1000	00010000
EE	0111 0111	01110111

← 11100010 00000100 11011000 01110100 00010000 01110111

Practice examples

Check for multicast and unicast addressing and also represent how they are transmitted over the ethernet cable.

a. 64:2B:13:45:61:13
b. 71:2B:13:45:61:42

Ethernet Evolution

Created by Xerox's Palo Alto Research Center (PARC)

Figure 3.6 *Ethernet evolution through four generations*

Standard Ethernet

Access Method: CSMA/CD

Figure 3.7 *Space/time model of a collision in CSMA*

CSMA/CD

Figure 3.8 *Collision of the first bit in CSMA/CD*

Minimum Frame Size

- Minimum frame size if needed to avoid collision
- Before sending the last bit of a frame, the sender must detect the collision and abort the transmission.
- This is because the sender does not keep a copy of the sender's frame.

$$T_{fr} = 2\,T_p$$

Where T_{fr} is transmission time and T_p is the propagation time

Example 3.3

In the standard Ethernet, if the maximum propagation time is 25.6 μs, what is the minimum size of the frame?

Solution

- The frame transmission time is $T_{fr} = 2 \times T_p = 51.2$ μs.
- This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision.
- The minimum size of the frame is 10 Mbps × 51.2 μs = 512 bits or 64 bytes.
- This is actually the minimum size of the frame for Standard Ethernet, as we discussed before.

CSMA/CD Flow diagram

Figure 3.9 *CSMA/CD flow diagram*

Ethernet Implementation

- The Standard Ethernet defined several implementations.
- In the nomenclature 10Base-X, the number defines the data rate (10 Mbps).
- The term Base means baseband (digital) signal.
- And 'X' approximately defines either the maximum size of the cable is 100 meters (for example 5 for 500 or 2 for 185 meters).
- If the type of the cable is T for unshielded twisted pair cable (UTP) and F for fiber-optic.

Figure 3.10 *Standard Ethernet implementation*

Fast Ethernet Implementation

- Fast Ethernet was designed to compete with LAN protocols such as FDDI or Fiber Channel.
- IEEE created Fast Ethernet under the name 802.3u.
- Fast Ethernet is backward-compatible with Standard Ethernet, but it can transmit data 10 times faster at a rate of 100 Mbps.
- The goals of Fast Ethernet can be summarized as follows:
  1. Upgrade the data rate to 100 Mbps.
  2. Make it compatible with Standard Ethernet.
  3. Keep the same 48-bit address.
  4. Keep the same frame format.
  5. Keep the same minimum and maximum frame lengths.

Figure 3.11 *Fast Ethernet implementation*

Gigabit Ethernet

- The need for an even higher data rate resulted in the design of the Gigabit Ethernet Protocol (1000 Mbps).
- The IEEE committee calls the Standard 802.3z.
- The goals of the Gigabit Ethernet design can be summarized as follows:
  1. Upgrade the data rate to 1 Gbps.
  2. Make it compatible with Standard or Fast Ethernet.
  3. Use the same 48-bit address.
  4. Use the same frame format.
  5. Keep the same minimum and maximum frame lengths.
  6. To support auto-negotiation as defined in Fast Ethernet.

Figure 3.12 *Gigabit Ethernet implementation*

Ten-Gigabit Ethernet

The IEEE committee created Ten-Gigabit Ethernet and called it Standard 802.3ae. The
goals of the Ten-Gigabit Ethernet design can be summarized as follows:

1. Upgrade the data rate to 10 Gbps.
2. Make it compatible with Standard, Fast, and Gigabit Ethernet.
3. Use the same 48-bit address.
4. Use the same frame format.
5. Keep the same minimum and maximum frame lengths.
6. Allow the interconnection of existing LANs into a metropolitan area network (MAN)
or a wide area network (WAN).
7. Make Ethernet compatible with technologies such as Frame Relay and ATM.

Summary

Table 3.1 *Summary of Standard Ethernet implementations*

Characteristics	*10Base5*	*10Base2*	*10Base-T*	*10Base-F*
Medium	Thick coax	Thin coax	2 UTP	2 Fiber
Maximum length	500 m	185 m	100 m	2000 m

Table 3.2 *Summary of Fast Ethernet implementations*

Characteristics	*100Base-TX*	*100Base-FX*	*100Base-T4*
Media	STP	Fiber	UTP
Number of wires	2	2	4
Maximum length	100 m	100 m	100 m

Table 3.3 *Summary of Gigabit Ethernet implementations*

Characteristics	*1000Base-SX*	*1000Base-LX*	*1000Base-CX*	*1000Base-T4*
Media	Fiber short-wave	Fiber long-wave	STP	Cat 5 UTP
Number of wires	2	2	2	4
Maximum length	550 m	5000 m	25 m	100 m

Table 3.4 *Ten-Gigabit Ethernet Implementation*

Characteristics	*10GBase-S*	*10GBase-L*	*10GBase-E*
Media	multi-mode fiber	single-mode fiber	single-mode fiber
Number of wires	2	2	2
Maximum length	300 m	10,000 m	40,000 m

WIRELESS LANS

Wireless communication is one of the fastest growing technologies. The demand for connecting devices without the use of cables is increasing everywhere. Wireless LANs can be found on college campuses, in office buildings, and in many public areas. In this section, we concentrate on two wireless technologies for LANs: IEEE 802.11 wireless LANs, sometimes called wireless Ethernet, and Bluetooth, a technology for small wireless LANs.

Topics Discussed in the Section

- ✓ IEEE 802.11 (WiFi)
- ✓ MAC Sublayer
- ✓ Addressing Mechanism
- ✓ IEEE 802.15 (Bluetooth)

IEEE 802.11 Architecture

The standard defines two kinds of services:

- Basic service set (BSS)
- Extended service set (ESS)

Basic service set (BSS)

- A basic service set is made of stationary or mobile wireless stations and an optional central base station, known as the **access point (AP).**
- The BSS without an AP is a stand-alone network and cannot send data to other BSSs. It is called *ad hoc architecture*.
- *A* BSS with an AP is sometimes referred to as an *infrastructure* network.

Figure 3.13 *Basic service sets (BSSs)*

IEEE 802.11 Architecture (Contd..)

Extended Service Set

- An ESS is made up of two or more BSSs with APs.
- The BSSs are connected through a *distribution system*, which is usually a wired LAN.
- The distribution system connects the APs in the BSSs.
- Uses two types of stations: mobile and stationary.
- The mobile stations are normal stations inside a BSS.
- The stationary stations are AP stations that are part of a wired LAN

Figure 3.14 *Extended service sets (ESSs)*

MAC sublayer

Wireless LANs cannot implement CSMA/CD for three reasons:

1. For collision detection a station must be able to send data and receive collision signals at the same time.

2. Collision may not be detected because of the hidden station problem.

3. The distance between stations can be great.

CSMA/CA

(carrier sense multiple access with collision avoidance)

DIFS: The station waits for a period of time after being idle

SIFS: Waiting a period of time

Figure 3.15 *CSMA/CA flow diagram*

Figure 3.16 *CSMA/CA and Network Allocation Vector (NAV)*

WiFi Frame Format

Figure 3.17 *Frame format*

Table 3.5 *Subfields in FC field*

Field	*Explanation*
Version	Current version is 0
Type	Type of information: management (00), control (01), or data (10)
Subtype	Subtype of each type (see Table 3.6)
To DS	Defined later
From DS	Defined later
More flag	When set to 1, means more fragments
Retry	When set to 1, means retransmitted frame
Pwr mgt	When set to 1, means station is in power management mode
More data	When set to 1, means station has more data to send
WEP	Wired equivalent privacy (encryption implemented)
Rsvd	Reserved

WiFi Frame Format(contd..)

Frame control (FC): The FC field is 2 bytes long and defines the type of frame and some control information.

D: In all frame types except one, this field defines the duration of the transmission that is used to set the value of NAV. In one control frame, this field defines the ID of the frame.

Addresses: There are four address fields, each 6 bytes long. The meaning of each address field depends on the value of the *To DS* and *From DS* subfields.

Sequence control: This field defines the sequence number of the frame to be used in flow control.

Frame body: This field, which can be between 0 and 2312 bytes, contains information based on the type and the subtype defined in the FC field.

FCS: The FCS field is 4 bytes long and contains a CRC-32 error detection sequence.

Table 3.6 *Values of subfields in control frames*

Subtype	*Meaning*
1011	Request to send (RTS)
1100	Clear to send (CTS)
1101	Acknowledgment (ACK)

Frame Types

Management Frame (00)	Control Frame (01)	Data Frame (10)
Initial communication between stations and access points.	Accessing the channel and acknowledging frames.	Carrying data and control information

Figure 3.18 *Control frames*

Hidden Station Problem

Figure 3.19 *Hidden station problem*

Figure 3.20 *Use of handshaking to prevent hidden station problem*

Exposed Station Problem

Figure 3.21 *Exposed station problem*

Exposed Station solution

Figure 3.22 *Use of handshaking in exposed station problem*

Bluetooth

- To connect devices of different functions.
- Forms an ad-hoc network (spontaneous).
- Two network types:
  - Piconet
  - Scatternet

Piconet:

- Can have total of 8 stations with 1 primary and 7 secondary stations.
- All secondary stations are synchronized.
- Communication:
  - one-to-one
  - one-to-many

Figure 3.23 *Piconet*

Bluetooth (contd..)

Scatternet

- Piconets can be combined to form what is called a **scatternet.**
- A secondary station in one piconet can be the primary station in another piconet.
- This station can receive messages from the primary station in the first piconet (as a secondary) and, acting as a primary, deliver them to secondaries in the second piconet.
- A station can be a member of two piconets.

Figure 3.24 *Scatternet*

Bluetooth Frame

N = 240 for 1-slot frame
N = 1490 for 3-slot frame
N = 2740 for 5-slot frame

Figure 3.25 *Frame format types*

Bluetooth Frame Field

❑ **Access code.** This 72-bit field normally contains synchronization bits and the identifier of the primary to distinguish the frame of one piconet from another.

❑ **Header.** This 54-bit field is a repeated 18-bit pattern. Each pattern has the following subfields:

a. Address. The 3-bit address subfield can define up to seven secondaries (1 to 7). If the address is zero, it is used for broadcast communication from the primary to all secondaries.

b. Type. The 4-bit type subfield defines the type of data coming from the upper layers. We discuss these types later.

c. F. This 1-bit subfield is for flow control. When set (1), it indicates that the device is unable to receive more frames (buffer is full).

d. A. This 1-bit subfield is for acknowledgment. Bluetooth uses stop-and-wait ARQ; 1 bit is sufficient for acknowledgment.

e. S. This 1-bit subfield holds a sequence number. Bluetooth uses stop-and-wait ARQ; 1 bit is sufficient for sequence numbering.

f. HEC. The 8-bit header error correction subfield is a checksum to detect errors in each 18-bit header section.

Bluetooth Frame Field

- **Data.** This subfield can be 0 to 2740 bits long. It contains data or control information coming from the upper layers.

  a. The sending station, after sensing that the medium is idle, sends a special small frame called request to send (RTS). In this message, the sender defines the total time it needs the medium.

  b. The receiver acknowledges the request (broadcast to all stations) by sending a small packet called clear to send (CTS).

  c. The sender sends the data frame.

  d. The receiver acknowledges the receipt of data.

Data Link Layer Access Methods

DCN

Data Link Layer- Access method

Contents

- Approaches to sharing transmission medium
- Random Access Protocols-
  - Aloha
  - Slotted Aloha
  - CSMA
  - CSMA/CD

Multiple Access communications

Figure 6.1 Multiple Access Communication

Approaches to sharing transmission medium

- Static Channelization : Partitioning the medium into separate channels and dedicating to particular users.
- Suitable when there is steady stream of information that makes efficient use of dedicated channel.
- Dynamic Medium Access Control : Dynamic sharing of the medium on per packet basis where the user traffic is not continuous.
- Eliminate the incidence of collisions and to achieve reasonable utilization of the medium.

Quality of Service (QoS)

- **Propagation Delay:** Amount of time it takes for the first bit of signal to travel from the sender to the receiver.
  - $T_P = d/c$ where d is the distance between the sender

  and the receiver and c is the speed of light (3 * 10^8 m/s).

- **Transmission Delay:** It is the amount of time required to push all of the packet's bits into the wire. In other words, this is the delay caused by the data-rate of the link.
- Transmission delay is a function of the packet's length and has nothing to do with the distance between the two nodes. This delay is proportional to the packet's length in bits. It is given by the following formula:
  - $T_d = N/R$ where *N* is the number of bits, and *R* is the rate of transmission (say in bits per second)

Problem

d

N

- The stations on a wireless network are a maximum of 300 km apart. If the network transmits 400-bit frames on a shared channel of 200 kbps. Find the propagation and Transmission Delay

R

- Solution
- Propagation Delay : = 300 X 1000 / 3 X 10^8 = 1ms

  $T_p = d/c$

- Transmission delay = 400/(200 X 10^3) = 2ms

  $T_d = N/R$

Data link layer

Data link layer divided into two functionality-oriented sublayers

Taxonomy of multiple-access protocols

RANDOM ACCESS

- In random access or contention methods, no station is superior to another station and none is assigned the control over another.
- No station permits, or does not permit, another station to send.
- At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.

Random Access (Contd..)

Two features give this method its name.

- First, there is no scheduled time for a station to transmit. Transmission is random among the stations. That is why these methods are called *random access.*
- Second, no rules specify which station should send next. Stations compete with one another to access the medium. That is why these methods are also called *contention* methods.
- In a random access method, each station has the right to the medium without being controlled by any other station. However, if more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified

Random Access (Contd..)

- To avoid access conflict or to resolve it when it happens, each station follows a procedure that answers the following questions:
  - When can the station access the medium?
  - What can the station do if the medium is busy?
  - How can the station determine the success or failure of the transmission?
  - What can the station do if there is an access conflict?

Frames in a pure ALOHA network

The idea is that each station sends a frame whenever it has a frame to send. However, since there is only one channel to share, there is the possibility of collision between frames from different stations.

Pure ALOHA network

- The pure ALOHA protocol relies on acknowledgments from the receiver.
- When a station sends a frame, it expects the receiver to send an acknowledgment.
- If the acknowledgment does not arrive after a time-out period, the station assumes that the frame (or the acknowledgment) has been destroyed and resends the frame.
- If all these stations try to resend their frames after the time-out, the frames will collide again.
- Pure ALOHA dictates that when the time-out period passes, each station waits a random amount of time before resending its frame. The randomness will help avoid more collisions. We call this time the **back-off time** T_B.

Procedure for pure ALOHA protocol

Example 12.1

The stations on a wireless ALOHA network are a maximum of 600 km apart. If we assume that signals propagate at 3×10^8 m/s, we find,

$$T_p = (6 \times 10^3) / (3 \times 10^8) = 2 \text{ ms.}$$

K = 2
R = 0,1,2,3

Now we can find the value of T_B for different values of K.

a. For K = 1, the range is {0, 1}. The station needs to generate a random number with a value of 0 or 1. This means that T_B is either 0 ms (0 × 2) or 2 ms (1 × 2), based on the outcome of the random variable.

b. For K = 2, the range is {0, 1, 2, 3}. This means that T_B can be 0, 2, 4, or 6 ms, based on the outcome of the random variable.

c. For K = 3, the range is {0, 1, 2, 3, 4, 5, 6, 7}. This means that T_B can be 0, 2, 4, . . . , 14 ms, based on the outcome of the random variable.

Vulnerable Time

- Vulnerable time is the time in which there is a possibility of collision.
- To compute the vulnerable time the following assumptions are taken into consideration:
  - We assume that the stations send fixed-length frames
  - Each frame takes T_{fr} sec to send (transmission time).

Pure ALOHA vulnerable time = 2 × T_{fr}

Example 12.2

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the requirement to make this frame collision-free?

Solution

Average frame transmission time T_{fr} = 200 bits/200 kbps = 1 ms.

The vulnerable time is 2×1 ms = 2 ms.

This means no station should send later than 1 ms and no station should start sending during the period (1-ms) that this station is sending.

Throughput of Pure ALOHA

Where G is the average number of frames generated by the system during one frame transmission time and **throughput is the average number of successful transmissions**.

Average number of successfully transmitted frames

$S = G \times e^{-2G}$.

The maximum throughput

$S_{max} = 1/(2e) = 0.184$ when G= (1/2).

If we let **ns** to be the number of stations and **nfs** to be the number of frames a station can send per second and T_{fr} is the transmission time then G is given as

$$G = ns \times nfs \times T_{fr}$$

Example

A pure ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

a. 1000 frames per second b. 500 frames per second

c. 250 frames per second.

Solution

The frame transmission time is 200/200 kbps or 1 ms.

a. If the system creates 1000 frames per second, this is 1 frame per millisecond.

The load is 1. In this case

$S = G \times e^{-2G}$ or $S = 0.135$ (13.5 percent).

This means that the throughput is $1000 \times 0.135 = 135$ frames.

Only 135 frames out of 1000 will probably survive.

Example (continued)

b. If the system creates 500 frames per second, this is (1/2) frame per millisecond.
The load is (1/2).
In this case $S = G \times e^{-2G}$ or $S = 0.184$ (18.4 percent).
This means that the throughput is $500 \times 0.184 = 92$,
that only 92 frames out of 500 will probably survive.
Note: that this is the maximum throughput case, percentage-wise.

c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4).
In this case $S = G \times e^{-2G}$ or $S = 0.152$ (15.2 percent).
This means that the throughput is $250 \times 0.152 = 38$.
Only 38 frames out of 250 will probably survive.

Problem

We have a pure ALOHA network with 100 stations. If τ_{fr} = 1 microseconds, what is the number of frames/s each station can send to achieve the maximum efficiency.

Solution:

To achieve the maximum efficiency in pure ALOHA, G = 1/2. If we let **ns** to be the number of stations and **nfs** to be the number of frames a station can send per second.

G = ns × nfs × τ_{fr} = 100 × nfs × 1 μs = 1/2 → nfs = 5000 frames/s

Frames in a slotted ALOHA network

In slotted ALOHA we divide the time into slots of T_{fr} s and force the station to send only at the beginning of the time slot.

Vulnerable time for slotted ALOHA protocol

Slotted ALOHA vulnerable time = T_{fr}

Throughput of slotted ALOHA

The throughput for slotted ALOHA is

$S = G \times e^{-G}$.

The maximum throughput

$S_{max} = 0.368$ when $G = 1$.

Example

A slotted ALOHA network transmits 200-bit frames on a shared channel of 200 kbps. What is the throughput if the system (all stations together) produces

a. 1000 frames per second b. 500 frames per second

c. 250 frames per second.

Solution

The frame transmission time is 200/200 kbps or 1 ms.

a. If the system creates 1000 frames per second, this is 1 frame per millisecond. The load is 1. In this case $S = G \times e^{-G}$ or $S = 0.368$ (36.8 percent). This means that the throughput is $1000 \times 0.0368 = 368$ frames. Only 386 frames out of 1000 will probably survive.

Example (continued)

b. If the system creates 500 frames per second, this is (1/2) frame per millisecond. The load is (1/2). In this case $S = G \times e^{-G}$ or $S = 0.303$ (30.3 percent). This means that the throughput is $500 \times 0.303 = 151$. Only 151 frames out of 500 will probably survive.

c. If the system creates 250 frames per second, this is (1/4) frame per millisecond. The load is (1/4). In this case $S = G \times e^{-G}$ or $S = 0.195$ (19.5 percent). This means that the throughput is $250 \times 0.195 = 49$. Only 49 frames out of 250 will probably survive.

Problem

- **Suppose that the slotted ALOHA protocol is used to share a 56 kbps channel. Suppose that frames are 1000 bits long. Find the maximum throughput of the system in frames/second.**
- Solution:

The maximum throughput for slotted ALOHA is 0.368

Since each frame is 1000 bits long, thus a 56 kbps channel can allow 56 frames to be sent per second.

Thus throughput (frames/second) = 56 X 0.368 = 20.68

Thus approx. 20 frames/sec

Carrier Sense Multiple Access (CSMA)

- The chance of collision can be reduced if a station senses the medium before trying to use it.
- Carrier sense multiple access (CSMA) requires that each station first listen to the medium (or check the state of the medium) before sending.
- In other words, CSMA is based on the principle "sense before transmit" or "listen before talk."
- CSMA can reduce the possibility of collision, but it cannot eliminate it.

Space/time model of the collision in CSMA

Vulnerable time in CSMA

For Station A
Transmission = t1 + Tp

Behavior of three persistence methods

During the channel busy period, three methods are followed to help the station to send data.

a. 1-persistent

b. Nonpersistent

c. p-persistent

1-persistent CSMA

- 1-persistent CSMA (Carrier Sense Multiple Access): To send data, a station first listens to the channel to see if anyone else is transmitting.
- If so, the station waits (keeps sensing it) until the channel becomes idle. Otherwise, it transmits a frame.
- If a collision occurs, the station waits a random amount of time and starts all over again.

a. 1-persistent

Non-persistent CSMA

- To send data, a station first listens to the channel to see if anyone else is transmitting.
- If so, the station waits a random period of time (instead of keeping sensing until the end of the transmission) and repeats the algorithm. Otherwise, it transmits a frame.
- If a collision occurs, the station waits a random amount of time and starts all over again.

b. Nonpersistent

P-Persistent CSMA

- The p-persistent method is used if the channel has time slots with a slot duration equal to or greater than the maximum propagation time.
- In this method, after the station finds the line idle it follows these steps:

1. With probability *p,* the station sends its frame.

2. With probability $q = 1 - p$, the station waits for the beginning of the next time slot and checks the line again.

a. If the line is idle, it goes to step 1.

b. If the line is busy, it acts as though a collision has occurred and uses the backoff procedure.

c. p-persistent

Flow diagram for three persistence methods

a. 1-persistent

b. Nonpersistent

c. p-persistent

CSMA/CD

- Carrier sense multiple access with collision detection (CSMA/CD) augments the algorithm to handle the collision.
- In this method, a station monitors the medium after it sends a frame to see if the transmission was successful.
- If a collision is detected the station aborts the transmission.

Collision of the first bit in CSMA/CD

Collision and abortion in CSMA/CD

Minimum Frame Size

- For *CSMA/CD* to work, we need a restriction on the frame size. Before sending the last bit of the frame, the sending station must detect a collision, if any, and abort the transmission.
- This is so because the station, once the entire frame is sent, does not keep a copy of the frame and does not monitor the line for collision detection.
- In the worst-case scenario, If the two stations involved in a collision are the maximum distance apart, the signal from the first takes time *Tp* to reach the second, and the effect of the collision takes another time *Tp* to reach the first
- Therefore, the frame transmission time T_{fr} must be at least two times the maximum propagation time T_p

$$T_{fr} = 2 \text{ X } T_p$$

Flow diagram for the CSMA/CD

Energy level during transmission, idleness, or collision

Zero Level : Idle state
Normal level: station has successfully captured the channel and is sending its frame.
Abnormal level: collision has occurred and energy is twice the normal energy.

Example

A network using CSMA/CD has a bandwidth of 10 Mbps. If the maximum propagation time (including the delays in the devices and ignoring the time needed to send a jamming signal, as we see later) is 25.6 μs, what is the minimum size of the frame?

Solution

The frame transmission time is $T_{fr} = 2 \times T_p = 51.2$ μs.

This means, in the worst case, a station needs to transmit for a period of 51.2 μs to detect the collision.

The minimum size of the frame is 10 Mbps × 51.2 μs = 512 bits or 64 bytes.

Collision of the first bit in CSMA/CD

In the above figure, the data rate is 10 Mbps, the distance between station A and C is 2000 m, and the propagation speed is 2 x 10^8 m/s. Station A starts sending a long frame at time t1 =0; station C starts sending a long frame at time t2 =3μs. The size of the frame is long enough to guarantee the detection of collision by both stations. Find:

a. The time when station C hears the collision (t3)'

b. The time when station A hears the collision (t4)'

c. The number of bits station A has sent before detecting the collision.

d. The number of bits station C has sent before detecting the collision.

Collision of the first bit in CSMA/CD

We have t1 = 0 and t2 = 3 μs

a. t3 – t1= (2000 m) / (2 × 10^8 m/s) =10 μs → t3 = 10 μs + t1 = 10 μs

b. t4 – t2 = (2000 m) / (2 × 10^8 m/s) =10 μs → t4 = 10 μs + t2 = 13 μs

c. Tfr(A) = t4 – t1 = 13 – 0 = 13 μs → BitsA = 10 Mbps × 13 μs = 130 bits

d. Tfr(C) = t3 – t2 = 10 – 3 = 07μs → BitsC = 10 Mbps × 07 μs = 70 bits.

CSMA/CA

- Carrier sense multiple access with collision avoidance (CSMA/CD) was invented for wireless networks.
- Collisions are avoided through the use of CSAM/CA three strategies:
  - Interframe Space: Waiting period by the station after encountering an idle frame.
  - Contention window: The amount of time divided into slots.
  - Acknowledgements: frame received message

Figure 12.16 *Contention window*

CSMA/CA Flow diagram

CSMA/CA and NAV

Frame Exchange Timeline

Textbooks

- T3 - Behrouz A. Forouzan, Data Communications and Networking, 5th Edition, Tata McGraw Hill, 2013 (chapter 12)

- T4 - Leon Garcia and Widjala, Communication Networks, 5th Edition, Tata McGraw Hill, 2017. (Chapter-6)

Network models and layered architecture

DCN

Addressing, Connecting and Switching

ADDRESSING

Addresses
Physical address
Logical address
Port address
Application-Specific address

ADDRESSING

Figure: *Addresses in the TCP/IP protocol suite*

PHYSICAL ADDRESSES

- Also known as link addresses
- Lowest level of addressing
- Included in the frame used by Data Link Layer
- Size and format of the address vary depending on the network.
  - Ex. Ethernet – 6 Bytes (48-bits) imprinted on network interfacing card (NIC)
  - Local Talk (Apple) – 1 Byte dynamic addressing (that changes each time the station comes up

Example 2.3

In Figure 2.16 a node with physical address 10 sends a frame to a node with physical address 87. The two nodes are connected by a link (a LAN). At the data link layer, this frame contains physical (link) addresses in the header. These are the only addresses needed. The rest of the header contains other information needed at this level. As the figure shows, the computer with physical address 10 is the sender, and the computer with physical address 87 is the receiver. The data link layer at the sender receives data from an upper layer. It encapsulates the data in a frame. The frame is propagated through the LAN. Each station with a physical address other than 87 drops the frame because the destination address in the frame does not match its own physical address. The intended destination computer, however, finds a match between the destination address in the frame and its own physical address.

Example 2.3

Figure 2.16 *Example 2.3: physical addresses*

Physical addressing can be:

- Unicast(single recipient),
- Multicast (group of recipients) and
- Broadcast(all system)

Example 2.4

As we will see in next few classes, most local area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below:

07:01:02:01:2C:4B

A 6-byte (12 hexadecimal digits) physical address

LOGICAL ADDRESSES

- Required for universal communications which are independent of underlying physical networks.
- Each host are identified uniquely regardless of physical network.
- Currently 32-bits Logical address are need over the Internet.
- No two publicly addressed and visible hosts on the Internet can have the same IP address.

Example 2.5

Figure 2.17 shows a part of internet with two routers connecting three LANs. Each device (computer or router) has a pair of addresses (logical and physical) for each connection. In this case, each computer is connected to only one link and therefore has only one pair of addresses. Each router, however, is connected to three networks. So each router has three pairs of addresses, one for each connection. Although it may be obvious that each router must have a separate physical address for each connection, it may not be obvious why it needs a logical address for each connection. We discuss these issues in routing classes. The computer with logical address A and physical address 10 needs to send a packet to the computer with logical address P and physical address 95. We use letters to show the logical addresses and numbers for physical addresses, but note that both are actually numbers, as we will see in later chapters.

Example 2.5

The physical addresses will change from hop to hop, but the logical addresses remain the same.

Figure: *Example 2.5: logical addresses*

numbers: Physical address , letters: logical address

PORT ADDRESSES

- Physical and Logical addresses are needed for data travel from source to destination.
- Port addressing is required for process to process communication.

TCP/IP Port Addresses – 16-bits in length

Example 2.6

Figure 2.18 shows two computers communicating via the Internet. The sending computer is running three processes at this time with port addresses a, b, and c. The receiving computer is running two processes at this time with port addresses j and k. Process a in the sending computer needs to communicate with process j in the receiving computer.

Note that although both computers are using the same application, FTP, for example, the port addresses are different because one is a client program and the other is a server program.

Example 2.6

Figure 2.18 *Example 2.6: port numbers*

The physical addresses change from hop to hop, but the logical and port addresses usually remain the same.

Example 2.7

As we will see in future chapters, a port address is a 16-bit address represented by one decimal number as shown.

753

A 16-bit port address represented as one single number

APPLICATION ADDRESSES

- User friendly addresses
- Ex: e-mail: network@gmail.com - user recipient
- Ex. URL: www.network.com – searching WWW

CONNECTING DEVICES

- To connect LANs and WANs together we use connecting devices.
- Connecting devices can operate in different layers of the Internet model.
- Three kinds of connecting devices:
  - **repeaters (or hubs),**
  - **bridges (or two-layer switches), and**
  - **routers (or three-layer switches).**

Figure: *Connecting devices*

Repeaters

- Works only in physical layer
- Some signals get distorted before it reaches the destination leading data loss.
- To avoid signal fading, repeaters receives signal before it comes weak and regenerates and retimes the original bit pattern.
- Later repeater send refreshed signal.
- Earlier: Ethernet LAN used Bus topology with repeaters now they use Star topology with **hub.**

Repeaters

Figure *Repeater or hub*

Note

A repeater forwards every bit; it has no filtering capability.

Bridge

- Works in physical and data link layers
- Regenerates the signal
- Can check the MAC addresses contained in frames
- Has filtering capability
- Can check the destination address of a frame and can decide from which ongoing port the frame should be send out

Bridge has a table used in filtering decision

Bridge

Bridge table

Address	Port
71:2B:13:45:61:41	1
71:2B:13:45:61:42	2
64:2B:13:45:61:12	3
64:2B:13:45:61:13	4

Figure A *Bridge*

Bridge does not change the physical (MAC) addresses in a frame.

Transparent Bridge

- Stations are unaware of existence of bridge
- No reconfiguration is required if the bridge goes down.
- According to 802.1d, system having transparent bridge should posses following three criteria:
  - Frames must be forwarded from one station to the another.
  - Forwarding table is automatically made by learning frame movements in the network
  - Loops in the system must be prevented

Learning Bridge

- Earlier bridge had forwarding table that were static.
- System administrator would manually enter each entry during the bridge set-up
- But it was not practical as addition and deleting of station would lead to manual modification
- Solution was to have dynamic table that maps the addresses to port automatically.
- For this we need to make bridge to learn the frame movement automatically using forward look-up decision table.
- So bridge inspects both source and destination addressing

Learning Bridge

Gradual building of Table

Address	Port

a. Original

Address	Port
71:2B:13:45:61:41	1

b. After A sends a frame to D

Address	Port
71:2B:13:45:61:41	1
64:2B:13:45:61:13	4

c. After D sends a frame to B

Address	Port
71:2B:13:45:61:41	1
64:2B:13:45:61:13	4
71:2B:13:45:61:42	2

d. After B sends a frame to A

Address	Port
71:2B:13:45:61:41	1
64:2B:13:45:61:13	4
71:2B:13:45:61:42	2
64:2B:13:45:61:12	3

e. After C sends a frame to D

Figure 3.43 *Learning bridge*

Bridge

Note

A bridge has a table used in filtering decisions.

A bridge does not change the physical (MAC) addresses in a frame.

Router

- Three layer device: operates on PL, DL and NL
- As PL device: regenerates the signal it receives
- As DL device: check physical address in each frame
- As NL device: check for network layer addressing
- NOTE:
  - Bridge changes collision domains
  - Route limits broadcast domains

- Can connect LANs, LAN-WAN and WANs together

Router is an internetworking device; it connects independent networks together.

Router

There are three major differences between a router and a repeater or a bridge.

1. A router has a physical and logical (IP) address for each of its interfaces.

2. A router acts only on those packets in which the physical destination address matches the address of the interface at which the packet arrives.

3. A router changes the physical address of the packet (both source and destination) when it forwards the packet.

Router

Figure 3.44 *Routing example*

Router is three layer switch: receive, process and dispatch a packet

Router

Note

A router is a three-layer (physical, data link, and network) device.

A repeater or a bridge connects segments of a LAN.
A router connects independent LANs or WANs to create an internetwork (internet).

A router changes the physical addresses in a packet.

SWITCHING

- When a message reaches a connecting device, a decision needs to be made to select one of the output ports through which the packet needs to be send out.
- In other words, the connecting device acts as a switch that connects one port to another port.

✓ Circuit Switching

✓ Packet Switching

CIRCUIT SWITHING

- A physical circuit is established between source and destination before delivery of message
- Once circuit is established, entire message is transformed from source and destination
- The source can then inform the network that the transmission is complete, which allows the network to open all switches and use the links and connecting devices for another connection.
- The circuit switching was never implemented at the network layer; it is mostly used at the physical layer.

Note

In circuit switching, the whole message is sent from the source to the destination without being divided into packets.

CIRCUIT SWITHING

Example 4.1

A good example of a circuit-switched network is the early telephone systems in which the path was established between a caller and a callee when the telephone number of the callee was dialed by the caller. When the callee responded to the call, the circuit was established. The voice message could now flow between the two parties, in both directions, while all of the connecting devices maintained the circuit. When the caller or callee hung up, the circuit was disconnected. The telephone network is not totally a circuit-switched network today.

PACKET SWITHING

- A message from the upper layer is divided into manageable packets and each packet is sent through the network.
- The source of the message sends the packets one by one; the destination of the message receives the packets one by one.
- The destination waits for all packets belonging to the same message to arrive before delivering the message to the upper layer.
- The connecting devices in a packet-switching network still need to decide how to route the packets to the final destination.

In packet switching, the message is first divided into manageable packets at the source before being transmitted. The packets are assembled at the destination.

PACKET SWITHING

- Packet at the source is divided into manageable packets, normally called **datagrams.**
- Individual datagrams are then transferred from the source to the destination.
- The received datagrams are assembled at the destination before recreating the original message.
- The packet-switched network layer of the Internet was originally designed as a connectionless service, but recently there is a tendency to change this to a connection-oriented service.

✓ Connectionless Service

✓ Connection-Oriented Service

Connectionless Service

Figure 4.3 *A connectionless packet-switched network*

Connectionless Service (contd..)

Figure 4.4 *Forwarding process in a connectionless network*

In a connectionless packet-switched network, the forwarding decision is based on the destination address of the packet.

Connectionless Service (contd..)

Figure 4.5 *Delay in a connectionless network*

Connection Oriented Service

Figure 4.6 *A connection-oriented packet switched network*

Connection Oriented Service

Note

In a connection-oriented packet switched network, the forwarding decision is based on the label of the packet.

Connection Oriented Service

Figure 4.7 *Forwarding process in a connection-oriented network*

Connection Oriented Service

Phases:

1. Setup Phase
2. Request Phase
3. Acknowledgement Phase
4. Data Transfer Phase
5. Teardown Phase

1.Setup Phase

- A router creates an entry for a virtual circuit.
- For example, suppose source A needs to create a virtual circuit to destination B.
- Two auxiliary packets need to be exchanged between the sender and the receiver:
  - the request packet and
  - the acknowledgment packet.

2. Request Phase

A request packet is sent from the source to the destination.

1. Source A sends a request packet to router R1.

2. Router R1 receives the request packet. It knows that a packet going from A to B goes out through port 3. For the moment, assume that it knows the output port. The router creates an entry in its table for this virtual circuit, but it is only able to fill three of the four columns. The router assigns the incoming port (1) and chooses an available incoming label (14) and the outgoing port (3). It does not yet know the outgoing label, which will be found during the acknowledgment step. The router then forwards the packet through port 3 to router R3.

3. Router R3 receives the setup request packet. The same events happen here as at router R1; three columns of the table are completed: in this case, incoming port(1), incoming label (66), and outgoing port (2).

4. Router R4 receives the setup request packet. Again, three columns are completed: incoming port (2), incoming label (22), and outgoing port (3).

5. Destination B receives the setup packet, and if it is ready to receive packets from A, it assigns a label to the incoming packets that come from A, in this case 77. This label lets the destination know that the packets come from A, and not other sources.

2. Request Phase

Figure 4.8 *Sending request packet in a virtual-circuit network*

3. Acknowledgement Phase

Figure 4.9 *Setup acknowledgement in a virtual-circuit network*

4. Data Transfer Phase

Figure 4.10 *Flow of one packet in an established virtual circuit*

5. Teardown Phase

- In the **teardown phase,** source A, after sending all packets to B, sends a special packet called a *teardown packet*.
- Destination B responds with a *confirmation* packet.
- All routers delete the corresponding entry from their tables.

DELAY IN CONNECTION-ORIENTED NETWORK

Figure 4.11 *Delay in a connection-oriented network*

Introduction to Network Layer- Network Service provided and issues

Department of DSCA, MIT, MAHE

DCN

Introduction to Network Layer- Network Service provided and issues

Topics Discussed

- ✓ Logical Addressing
- ✓ Services Provided at the Source Computer
- ✓ Services Provides at the Each Router
- ✓ Services Provided at the Destination Computer

Logical Addressing

Figure 4.12 *An imaginary part of the Internet*

Service Provider at the Source Computer

Figure 4.13 *Services provided at the source computer*

Service Provider at the Source Computer

- Packetizing
- Finding logical address of next-hop
- Finding MAC address of next-hop
- Fragmentation

Service Provider at the Source Computer

- **Packetizing**

  - Network layer encapsulates the data coming from the upper layer into a datagram.
  - Encapsulation is done by adding a header to the data that contains the logic source and destination address of the packet, information about fragmentation, the protocol ID of the protocol that has requested the service, the data length, and possibly some options.
  - The network layer also includes a checksum that is calculated only over the datagram header.

Note:
The upper layer protocol only provides the destination logical address; the logical source address comes from the network layer itself

Service Provider at the Source Computer

- **Finding logical address of next-hop**
  - The prepared datagram contains the source and destination addresses of the packet.
  - The datagram may have to pass through many networks to reach its final destination.
  - If the destination computer is not connected to the same network as the source, the datagram should be delivered to the next router.
  - The source and destination address in the datagram does not tell anything about the logical address of the next hop.
  - The network layer at the source computer needs to consult a routing table to find the logical address of the next hop.

Service Provider at the Source Computer

- **Finding MAC address of next-hop**
  - The network layer does not actually deliver the datagram to the next hop; it is the duty of the data link layer to do the data delivery.
  - The data link layer needs the MAC to address the next-hop to do the delivery.
  - To find the MAC address of the next-hop, the network layer could use another table to map the next-hop logical address to the MAC address.
  - This task has been assigned to another auxiliary protocol called Address Resolution Protocol (ARP) that finds the MAC address of the next-hop given the logical address.

Service Provider at the Source Computer

- **Fragmentation**
  - Most LANs and WANs have a limit on the size of the data to be carried in a frame (MTU).
  - The datagram prepared at the network layer, may be larger than that limit.
  - Thus datagram needs to be fragmented into smaller units before being passed to the data link layer.
  - Fragmentation needs to preserve the information at the header of the datagram.
  - In other words, although the data can be fragmented, the header needs to be repeated.
  - In addition, some more information needs to be added to the header to define the position of the fragment in the whole datagram.

Service Provider at the each Router

Figure 4.14 *Processing at each router*

Service Provider at the Destination

Figure 4.15 *Processing at the destination computer*

OTHER SERVICES

In this section we introduce some issues related to the network layer.

- ✓ Error Control
- ✓ Flow Control
- ✓ Congestion Control
- ✓ Routing
- ✓ Security

Error Control

- **Error control** means including a mechanism for detecting corrupted, lost, or duplicate datagrams.
- Error control also includes a mechanism for correcting errors after they have been detected.
- Basically error control is not provided exactly at the network layer, it is done by physical layer.
- The data link layer that controls the behavior of these networks (LANs or WANs) use error control.
- If a hop-to-hop error control is already implemented at the data link layer, why do we need error control at the network layer?
- Some areas in the path of the datagram that some errors may occur, but never checked; the error control at the data link layer can miss any error that occurs when the datagram is being processed by the router.

Error Control

Figure 4.16 *Error checking at the data link layer*

Flow Control

- **Flow control** regulates the amount of data a source can send without overwhelming the receiver.
- If the upper layer at the source computer produces data faster than the upper layer at the destination computer can consume it, the receiver will be overwhelmed with data.
- To control the flow of data, the receiver needs to send some feedback to the sender to inform the latter it is overwhelmed with data.

Congestion Control

- ***Congestion*** in the network layer is a situation in which too many datagrams are present in an area of the Internet.
- Congestion may occur if the number of datagrams sent by source computers is beyond the capacity of the network or routers.
- In this situation, some routers may drop some of the datagrams.
- However, as more datagrams are dropped, the situation may become worse because, due to the error control mechanism at the upper layers, the sender may send duplicates of the lost packets.
- If the congestion continues, sometimes a situation may reach a point that collapses the system and no datagram is delivered.

Quality of Service

- Internet has allowed new applications such as multimedia communication (in particular real-time communication of audio and video), the **quality of service (QoS)** of the communication has become more and more important.

- The Internet has thrived to provide a better quality of service to support these applications. However, to keep the network layer untouched, these provisions are mostly implemented in the upper layer.

Routing

- A router creates its routing table to help in forwarding a datagram in a connectionless service or helps in creating a virtual circuit, during the setup phase, in a connection-oriented service.
- This can be done by *routing protocols*, that help hosts and routers make their routing tables, maintain them, and update them.
- These are separate protocols that sometimes use the service of the network layer and sometimes the service of some transport layer protocols to help the network layer do its job.

Security

- The network layer was designed with no security provision.
- Today, however, security is a big concern. To provide security for a connectionless network layer, we need to have another virtual level that changes the connectionless service to a connection-oriented service.
- This virtual layer, called IPSec

Classful Addressing- binary, decimal,
256 base number representation and operations

DCN

Classful Addressing

Introduction

- **The identifier used in the IP layer of the TCP/IP protocol suite to identify each device connected to the Internet is called the Internet address or IP address.**

- **An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet; an IP address is the address of the interface.**

  - ✓ **Notation**
  - ✓ **Range of Addresses**
  - ✓ **Operations**

Address Space

- A protocol like IPv4 that defines addresses has an **address space.**
- An address space is the total number of addresses used by the protocol.
- If a protocol uses *b* bits to define an address, the address space is 2^*b* because each bit can have two different values (0 or 1).
- IPv4 uses 32-bit addresses, which means that the address space is 2^32 or 4,294,967,296

The address space of IPv4 is 2^{32} or 4,294,967,296.

Notation

- There are three common notations to show an IPv4 address:
  - binary notation (base 2),
  - dotted-decimal notation (base 256), and
  - hexadecimal notation (base 16).
- The most prevalent, however, is base 256.

Numbers in base 2, 16, and 256.

Binary Notation: Base 2

- In **binary notation,** an IPv4 address is displayed as 32 bits.
- To make the address more readable, one or more spaces is usually inserted between each octet (8 bits).
- Each octet is often referred to as a byte.
- So it is common to hear an IPv4 address referred to as a 32-bit address, a 4-octet address, or a 4-byte address.

01110101 10010101 00011101 11101010

Dotted-Decimal Notation: Base 256

- To make the IPv4 address more compact and easier to read, an IPv4 address is usually written in decimal form with a decimal point (dot) separating the bytes.
- This format is referred to as **dotted-decimal notation**.

Figure 5.1 ***Dotted-decimal notation***

Note: As each byte (octet) is only 8 bits, each number in the dotted-decimal notation is between 0 and 255.

255.255.255.255

Converting binary to dotted decimal

10000000 00001011 00000011 00011111

7 6 5 4 3 2 1 0
10000000

= 2^7
= 128

00001011

= 2^3 + 2^1 + 2^0
= 8+2+1
= 11

00000011

= 2^1 + 2^0 = 2+1
= 3

00011111

= 2^4 + 2^3 + 2^2+2^1+2^0
= 16 + 8 + 4 + 2 + 1
= 31

Example 5.1

Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111
c. 11100111 11011011 10001011 01101111
d. 11111001 10011011 11111011 00001111

Solution

We replace each group of 8 bits with its equivalent decimal number and add dots for separation:

a. 129.11.11.239
b. 193.131.27.255
c. 231.219.139.111
d. 249.155.251.15

Converting dotted decimal to binary

154.31.16.13

1. Question: Can I subtract 128 from 154? Answer: YES. So we assign 1 to 128.

128	64	32	16	8	4	2	1
1							

2. Question: Can I subtract 64 from 26? Answer: NO. So we assign 0 to 64.

128	**64**	32	16	8	4	2	1
1	**0**						

3. Question: Can I subtract 32 from 26? Answer: NO. So we assign 0 to 32.

128	64	**32**	16	8	4	2	1
1	0	**0**					

4. Question: Can I subtract 16 from 26? Answer: YES. So we assign 1 to 16.

128	64	32	**16**	8	4	2	1
1	0	0	**1**				

5. That will give us a remainder of 10. (26-16=10). Question: Can I subtract 8 from 10? Answer: YES. So we assign 1 to 8.

128	64	32	16	**8**	4	2	1
1	0	0	1	**1**			

Converting dotted decimal to binary

154.31.16.13

6. That will give us a remainder of 2. (10-8=2). Question: Can I subtract 4 from 2? Answer: NO. So we assign 0 to 4.

128	64	32	16	8	**4**	2	1
1	0	0	1	1	**0**		

7. Question: can I subtract 2 from 2? Answer: YES. So we assign 1 to 2.

128	64	32	16	8	4	**2**	1
1	0	0	1	1	0	**1**	

8. That will give us a remainder of 0. So for the rest of the values in our row, we can assign 0.

128	64	32	16	8	4	2	**1**
1	0	0	1	1	0	1	**0**

So now we know that a decimal number 154 is **10011010** converted to binary form. To double check, we take the values assigned with 1 and add them together: **128+16+8+2=154**

Src: https://petri.com/csc_convert_ip_address_from_decimal_to_binary

154.31.16.13

		b =							
		7	6	5	4	3	2	1	0
		128	64	32	16	8	4	2	1
a=31	31-16=15	0	0	0	1				
	15-8 = 7	0	0	0	1	1			
	7-4 = 3	0	0	0	1	1	1		
	03-02=1	0	0	0	1	1	1	1	
	1-1 = 0	0	0	0	1	1	1	1	1

a=16	16-16=0	0	0	0	1	0	0	0	0

a=13	13-8 = 5	0	0	0	0	1			
	5-4 = 1	0	0	0	0	1	1		
	1-2=-1	0	0	0	0	1	1	0	
	1-1=0	0	0	0	0	1	1	0	1

10011010 00011111 00010000 00001101

Example 5.2

Change the following IPv4 addresses from dotted-decimal notation to binary notation.

a. **111.56.45.78**
b. **221.34.7.82**
c. ***241.8.56.12***
d. ***75.45.34.78***

Solution

We replace each decimal number with its binary equivalent:

a. **01101111 00111000 00101101 01001110**
b. **11011101 00100010 00000111 01010010**
c. **11110001 00001000 00111000 00001100**
d. **01001011 00101101 00100010 01001110**

Example 5.3

Find the error, if any, in the following IPv4 addresses:

a. **111.56.045.78**
b. **221.34.7.8.20**
c. **75.45.301.14**
d. **11100010.23.14.67**

Solution

a. **There should be no leading zeroes (045).**
b. **We may not have more than 4 bytes in an IPv4 address.**
c. **Each byte should be less than or equal to 255.**
d. **A mixture of binary notation and dotted-decimal notation**

Hexadecimal Notation: Base 16

Each hexadecimal digit is equivalent to four bits. This means that a 32-bit address has 8 hexadecimal digits.

01110101 10010101 00011101 11101010

0111 0101 = 75

1001 0101 = 95

0001 1101 = 1D

1110 1010 = EA

0X75951DEA

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Example 5.4

Change the following IPv4 addresses from binary notation to hexadecimal notation.

a. **10000001 00001011 00001011 11101111**

b. **11000001 10000011 00011011 11111111**

Solution

We replace each group of 4 bits with its hexadecimal equivalent. Note that 0X (or 0x) is added at the beginning or the subscript 16 at the end.

a. 0X810B0BEF or $810B0BEF_{16}$

b. 0XC1831BFF or $C1831BFF_{16}$

Range of Addresses

- Sometimes we need to find the number of addresses in a range if the first and last address is given.
- Other times, we need to find the last address if the first address and the number of addresses in the range are given.
- In this case, we can perform subtraction or addition operations in the corresponding base (2, 256, or 16).
- Alternatively, we can covert the addresses to decimal values (base 10) and perform operations in this base.

Example 5.5

Find the number of addresses in a range if the first address is 146.102.29.0 and the last address is 146.102.32.255.

Solution

- **We can subtract the first address from the last address in base 256 (see Appendix B).**
- **The result is 0.0.3.255 in this base.**
- **To find the number of addresses in the range (in decimal), we convert this number to base 10 and add 1 to the result.**

$$\text{Number of addresses} = (0 \times 256^3 + 0 \times 256^2 + 3 \times 256^1 + 255 \times 256^0) + 1 = 1024$$

Example 5.6

The first address in a range of addresses is 14.11.45.96. If the number of addresses in the range is 32, what is the last address?

Solution

- **We convert the number of addresses minus 1 to base 256, which is 0.0.0.31.**
- **We then add it to the first address to get the last address. Addition is in base 256.**

$$\text{Last address} = (14.11.45.96 + 0.0.0.31)_{256} = 14.11.45.127$$

Operations

- ✓ ***Bitwise NOT operation***
- ✓ ***Bitwise AND operation***
- ✓ ***Bitwise OR operation***

Bitwise NOT operation

- Unary operation
- Every 0 is changed to 1 and 1 to 0 in 32-bit binary format

NOT operation

Input	Output
0	1
1	0

Operation for each bit

Figure 5.2 ***Bitwise NOT operation***

Example 5.7

The following shows how we can apply the NOT operation on a 32-bit number in binary.

Original number:	00010001	01111001	00001110	00100011
Complement:	11101110	10000110	11110001	11011100

We can use the same operation using the dotted-decimal representation and the short cut.

Original number:	17	.	121	.	14	.	35
Complement:	238	.	134	.	241	.	220

255-17=238

255-121=134

255-14=241

255-35=220

Bitwise AND operation

- Binary operation with two input
- Compares corresponding bits and selects the smaller bit from the two

AND

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Operation for each bit

Figure 5.3 ***Bitwise AND operation***

Example 5.8

The following shows how we can apply the AND operation on two 32-bit numbers in binary.

First number:	00010001	01111001	00001110	00100011
Second number:	11111111	11111111	10001100	00000000
Result	00010001	01111001	00001100	00000000

We can use the same operation using the dotted-decimal representation and the short cut.

First number:	17	.	121	.	14	.	35
Second number:	255	.	255	.	140	.	0
Result:	17	.	121	.	12	.	0

Example 5.8

- Although we can directly use the AND operation on the 32-bit binary representation of two numbers, when the numbers are represented in dotted-decimal notation, we can use two short cuts.

- **1.** When at least one of the numbers is 0 or 255, the AND operation selects the smaller byte (or one of them if equal).

- **2.** When none of the two bytes is either 0 or 255, we can write each byte as the sum of eight terms, where each term is a power of 2. We then select the smaller term in each pair (or one of them if equal) and add them to get the result.

We have applied the first short cut on the first, second, and the fourth byte; we have applied the second short cut on the third byte. We have written 14 and 140 as the sum of terms and selected the smaller term in each pair as shown below.

Powers	2^7		2^6		2^5		2^4		2^3		2^2		2^1		2^0
Byte (14)	0	+	0	+	0	+	0	+	8	+	4	+	2	+	0
Byte (140)	128	+	0	+	0	+	0	+	8	+	4	+	0	+	0
Result (12)	0	+	0	+	0	+	0	+	8	+	4	+	0	+	0

Bitwise OR operation

- The bitwise OR operation is a binary operation; it takes two inputs.
- The OR operation compares the corresponding bits in the two numbers and selects the larger bit from the two (or one of them if equal).

OR

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

Operation for each bit

Figure 5.4 ***Bitwise OR operation***

Example 5.9

- Although we can directly use the OR operation on the 32-bit binary representation of the two numbers, when the numbers are represented in dotted-decimal notation, we can use two short cuts.
- **1.** When at least one of the two bytes is 0 or 255, the OR operation selects the larger byte (or one of them if equal).
- **2.** When none of the two bytes is 0 or 255, we can write each byte as the sum of eight terms, where each term is a power of 2. We then select the larger term in each pair (or one of them if equal) and add them to get the result of OR operation.

The following shows how we can apply the OR operation on two 32-bit numbers in binary.

First number:	00010001	01111001	00001110	00100011
Second number:	11111111	11111111	10001100	00000000
Result	11111111	11111111	10001110	00100011

We can use the same operation using the dotted-decimal representation and the short cut.

First number:	17	.	121	.	14	.	35
Second number:	255	.	255	.	140	.	0
Result:	255	.	255	.	142	.	35

We have used the first short cut for the first and second bytes and the second short cut for the third byte.

Address Blocks and 2-level addressing, Extracting Block Information.

DCN

Address Blocks and 2-level addressing, Extracting Block Information

Topics

- Classes
- Classes and Blocks
- Two-Level Addressing
- Extracting Block Information

Classfull Addressing

Class A: 2^{31} = 2,147,483,648 addresses, 50%

Class B: 2^{30} = 1,073,741,824 addresses, 25%

Class C: 2^{29} = 536,870,912 addresses, 12.5%

Class D: 2^{28} = 268,435,456 addresses, 6.25%

Class E: 2^{28} = 268,435,456 addresses, 6.25%

Figure 5.5 ***Occupation of address space***

Recognizing Classes

- Address class be identified by either binary or dotted-decimal notation.
- In the binary notation, the first few bits can immediately tell us the
- class of the address.
- In the dotted-decimal notation, the value of the first byte can give the class of an address

Figure 5.6 ***Finding the class of address***

How Computer Knows the IP address

Computers often store IPv4 addresses in binary notation. In this case, it is very convenient to write an algorithm to use a continuous checking process for finding the address as shown below

Figure 5.7 ***Finding the class of an address using continuous checking***

Example 5.10

Find the class of each address:

a. **00000001 00001011 00001011 11101111**
b. **11000001 10000011 00011011 11111111**
c. **10100111 11011011 10001011 01101111**
d. **11110011 10011011 11111011 00001111**

Solution

See the procedure in Figure 5.7.

a. The first bit is 0. This is a class A address.
b. The first 2 bits are 1; the third bit is 0. This is a class C address.
c. The first bit is 1; the second bit is 0. This is a class B address.
d. The first 4 bits are 1s. This is a class E address.

Example 5.11

Find the class of each address:

a. **227.12.14.87**

b. **193.14.56.22**

c. **14.23.120.8**

d. **252.5.15.111**

Solution

a. **The first byte is 227 (between 224 and 239); the class is D.**

b. **The first byte is 193 (between 192 and 223); the class is C.**

c. **The first byte is 14 (between 0 and 127); the class is A.**

d. **The first byte is 252 (between 240 and 255); the class is E.**

Netid and Hostid

- In classful addressing, an IP address in classes A, B, and C is divided into **netid** and **hostid.**
- These parts are of varying lengths, depending on the class of the address.
- Figure below shows the netid and hostid bytes.
- **Note** that classes D and E are not divided into netid and hostid,

Figure 5.8 ***Netid and hostid***

Classes and Blocks

CLASS-A

- Since only 1 byte in class A defines the netid and the left most bit should be 0, the next 7 bits can be changed to find the number of blocks in this class.
- Therefore, class A is divided into 2^7 = 128 blocks that can be assigned to 128 organizations.
- However, each block in this class contains 16,777,216 addresses, which means the organization should be a really large one to use all these addresses.

Figure 5.9 ***Blocks in Class A***

Millions of class A addresses are wasted.

Classes and Blocks (contd)..

CLASS-B

- Since 2 bytes in class B define the netid and the two leftmost bit should be 10 (fixed), the next 14 bits can be changed to find the number of blocks in this class.
- Therefore, class B is divided into 2^14 = 16,384 blocks that can be assigned to 16,384 organizations.
- However, each block in this class contains 65,536 addresses. Not so many organizations can use so many addresses.

Figure 5.10 ***Blocks in Class B***

Many class B addresses are wasted.

Classes and Blocks (contd)..

CLASS-C

- Since 3 bytes in class C define the class and the three leftmost bits should be 110 (fixed), the next 21 bits can be changed to find the number of blocks in this class.
- Therefore, class C is divided into 2^21 = 2,097,152 blocks, in which each block contains 256 addresses, that can be assigned to 2,097,152 organizations.
- Each block contains 256 addresses.

Figure 5.11 ***Blocks in Class C***

Classes and Blocks (contd)..

CLASS-D

- There is just one block of class D addresses.
- It is designed for multicasting.
- Each address in this class is used to define one group of hosts on the Internet.
- When a group is assigned an address in this class, every host that is a member of this group will have a multicast address in addition to its normal (unicast) address.

Figure 5.12 ***The single block in Class D***

Class D addresses are made of one block, used for multicasting.

Classes and Blocks (contd)..

CLASS-E

There is just one block of class E addresses. It was designed for use as reserved addresses.

Figure 5.13 ***The single block in Class E***

The only block of class E addresses was reserved for future purposes

Two-Level Addressing

- The whole purpose of IPv4 addressing is to define a destination for an Internet packet (at the network layer).
- When classful addressing was designed, it was assumed that the whole Internet is divided into many networks and each network connects many hosts ---- Internet
- A network was normally created by an organization that wanted to be connected to the Internet.
- The Internet authorities allocated a block of addresses to the organization (in class A, B, or C).
- Since all addresses in a network belonged to a single block, each address in classful addressing contains two parts: netid and hostid.
- The **netid** defines the network; the **hostid** defines a particular host connected to that network.

Class A: $n = 8$
Class B: $n = 16$
Class C: $n = 24$

Figure 5.14 *Two-level addressing in classful addressing*

Example 5.12

Two-level addressing can be found in other communication systems. For example, a telephone system inside the India: area code and local part. The area code defines the area, the local part defines a particular telephone subscriber in that area.

Extracting Information in a Block

Examples:
12.24.76.8/8
23.14.67.92/12
220.8.24.255/24

- A block is a range of addresses. Given any address in the block, we normally like to know three pieces of information about the block:
  - the number of addresses,
  - the first address, and
  - the last address.
- To extract these pieces of information, we need to know the class of the address.
- After the class of the block is found, we know the value of *n*, the length of netid in bits.

1. The number of addresses in the block, *N*, can be found using $N = 2^{(32-n)}$.
2. To find the first address, we keep the n leftmost bits and set the (32− n) rightmost bits all to 0s.
3. To find the last address, we keep the n leftmost bits and set the (32− n) rightmost bits all to 1s.

Extracting Information in a Block

Class A: $n = 8$

Class B: $n = 16$

Class C: $n = 24$

Figure 5.15 ***Information extraction in classful addressing***

Example 5.13

An address in a block is given as 73.22.17.25. Find the number of addresses in the block, the first address, and the last address.

Solution

Since 73 belongs to class A, n = 8

1. **The number of addresses in this block is N = 2^{32-n} = 16,777,216.**

2. **To find the first address, we keep the leftmost 8 bits and set the rightmost 24 bits all to 0s.**
   The first address is 73.0.0.0/8, in which 8 is the value of *n*.

3. **To find the last address, we keep the leftmost 8 bits and set the rightmost 24 bits all to 1s.**
   The last address is 73.255.255.255.

Example 5.13 contd..

Figure 5.17 ***Solution to Example 5.13***

Solve by self

A classless address is given as 167.199.170.82/**27**. We can find the above three pieces of information.

The number of addresses in the network is 2^(32 − *n*) = 25 = 32 addresses. As n = 27

The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s.

Address: 167.199.170.82/27	10100111	11000111	10101010	01010010
First address: 167.199.170.64/27	10100111	11000111	10101010	01000000

The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s.

Address: 167.199.170.82/27	10100111	11000111	10101010	01011111
Last address: 167.199.170.95/27	10100111	11000111	10101010	01011111

Example 5.14

An address in a block is given as 180.8.17.9. Find the number of addresses in the block, the first address, and the last address.

Solution

As 180 belongs to class B, n = 16

1. The number of addresses in this block is

$N = 2^{32-n} = 65{,}536.$

2. To find the first address, we keep the leftmost 16 bits and set the rightmost 16 bits all to 0s.

The first address is 18.8.0.0/16, in which 16 is the value of *n*.

3. To find the last address, we keep the leftmost 16 bits and set the rightmost 16 bits all to 1s.

The last address is 18.8.255.255.

Example 5.14 contd..

Figure 5.17 ***Solution to Example 5.14***

Example 5.15

An address in a block is given as 200.11.8.45. Find the number of addresses in the block, the first address, and the last address.

Solution

200 belongs to class C.

1. **The number of addresses in this block is $N = 2^{32-n} = 256$.**
2. **To find the first address, we keep the leftmost 24 bits and set the rightmost 8 bits all to 0s. The first address is 200.11.8.0/16, in which 24 is the value of *n*.**
3. **To find the last address, we keep the leftmost 24 bits and set the rightmost 8 bits all to 1s. The last address is 200.11.8.255/16.**

Example 5.15

Figure 5.18 ***Solution to Example 5.15***

An Example

Figure 5.19 ***Sample Internet***

We can see….

1. A LAN with the network address 220.3.6.0 (class C).

2. A LAN with the network address 134.18.0.0 (class B).

3. A switched WAN (class C), such as Frame Relay or ATM, that can be connected to many routers. Where one router connects the WAN to the left LAN, one connects the WAN to the right LAN, and one connects the WAN to the rest of the internet.

Network Address, Network mask, Subnetting

DCN

Network Address, Network mask, Subnetting

Network Address

- **Network address,** is particularly important because it is used in routing a packet to its destination network.
- For the moment, let us assume that an internet is made of *m* networks and a router with *m* interfaces.
- When a packet arrives at the router from any source host, the router needs to know to which network the packet should be sent; the router needs to know from which interface the packet should be sent out.
- After the network address has been found, the router consults its routing table to find the corresponding interface from which the packet should be sent out.
- The network address is actually the identifier of the network; each network is identified by its network address.

Network Address (contd..)

Figure 5.20 *Network addresses*

Network Mask

- The methods we described previously for extracting the network address are mostly used to show the concept.
- The routers in the Internet normally use an algorithm to extract the network address from the destination address of a packet.
- To do this, we need a network mask.
- A **network mask** or a **default mask** in classful addressing is a 32-bit number with *n* leftmost bits all set to 1s and (32 − *n*) rightmost bits all set to 0s.
- Since *n* is different for each class in classful addressing, we have three default masks in classful addressing

Network Mask (contd..)

Figure 5.21 *Network mask*

Network Mask (contd..)

- To extract the network address from the destination address of a packet, a router uses the AND operation.
- When the destination address (or any address in the block) is ANDed with the default mask, the result is the network address
- The router applies the AND operation on the binary (or hexadecimal representation) of the address and the mask

Figure 5.22 ***Finding a network address using the default mask***

Example 5.16

A router receives a packet with the destination address 201.24.67.32. Show how the router finds the network address of the packet.

Solution

Since the class of the address is B, we assume that the router applies the default mask for class B, 255.255.0.0 to find the network address.

Destination address	→	201	.	24	.	67	.	32
Default mask	→	255	.	255	.	0	.	0
Network address	→	201	.	24	.	0	.	0

Three-Level Addressing: Subnetting

- the IP addresses were originally designed with two levels of
- addressing.—host and network address
- But there is need for more than two level of addressing the two reasons are:

1. An organization that was granted a block in class A or B needed to divide its large network into several subnetworks for better security and management.
2. Since the blocks in class A and B were almost depleted and the blocks in class C were smaller than the needs of most organizations, an organization that has been granted a block in class A or B could divide the block into smaller subblocks and share them with other organizations.

- **The idea of splitting a block to smaller blocks is referred to as subnetting.**

Example 5.17

Three-level addressing can be found in the telephone system if we think about the local part of a telephone number as an exchange and a subscriber connection:

(626) 358 - 1301

in which 626 is the area code, 358 is the exchange, and 1301 is the subscriber connection.

Example 5.18

Figure 5.23 shows a network using class B addresses before subnetting. We have just one network with almost 2^{16} hosts. The whole network is connected, through one single connection, to one of the routers in the Internet. Note that we have shown /16 to show the length of the netid (class B).

Figure 5.23 ***Example 5.18***

Example 5.19

Figure 5.24 shows the same network in Figure 5.23 after subnetting. The whole network is still connected to the Internet through the same router. However, the network has used a private router to divide the network into four subnetworks. The rest of the Internet still sees only one network; internally the network is made of four subnetworks. Each subnetwork can now have almost 2^{14} hosts. The network can belong to a university campus with four different schools (buildings). After subnetting, each school has its own subnetworks, but still the whole campus is one network for the rest of the Internet.

Note that /16 and /18 show the length of the netid and subnetids.

Note that /16 and /18 show the length of the netid and subnetids.

Subnet mask

- The network mask is used when a network is not subnetted.
- When we divide a network to several subnetworks, we need to create a subnetwork mask (or subnet mask) for each subnetwork.
- A subnetwork has subnetid and hostid as shown in Figure 5.25.

Figure 5.25 ***Network mask and subnetwork mask***

Subnet mask (contd..)

- Subnetting increases the length of the netid and decreases the length of hostid.
- When we divide a network to s number of subnetworks, each of equal numbers of hosts, we can calculate the subnetid for each subnetwork as:

$$n_{\text{sub}} = n + \log_2 s$$

- Where in which **n** is the length of netid, nsub is the length of each subnetid, and **s** is the number of subnets which must be a power of 2.

Example 5.20

In Example 5.19, we divided a class B network into four subnetworks.

$$n_{sub} = n + \log_2 s$$

The value of n = 16 and the value of

$$n_1 = n_2 = n_3 = n_4 = 16 + \log_2 4 = 18.$$

This means that the subnet mask has eighteen 1s and fourteen 0s.

In other words, the subnet mask is 255.255.192.0 which is different from the network mask for class B (255.255.0.0).

255.255.192.0

1111 1111 1111 1111 1100 0000 0000 0000

Subnet Address

- the first address in the subnet is the identifier of the subnet and is used by the router to route the packets destined for that subnetwork.
- Given any address in the subnet, the router can find the subnet mask using the same procedure we discussed to find the network mask: ANDing the given address with the subnet mask.
- The short cuts we discussed in the previous section can be used to find the subnet address.
    - 255 or Zero, lowest value is output
    - Else convert to binary and apply AND

AND

Input 1	Input 2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Operation for each bit

Example 5.21

In Example 5.19, we show that a network is divided into four subnets. Since one of the addresses in subnet 2 is 141.14.120.77, we can find the subnet address as:

Address	→	141	.	14	.	120	.	77
Mask	→	255	.	255	.	192	.	0
Subnet Address	→	141	.	14	.	64	.	0

The values of the first, second, and fourth bytes are calculated using the first short cut for AND operation. The value of the third byte is calculated using the second short cut for the AND operation.

Address (120)	0	+	64	+	32	+	16	+	8	+	0	+	0	+	0
Mask (192)	128	+	64	+	0	+	0	+	0	+	0	+	0	+	0
Result (64)	0	+	64	+	0	+	0	+	0	+	0	+	0	+	0

Supernetting

- **Subnetting could not completely solve address depletion problems** in classful addressing because **most organizations did not want to share their granted blocks** with others.
- Since class C blocks were still available but the size of the block did not meet the requirement of new organizations that wanted to join the Internet, one solution was supernetting.
- In **supernetting**, an organization can combine several class C blocks to create a larger range of addresses.
- In other words, several networks are combined to create a supernetwork.
- By doing this, an organization can apply for several class C blocks instead of just one.
- **Example**, an organization that needs 1000 addresses can be granted four class C blocks.

Supernetting (contd..)

- A **supernet mask** is the reverse of a subnet mask.
- A subnet mask for class C has more 1s than the default mask for this class.
- A supernet mask for class C has less 1s than the default mask for this class.
- From below diagram:
  - subnet mask that divides a block into eight subblocks has three more 1s (2^3 = 8) than the default mask;
  - A supernet mask that combines eight blocks into one superblock has three less 1s than the default mask.

Divide 1 class C block into 8 subblocks
Subnet mask
1 1 1

Supernetting (contd..)

- In supernetting, the number of class C addresses that can be combined to make a supernet needs to be a power of 2.
- The length of the supernetid can be found using the formula

$$n_{\text{super}} = n - \log_2 c$$

- Where in n_{super} defines the length of the supernetid in bits and c defines the number of class C blocks that are combined.

Drawbacks of supernetting:

1. The number of blocks to combine needs to be a power of 2, which means an organization that needed seven blocks should be granted at least eight blocks (address wasting).
2. Supernetting and subnetting really complicats the routing of packets in the Internet.

Classless addressing and block allocation

DCN

Classless addresses and Address Block allocation

Topics covered

- ✓ Variable –Length Blocks
- ✓ Two-Level Addressing
- ✓ Block Allocation
- ✓ Subnetting

Introduction

- Subnetting and supernetting in classful addressing did not really solve the address depletion problem.
- With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution.
- Although the long-range solution has already been devised and is called IPv6, a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization.
- The short-term solution still uses IPv4 addresses, but it is called *classless addressing*.

Introduction (contd..)

- During the 1990s, Internet Service Providers (ISPs)-provides Internet access for individuals, small businesses, and midsize organizations that.
- An ISP is granted a large range of addresses and then subdivides the addresses (in groups of 1, 2, 4, 8, 16, and so on), giving a range of addresses to a household or a small business.
- The customers are connected via a dial-up modem, DSL, or cable modem to the ISP.
- However, each customer needs some IPv4 addresses.
- In 1996, the Internet authorities announced a new architecture called classless addressing.
- In classless addressing, variable-length blocks are used that belong to no classes.
- We can have a block of 1 address, 2 addresses, 4 addresses, 128 addresses, and so on.

Variable-Length Blocks

- In classless addressing, the whole address space is divided into variable length blocks.
- Theoretically, we can have a block of 2^0, 2^1, 2^2, . . . , 2^32 addresses.
- The only restriction, as we discuss later, is that the number of addresses in a block needs to be a power of 2.
- An organization can be granted one or more block/s of addresses.

Figure 5.27 *Variable-length blocks in classless addressing*

Two-Level Addressing

- When an organization is granted a block of addresses, the block is actually divided into two parts, the **prefix** and the **suffix.**
- The prefix plays the same role as the netid; the suffix plays the same role as the hostid.
- All addresses in the block have the same prefix; each address has a different suffix.

Figure 5.28 ***Prefix and suffix***

In classless addressing, the prefix defines the network and the suffix defines the host.

Two-Level Addressing

- In classful addressing, the length of the ***netid***, n, depends on the class of the address; it can be only 8, 16, or 24.
- In classless addressing, the length of the ***prefix***, n, depends on the size of the block; it can be 0, 1, 2, 3, . . . , 32.
- In classless addressing, the **value of n is referred to as prefix** length; **the value of 32 − n is referred to as suffix length.**

Example 5.22

What is the prefix length and suffix length if the whole Internet is considered as one single block with 4,294,967,296 addresses?

Solution

In this case, the prefix length is 0 and the suffix length is 32.

All 32 bits vary to define 2^{32} = 4,294,967,296 hosts in this single block.

Example 5.23

What is the prefix length and suffix length if the Internet is divided into 4,294,967,296 blocks and each block has one single address?

Solution

In this case, the prefix length for each block is 32 and the suffix length is 0.

All 32 bits are needed to define 2^{32} = 4,294,967,296 blocks.

The only address in each block is defined by the block itself.

Example 5.24

The number of addresses in a block is inversely related to the value of the prefix length, *n*.

A small *n* means a larger block; a large *n* means a small block.

Slash Notation

The netid length in classful addressing or the prefix length in classless addressing play a very important role when we need to extract the information about the block from a given address in the block. However, there is a difference here in classful and classless addressing.

- In **classful addressing**, the netid length is inherent in the address. Given an address, we know the class of the address that allows us to find the netid length (8, 16, or 24).
- In **classless addressing**, the prefix length cannot be found if we are given only an address in the block. The given address can belong to a block with any prefix length.

Figure 5.29 ***Slash notation***

Therefore, slash notation is used to define the prefix length

The slash notation is formally referred to as **classless interdomain routing** or **CIDR** (pronounced cider) notation.

Note

In classless addressing, we need to know one of the addresses in the block and the prefix length to define the block.

Example 5.25

In classless addressing, an address cannot per se define the block the address belongs to.

For example, the address 230.8.24.56 can belong to many blocks some of them are shown below with the value of the prefix associated with that block:

Prefix length:16	→	Block:	230.8.0.0	to	230.8.255.255
Prefix length:20	→	Block:	230.8.16.0	to	230.8.31.255
Prefix length:26	→	Block:	230.8.24.0	to	230.8.24.63
Prefix length:27	→	Block:	230.8.24.32	to	230.8.24.63
Prefix length:29	→	Block:	230.8.24.56	to	230.8.24.63
Prefix length:31	→	Block:	230.8.24.56	to	230.8.24.57

Network Mask

- The idea of network mask in classless addressing is the same as the one in classful addressing.
- A network mask is a 32-bit number with the n leftmost bits all set to 0s and the rest of the bits all set to 1s.

Example 5.26

The following addresses are defined using slash notations.

a. **In the address 12.23.24.78/8, the network mask is 255.0.0.0. The mask has eight 1s and twenty-four 0s. The prefix length is 8; the suffix length is 24.**

b. **In the address 130.11.232.156/16, the network mask is 255.255.0.0. The mask has sixteen 1s and sixteen 0s.The prefix length is 16; the suffix length is 16.**

c. **In the address 167.199.170.82/27, the network mask is 255.255.255.224. The mask has twenty-seven 1s and five 0s. The prefix length is 27; the suffix length is 5.**

$$\boxed{1011}$$

$$\begin{array}{r} 0011 \\ \hline 0100_2 \end{array}$$

	R	Q
2^1	1	5
2^2	3	2
2^3	3	1
2^4	11	0

2

Extracting Block Information

- An address in slash notation (CIDR) contains all information we need about the block:
  - The number of addresses,
  - The first address (network address) and
  - The last address
- These three pieces of information can be found as follows:

n = 16

0 . 0 . 1111 . 1111

- The number of addresses in the block can be found as:

$$N = 2^{32-n}$$

- The first address (network address) in the block can be found by ANDing the address with the network mask:

First address = (any address) AND (network mask)

- The last address in the block can be found by either adding the first address with the number of addresses or, directly, by ORing the address with the complement (NOTing) of the network mask:

Last address = (any address) OR [NOT (network mask)]

Alternatively, we can keep the n leftmost bits of any address in the block and set the $32 - n$ bits to 1s to find the last address.

Example 5.27

One of the addresses in a block is 167.199.170.82/27. Find the number of addresses in the network, the first address, and the last address.

Solution

The value of n is 27. The network mask has twenty-seven 1s and five 0s. It is 255.255.255.240.

a. The number of addresses in the network is $2^{32-n} = 32$.

b. We use the AND operation to find the first address (network address). The first address is 167.199.170.64/27.

Address in binary:	10100111	11000111	10101010	01010010
Network mask:	11111111	11111111	11111111	11100000
First address:	10100111	11000111	10101010	01000000

c. To find the last address, we first find the complement of the network mask and then OR it with the given address: The last address is 167.199.170.95/27.

Address in binary:	10100111	11000111	10101010	01010010
Complement of network mask:	00000000	00000000	00000000	00011111
Last address:	10100111	11000111	10101010	01011111

Example 5.28

One of the addresses in a block is 17.63.110.114/24. Find the number of addresses, the first address, and the last address in the block.

Solution

The network mask is 255.255.255.0.

a. The number of addresses in the network is $2^{32-24} = 256$.

b. To find the first address, we use the short cut methods discussed early in the chapter. The first address is 17.63.110.0/24.

Address:	17	.	63	.	110	.	114
Network mask:	255	.	255	.	255	.	0
First address (AND):	17	.	63	.	110	.	0

c. To find the last address, we use the complement of the network mask and the first short cut method we discussed before. The last address is 17.63.110.255/24.

Address in binary:	10100111	11000111	10101010	01010010
Complement of network mask:	00000000	00000000	00000000	00011111
Last address:	10100111	11000111	10101010	01011111

Example 5.29

One of the addresses in a block is 110.23.120.14/20. Find the number of addresses, the first address, and the last address in the block.

Solution

The network mask is 255.255.240.0.

a. The number of addresses in the network is $2^{32-20} = 4096$.

b. To find the first address, we apply the first short cut to bytes 1, 2, and 4 and the second short cut to byte 3. The first address is 110.23.112.0/20.

Address:	110	.	23	.	120	.	14
Network mask:	255	.	255	.	240	.	0
First address (AND):	110	.	23	.	112	.	0

c. To find the last address, we apply the first short cut to bytes 1, 2, and 4 and the second short cut to byte 3. The OR operation is applied to the complement of the mask. The last address is 110.23.127.255/20.

Address:	110	.	23	.	120	.	14
Network mask:	0	.	0	.	15	.	255
Last address (OR):	110	.	23	.	127	.	255

Block Allocation

- How are the blocks allocated?
- Internet Corporation for Assigned Names and Addresses (ICANN) allocates block to ISP.
- ISP allocates the block to Internet user based on CIDR.
- Some restrictions of CIDR are:

1. All IP addresses should continuous

size

2. Block should be power of 2

3. First IP address in the block should be evenly divisible by size of the block.

2^n

100.1.2.32

100.1.2.33

⋮

100.1.2.47

R1 → ✓

R2 − 16 = 24

R3

n

⇓

100.1.2.00100000

✓

ISP → CIDR

20 . 10 . 30 . 32

20 . 10 . 30 . 33

⋮

20 . 10 . 30 . 63

3

R1

R2 – 32

2^5

R3 20.10.30.32 ÷ 2^5

20.10.30.001 00000

R

150.10.20.64

150.10.20.65

⋮

150.10.20.127

CIDR ?

Block

100.1.2.32/28 / n.

2^4

100.1.2.00100000

32-4

= 28

20.10.30.32/27

150.10.20.64/26

2

Extract info about a block

1

2 How allocate a block

Example 5.30

An ISP has requested a block of 1000 addresses. The following block is granted.

a. **Since 1000 is not a power of 2, 1024 addresses are granted (1024 = 2^10).**
b. **The prefix length for the block is calculated as n = 32 − $\log_2 1024$ = 22.**
c. **The beginning address is chosen as 18.14.12.0 (which is divisible by 1024).**

The granted block is 18.14.12.0/22. The first address is 18.14.12.0/22 and the last address is 18.14.15.255/22.

Relation to Classful Addressing

- Classful addressing is a special case of the classless addressing in which the blocks in class A, B, and C have the prefix length $n_A = 8$, $n_B = 16$, and $n_C = 24$.
- A block in classful addressing can be easily changed to a block in class addressing if we use the prefix length defined in Table 5.1

Table 5.1 *Prefix length for classful addressing*

Class	*Prefix length*	*Class*	*Prefix length*
A	/8	D	/4
B	/16	E	/4
C	/24		

Example 5.31

Assume an organization has given a class A block as 73.0.0.0 in the past. If the block is not revoked by the authority, the classless architecture assumes that the organization has a block 73.0.0.0/8 in classless addressing.

Subnetting

32 – [8 8 8 8]

- Three levels of hierarchy can be created using subnetting. An organization (or an ISP) that is granted a range of addresses may divide the range into several subranges and assign each subrange to a subnetwork (or subnet).
- A subnetwork can be divided into several sub-subnetworks.
- A sub-subnetwork can be divided into several sub-sub-subnetworks. And so on.
- Three concepts need to familiarized:
    - Designing subnets
    - Finding information about each subnetwork
    - Address aggregation

16

1. Designing Subnets

- The subnetworks in a network should be carefully designed to enable the routing of packets.
- We assume the total number of addresses granted to the organization is N, the prefix length is n, the assigned number of addresses to each subnetwork is N_{sub}, the prefix length for each subnetwork is n_{sub}, and the total number of subnetworks is s.
- Then, the following steps need to be carefully followed to guarantee the proper operation of the subnetworks.

1. The number of addresses in each subnetwork should be a power of 2.
2. The prefix length for each subnetwork should be found using the following formula:

$$n_{sub} = n + \log_2 (N/N_{sub})$$

3. The starting address in each subnetwork should be divisible by the number of addresses in that subnetwork. This can be achieved if we first assign addresses to larger networks.

2. Finding information about each subnetwork

- After designing the subnetworks, the information about each subnetwork, such as first and last address, can be found using the process we described to find the information about each network in the Internet.

Example 5.32

An organization is granted the block 130.34.12.64/26. The organization needs four subnetworks, each with an equal number of hosts. Design the subnetworks and find the information about each network.

32 − 26

N = 2 =

Solution

The number of addresses for the whole network can be found as N = 2^(32 − 26) = 64. The first address in the network is 130.34.12.64/26 and the last address is 130.34.12.127/26.

We now design the subnetworks:

1. We grant 16 addresses for each subnetwork to meet the first requirement (64/16 is a power of 2).
2. The subnetwork mask for each subnetwork is:

prefix

$$n_1 = n_2 = n_3 = n_4 = n + \log_2 (N/N_i) = 26 + \log_2 4 = 28$$

28

Example 5.32 *Continued*

3. We grant 16 addresses to each subnet starting from the first available address. Figure 5.30 shows the subblock for each subnet. Note that the starting address in each subnetwork is divisible by the number of addresses in that subnetwork.

Figure 5.30 *Solution to Example 5.32*

Example 5.33

24

An organization is granted a block of addresses with the beginning address 14.24.74.0/24. The organization needs to have 3 subblocks of addresses to use in its three subnets as shown below:

- **One subblock of 120 addresses.**
- **One subblock of 60 addresses.**
- **One subblock of 10 addresses.**

Solution

There are 2^{32-24} = 256 addresses in this block. The first address is 14.24.74.0/24; the last address is 14.24.74.255/24.

a. The number of addresses in the first subblock is not a power of 2. We allocate 128 addresses. The subnet mask is 25. The first address is 14.24.74.0/25; the last address is 14.24.74.127/25.

256 ⇒ 128 → Prob

Example 5.33 Continued

b. The number of addresses in the second subblock is not a power of 2 either. We allocate 64 addresses. The subnet mask is 26. The first address in this block is 14.24.74.128/26; the last address is 14.24.74.191/26.

64 mask

c. The number of addresses in the third subblock is not a power of 2 either. We allocate 16 addresses. The subnet mask is 28. The first address in this block is 14.24.74.192/28; the last address is 14.24.74.207/28.

16

d. If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is 14.24.74.209. The last address is 14.24.74.255.

e. Figure 5.31 shows the configuration of blocks. We have shown the first address in each block.

Example 5.33 Continued

Figure 5.31 ***Solution to Example 5.33***

Example 5.34

Assume a company has three offices: Central, East, and West. The Central office is connected to the East and West offices via private, WAN lines. The company is granted a block of 64 addresses with the beginning address 70.12.100.128/26. The management has decided to allocate 32 addresses for the Central office and divides the rest of addresses between the two other offices.

64 — 32, 16, 16

1. **The number of addresses are assigned as follows:**

Central office $N_c = 32$	East office $N_e = 16$	West office $N_w = 16$

2. **We can find the prefix length for each subnetwork:**

$n_c = n + \log_2(64/32) = 27$	$n_e = n + \log_2(64/16) = 28$	$n_w = n + \log_2(64/16) = 28$

N = 64 , N sub = 32/16/16

Example 5.34 Continued

3. Figure 5.32 shows the configuration designed by the management.

- The Central office uses addresses 70.12.100.128/27 to 70.12.100.159/27.
- The company has used three of these addresses for the routers and has reserved the last address in the subblock.
- The East office uses the addresses 70.12.100.160/28 to 70.12.100.175/28.
- One of these addresses is used for the router and the company has reserved the last address in the subblock.
- The West office uses the addresses 70.12.100.160/28 to 70.12.100.175/28.
- One of these addresses is used for the router and the company has reserved the last address in the subblock.
- The company uses no address for the point-to-point connections in WANs.

Example 5.34 Continued

Figure 5.32 ***Example 5.34***

3. Address Aggregation

- One of the advantages of CIDR architecture is address aggregation.
- Internet Corporation for Assigned Names (ICANN) assigns a large block of addresses to an ISP.
- Each ISP in turn divides its assigned block into smaller subblocks and grants the subblocks to its customers; many blocks of addresses are aggregated in one block and granted to one ISP.

Example 5.35

N

An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

- **The first group has 64 customers; each needs approximately 256 addresses.** 64 × 256
- **The second group has 128 customers; each needs approximately 128 addresses.** 128 × 128
- **The third group has 128 customers; each needs approximately 64 addresses.** 128 × 64

We design the subblocks and find out how many addresses are still available after these allocations.

Example 5.35 Continued

Solution

Let us solve the problem in two steps. In the first step, we allocate a subblock of addresses to each group. The total number of addresses allocated to each group and the prefix length for each subblock can found as

Group 1: $64 \times 256 = 16{,}384$	$n_1 = 16 + \log_2 (65536/16384) = 18$
Group 2: $128 \times 128 = 16{,}384$	$n_2 = 16 + \log_2 (65536/16384) = 18$
Group 3: $128 \times 64 = 8192$	$n_3 = 16 + \log_2 (65536/8192) = 19$

Figure 5.33 shows the design for the first hierarchical level. Figure 5.34 shows the second level of the hierarchy. Note that we have used the first address for each customer as the subnet address and have reserved the last address as a special address.

Example 5.35 Continued

Figure 5.33 *Solution to Example 5.35: first step*

Example 5.35 Continued

Group: $n = 18$

Subnet: $n = 18 + \log_2 (16385/256) = 24$

Group: $n = 18$

Subnet: $n = 18 + \log_2 (16385/128) = 25$

Group: $n = 19$

Subnet: $n = 19 + \log_2 (8192/64) = 26$

Figure 5.34 ***Solution to Example 5.35: second step***

Special address and Network Address Translation

DCN

Special address and Network Address Translation

SPECIAL ADDRESSES

- **In classful addressing some addresses were reserved for special purposes.**
- **The classless addressing scheme inherits some of these special addresses from classful addressing.**
- **Topics discussed:**
  - ✓ **Special Blocks**
    - ✓ **All zero address**
    - ✓ **All-ones address: limited BC address**
    - ✓ **Loopback addresses**
    - ✓ **Private address**
  - ✓ **Special Addresses in each Block**
    - ✓ **Network address**
    - ✓ **Direct broadcast address**

All-Zeros Address

The block 0.0.0.0/32, which contains only one single address, is reserved for communication when a host needs to send an IPv4 packet but it does not know its own address.

Figure 5.35 *Example of using the all-zero address*

All-Ones Addresses: Limited to broadcast address

The block 255.255.255.255/32, which contains one single address, is reserved for limited broadcast address in the current network. A host that wants to send a message to every other host can use this address as a destination address in an IPv4 packet.

Figure 5.36 *Example of limited broadcast address*

Loopback Address

- The block 127.0.0.0/8 is used for the **loopback address,** which is an address used to test the software on a machine.
- When this address is used, a packet never leaves the machine; it simply returns to the protocol software.
- It can be used to test the IPv4 software.

Figure 5.37 ***Example of loopback address***

Private Address

A number of blocks are assigned for private use. They are not recognized globally. These blocks are depicted in Table 5.2. These addresses are used either in isolation or in connection with network address translation techniques

Table 5.2 *Addresses for private networks*

Block	*Number of addresses*	*Block*	*Number of addresses*
10.0.0.0/8	16,777,216	192.168.0.0/16	65,536
172.16.0.0/12	1,047,584	169.254.0.0/16	65,536

Multicast Address

The block 224.0.0.0/4 is reserved for multicast communication. We discuss multicasting.

Direct broadcast address

- The last address in a block or subblock (with the suffix set all to 1s) can be used as a **direct broadcast address.**
- This address is usually used by a router to send a packet to all hosts in a specific network.

Figure 5.38 *Example of a directed broadcast address*

Network Address Translation (NAT)

- **The distribution of addresses through ISPs has created a new problem.**
- **If the business grows or the household needs a larger range, the ISP may not be able to grant the demand because the addresses before and after the range may have already been allocated to other networks.**
- **In most situations, however, only a portion of computers in a small network need access to the Internet simultaneously.**
- **A technology that can help in this cases is *network address translation (NAT).***

✓ **Address Translation**

✓ **Translation Table**

NAT

- The technology allows a site to use a set of private addresses for internal communication and a set of global Internet addresses (at least one) for communication with the rest of the world.
- The site must have only one single connection to the global Internet through a NAT-capable router that runs NAT software.

Figure 5.39 ***NAT***

As the figure shows, the private network uses private addresses. The router that connects the network to the global address uses one private address and one global address. The private network is transparent to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8.

Address Translation

- All of the outgoing packets go through the NAT router, which replaces the *source address* in the packet with the global NAT address.
- All incoming packets also pass through the NAT router, which replaces the *destination address* in the packet (the NAT router global address) with the appropriate private address.

Figure 5.40 ***Address translation***

Address Translation

- The reader may have noticed that translating the source addresses for an outgoing packet is straightforward.
- But how does the NAT router know the destination address for a packet coming from the Internet?
- There may be tens or hundreds of private IP addresses, each belonging to one specific host. The problem is solved if the NAT router has a translation table.

✓ Using one IP address
✓ Using a pool of IP addresses

Using One IP Address

Figure 5.41 ***Translation***

NAT Table with only IP addresses (1)

NAT Table with Pool of IP addresses (3)

- If using only one global address
  - Only one private-network host to access the same external host
- If using a pool of global addresses (e.g. 4 addr)
  - No more than 4 connections can be made to the same destination
  - No private-network host can access two external server programs (e.g. HTTP and TELNET) at the same external host at the same time ???
  - Two private-network hosts cannot access the same external server program at the same time (by using the same global address)

NAT Table with IP address & Port # (1)

Same external host (Telnet)

1 global address

3 privates

	NAT Table for **G1**		
P1:#1→D1:#23	P1:#1	**G1:#1**	G1:#1→D1:#23
P1:#2→D1:#23	P1:#2	**G1:#2**	G1:#2→D1:#23
P2:#1→D1:#23	P2:#1	**G1:#3**	G1:#3→D1:#23
P3:#1→D1:#23	P3:#1	**G1:#4**	G1:#4→D1:#23

Must be unique

D1:#23→P1:#1	D1:#23→G1:#1
D1:#23→P1:#2	D1:#23→G1:#2
D1:#23→P2:#1	D1:#23→G1:#3
D1:#23→P3:#1	D1:#23→G1:#4

NAT Table with IP address & Port # (2)

NAT Table with IP address & Port # (3)

NAT Table Pool of IP addresses

- NAT Table in the textbook
  - This may not be the best way to understand NAT

Server

Private Address	*Private Port*	*External Address*	*External Port*	*Transport Protocol*
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
…	…	…	…	…

Must be unique

Otherwise reassign a new private port and cache the mapping (the new port --> the old port)

Different delivery and forwarding methods

Delivery and Forwarding of IP Packets

OBJECTIVES:

- To discuss the delivery of packets in the network layer and distinguish between direct and indirect delivery.
- To discuss the forwarding of packets in the network layer and distinguish between destination-address–based forwarding and label-based forwarding.
- To discuss different forwarding techniques, including next-hop, network-specific, host-specific, and default.
- To discuss the contents of routing tables in classful and classless addressing and some algorithms used to search the tables.
- To introduce MPLS technology and show how it can achieve label based forwarding.
- To list the components of a router and explain the purpose of each component and their relations to other components.

Chapter Outline

6.1 Delivery

6.2 Forwarding

6.3 Structure of a Router

6-1 DELIVERY

The network layer supervises the handling of the packets by the underlying physical networks. We define this handling as the delivery of a packet. The delivery of a packet to its final destination is accomplished using two different methods of delivery: direct and indirect.

Topics Discussed in the Section

- ✓ **Direct Delivery**
- ✓ **Indirect Delivery**

Figure 6.1 *Direct delivery*

Figure 6.2 *Indirect delivery*

6-2 FORWARDING

Forwarding means to place the packet in its route to its destination. Since the Internet today is made of a combination of links (networks), forwarding means to deliver the packet to the next hop (which can be the final destination or the intermediate connecting device). Although the IP protocol was originally designed as a connectionless protocol, today the tendency is to use IP as a connection-oriented protocol.

Topics Discussed in the Section

- ✓ **Forwarding Based on Destination Address**
- ✓ **Forwarding Based on Label**

Figure 6.3 *Next-hop method*

A

Destination	Route
Host B	R1, R2, Host B

R1

Destination	Route
Host B	R2, Host B

R2

Destination	Route
Host B	Host B

a. Routing tables based on route

A

Destination	Next Hop
Host B	R1

R1

Destination	Next Hop
Host B	R2

R2

Destination	Next Hop
Host B	---

b. Routing tables based on next hop

Figure 6.4 *Network-specific method*

Network-specific
routing table for host S

Destination	Next Hop
N2	R1

Host-specific
routing table for host S

Destination	Next Hop
A	R1
B	R1
C	R1
D	R1

Figure 6.5 *Host-specific routing*

Routing table for host A

Destination	Next Hop
Host B	R3
N2	R1
N3	R3
.....	.....

Figure 6.6 *Default routing*

Routing table for host A

Destination	Next Hop
N2	R1
.....	.....
Default	R2

Figure 6.7 *Simplified forwarding module in classful address without subnetting*

a. 0 to 7, the class is A.
b. 8 to 11, the class is B.
c. 12 or 13, the class is C
d. 14, the class is D.
e. 15, the class is E.

Right shift with 28 bits to find the class

Example 6.1

Figure 6.8 shows an imaginary part of the Internet. Show the routing tables for router R1.

Solution

Figure 6.9 shows the three tables used by router R1. Note that some entries in the next-hop address column are empty because in these cases, the destination is in the same network to which the router is connected (direct delivery). In these cases, the next-hop address used by ARP is simply the destination address of the packet as we will see in Chapter 8.

Figure 6.8 *Configuration for routing, Example 6.1*

Figure 6.9 *Tables for Example 6.1*

Class A

Network address	Next-hop address	Interface
111.0.0.0	----------	m0

Class B

Network address	Next-hop address	Interface
145.80.0.0	----------	m1
170.14.0.0	----------	m2

Class C

Network address	Next-hop address	Interface
192.16.7.0	111.15.17.32	m0

Default: 111.30.31.18, m0

Example 6.2

Router R1 in Figure 6.8 receives a packet with destination address 192.16.7.14. Show how the packet is forwarded.

Solution

The destination address is

11000000 00010000 00000111 0001110.

A copy of the address is shifted 28 bits to the right. The result is

00000000 00000000 00000000 00001100 or 12.

The destination network is class C. The network address is extracted by masking off the leftmost 24 bits of the destination address; the result is 192.16.7.0. The table for Class C is searched. The network address is found in the first row. The next-hop address 111.15.17.32. and the interface m0 are passed to ARP (see Chapter 8).

11000000 00010000 00000111 0001110

167.24.160.5

↓

Next hop

Network id

Example 6.3

Router R1 in Figure 6.8 receives a packet with destination address 167.24.160.5. Show how the packet is forwarded.

Solution

The destination address in binary is

10100111 00011000 10100000 00000101.

A copy of the address is shifted 28 bits to the right. The result is

00000000 00000000 00000000 00001010 or 10.

The class is B. The network address can be found by masking off 16 bits of the destination address, the result is 167.24.0.0. The table for Class B is searched. No matching network address is found. The packet needs to be forwarded to the default router (the network is somewhere else in the Internet). The next-hop address 111.30.31.18 and the interface number m0 are passed to ARP.

Figure 6.10 *Simplified forwarding module in classful address with subnetting*

Example 6.4

Figure 6.11 shows a router connected to four subnets. Note several points.

First, the site address is 145.14.0.0/16 (a class B address). Every packet with a destination address in the range 145.14.0.0 to 145.14.255.255 is delivered to the interface m4 and distributed to the final destination subnet by the router.

Second, we have used the address x.y.z.t/n for the interface m4 because we do not know to which network this router is connected.

Third, the table has a default entry for packets that are to be sent out of the site. The router is configured to apply the subnet mask /18 to any destination address.

Figure 6.11 *Configuration for Example 6.4*

Subnet address	Next-hop address	Interface number
145.14.0.0	----------	m0
145.14.64.0	----------	m1
145.14.128.0	----------	m2
145.14.192.0	----------	m3
0.0.0.0	default router	m4

Example 6.5

The router in Figure 6.11 receives a packet with destination address 145.14.32.78. Show how the packet is forwarded.

Solution

The mask is /18. After applying the mask, the subnet address is 145.14.0.0. The packet is delivered to ARP (see Chapter 8) with the next-hop address 145.14.32.78 and the outgoing interface m0.

Example 6.6

A host in network 145.14.0.0 in Figure 6.11 has a packet to send to the host with address 7.22.67.91. Show how the packet is routed.

Solution

The router receives the packet and applies the mask (/18). The network address is 7.22.64.0. The table is searched and the address is not found. The router uses the address of the default router (not shown in figure) and sends the packet to that router.

Figure 6.12 *Simplified forwarding module in classless address*

Figure 6.13 *Configuration for Example 6.7*

Example 6.7

Make a routing table for router R1 using the configuration in Figure 6.13.

Solution

Table 6.1 shows the corresponding table

Table 6.1 *Routing table for router R1 in Figure 6.13*

Mask	*Network Address*	*Next Hop*	*Interface*
/26	180.70.65.192	-	m2
/25	180.70.65.128	-	m0
/24	201.4.22.0	-	m3
/22	201.4.16.0	....	m1
Default	Default	180.70.65.200	m2

Example 6.8

Show the forwarding process if a packet arrives at R1 in Figure 6.13 with the destination address 180.70.65.140.

Solution

The router performs the following steps:

1. **The first mask (/26) is applied to the destination address. The result is 180.70.65.128, which does not match the corresponding network address.**
2. **The second mask (/25) is applied to the destination address. The result is 180.70.65.128, which matches the corresponding network address. The next-hop address (the destination address of the packet in this case) and the interface number m0 are passed to ARP (see Chapter 8) for further processing.**

Example 6.9

Show the forwarding process if a packet arrives at R1 in Figure 6.13 with the destination address 201.4.22.35.

Solution

The router performs the following steps:

1. **The first mask (/26) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 1).**
2. **The second mask (/25) is applied to the destination address. The result is 201.4.22.0, which does not match the corresponding network address (row 2).**
3. **The third mask (/24) is applied to the destination address. The result is 201.4.22.0, which matches the corresponding network address.**

Example 6.10

Show the forwarding process if a packet arrives at R1 in Figure 6.13 with the destination address 18.24.32.78.

Solution

This time all masks are applied to the destination address, but no matching network address is found. When it reaches the end of the table, the module gives the next-hop address 180.70.65.200 and interface number m2 to ARP (see Chapter 8). This is probably an outgoing package that needs to be sent, via the default router, to someplace else in the Internet.

Example 6.11

Now let us give a different type of example. Can we find the configuration of a router if we know only its routing table? The routing table for router R1 is given in Table 6.2. Can we draw its topology?

Solution

We know some facts but we don't have all for a definite topology. We know that router R1 has three interfaces: m0, m1, and m2. We know that there are three networks directly connected to router R1. We know that there are two networks indirectly connected to R1. There must be at least three other routers involved (see next-hop column). We do not know if network 140.6.12.64 is connected to router R3 directly or through a point-to-point network (WAN) and another router. Figure 6.14 shows our guessed topology.

Table 6.2 *Routing table for Example 6.11*

Mask	*Network Address*	*Next-Hop Address*	*Interface Number*
/26	140.6.12.64	180.14.2.5	m2
/24	130.4.8.0	190.17.6.2	m1
/16	110.70.0.0	----------	m0
/16	180.14.0.0	----------	m2
/16	190.17.0.0	----------	m1
Default	Default	110.70.4.6	m0

Figure 6.14 *Guessed topology for Example 6.11*

Figure 6.15 *Address aggregation*

Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/26	140.24.7.192	----------	m3
/0	0.0.0.0	default router	m4

Routing table for R1

Mask	Network address	Next-hop address	Interface
/24	140.24.7.0	----------	m0
/0	0.0.0.0	default router	m1

Routing table for R2

Figure 6.16 *Longest mask matching*

Routing table for R2

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	----------	m1
/24	140.24.7.0	----------	m0
/??	???????	?????????	m1
/0	0.0.0.0	default router	m2

Routing table for R1

Mask	Network address	Next-hop address	Interface
/26	140.24.7.0	----------	m0
/26	140.24.7.64	----------	m1
/26	140.24.7.128	----------	m2
/0	0.0.0.0	default router	m3

Routing table for R3

Mask	Network address	Next-hop address	Interface
/26	140.24.7.192	----------	m0
/??	???????	?????????	m1
/0	0.0.0.0	default router	m2

Hierarchical Routing

To solve the problem of gigantic routing tables, we can create a sense of hierarchy in the routing tables.

Example 6.12

As an example of hierarchical routing, let us consider Figure 6.17. A regional ISP is granted 16,384 addresses starting from 120.14.64.0. The regional ISP has decided to divide this block into 4 subblocks, each with 4096 addresses. Three of these subblocks are assigned to three local ISPs, the second subblock is reserved for future use. Note that the mask for each block is /20 because the original block with mask /18 is divided into 4 blocks.

Figure 6.17 *Hierarchical routing with ISPs*

Forwarding Based On Label

Figure 6.18 *Example 6.13: Forwarding based on destination address*

Example 6.14

Figure 6.19 shows a simple example of using a label to access a switching table. Since the labels are used as the index to the table, finding the information in the table is immediate.

Multi-Protocol Label Switching(MPLS)

- Conventional routers in the Internet can be replaced by MPLS routers that can behave like a router and a switch.
  - When behaving like a router, MPLS can forward the packet based on the destination address;
  - When behaving like a switch, it can forward a packet based on the label.

Figure 6.20 ***MPLS header is added to an IP packet***

Figure 6.21 *MPLS header made of stack of labels*

- **Label.** This 20-bit field defines the label that is used to index the routing table in the router.
- **Exp.** This 3-bit field is reserved for experimental purposes.
- **S.** The one-bit stack field defines the situation of the subheader in the stack. When the bit is 1, it means that the header is the last one in the stack.
- **TTL.** This 8-bit field is similar to the TTL field in the IP datagram, time-to-live. Each visited router decrement the value of this field. When it reaches zero, the packet is discarded to prevent looping.

6-3 STRUCTURE OF A ROUTER

In our discussion of forwarding and routing, we represented a router as a black box that accepts incoming packets from one of the input ports (interfaces), uses a routing table to find the output port from which the packet departs, and sends the packet from this output port.

Figure 6.22 *Router components*

Chapter 7

Internet Protocol Version4 (IPv4)

7-1 INTRODUCTION

The Internet Protocol (IP) is the transmission mechanism used by the TCP/IP protocols at the network layer.

Figure 7.1 *Position of IP in TCP/IP protocol suite*

7-2 DATAGRAMS

Packets in the network (internet) layer are called *datagrams*.

A datagram is a variable-length packet consisting of two parts: header and data.

The header is 20 to 60 bytes in length and contains information essential to routing and delivery.

Figure 7.2 *IP datagram*

Figure 7.3 *Service type*

Table 7.1 *Values for codepoints*

Category	*Codepoint*	*Assigning Authority*
1	XXXXX0	Internet
2	XXXX11	Local
3	XXXX01	Temporary or experimental

Note

The total length field defines the total length of the datagram including the header.

Figure 7.4 ***Encapsulation of a small datagram in an Ethernet frame***

Figure 7.5 ***Multiplexing***

Table 7.2 *Protocols*

Value	*Protocol*	*Value*	*Protocol*
1	ICMP	17	UDP
2	IGMP	89	OSPF
6	TCP		

Example 7.1

An IP packet has arrived with the first 8 bits as shown:

01000010

The receiver discards the packet. Why?

Solution

There is an error in this packet. The 4 left-most bits (0100) show the version, which is correct. The next 4 bits (0010) show the wrong header length ($2 \times 4 = 8$). The minimum number of bytes in the header must be 20. The packet has been corrupted in transmission.

Example 7.2

In an IP packet, the value of HLEN is 1000 in binary. How many bytes of options are being carried by this packet?

Solution

The HLEN value is 8, which means the total number of bytes in the header is 8 × 4 or 32 bytes. The first 20 bytes are the base header, the next 12 bytes are the options.

Example 7.4

An IP packet has arrived with the first few hexadecimal digits as shown below:

45000028000100000102 . . .

How many hops can this packet travel before being dropped? The data belong to what upper layer protocol?

Solution

To find the time-to-live field, we skip 8 bytes (16 hexadecimal digits). The time-to-live field is the ninth byte, which is 01. This means the packet can travel only one hop. The protocol field is the next byte (02), which means that the upper layer protocol is IGMP (see Table 7.2)

7-3 FRAGMENTATION

- **A datagram can travel through different networks.**
- **Each router decapsulates the IP datagram from the frame it receives, processes it, and then encapsulates it in another frame.**
- **The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled.**
- **The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.**

Maximum Transfer Unit (MTU)

- **Each data link layer protocol has its own frame format in most protocols.**
- **One of the fields defined in the format is the maximum size of the data field.**
- **In other words, when a datagram is encapsulated in a frame, the total size of the datagram must be less than this maximum size → MTU = 65,535 bytes.**
- **The value for:**
  - **Ethernet LAN is 1500 bytes,**
  - **FDDI LAN is 4352 bytes,**
  - **PPP is 296 bytes.**

Figure 7.6 *MTU*

Only data in a datagram is fragmented.

Fragmentation (contd..)

- The source usually does not fragment the IP packet.
- The transport layer will instead segment the data into a size that can be accommodated by IP and the data link layer in use.
- When a datagram is fragmented, each fragment has its own header with most of the fields repeated, but some change.
- A fragmented datagram may itself be fragmented if it encounters a network with an even smaller MTU.
- In other words, a datagram can be fragmented several times before it reaches the final destination.

Fields Related to Fragmentation

Figure 7.7 ***Flags field***

D	0	The datagram can be fragmented if necessary
	1	The machine must not fragment the datagram.
M	0	It means this is the last or only fragment
	1	It means the datagram is not the last fragment; there are more fragments after this one.

Figure 7.8 *Fragmentation example*

NOTE:
Remember that the value of the offset is measured in units of 8 bytes. This is done because the length of the offset field is only 13 bits long and cannot represent a sequence of bytes greater than 8191.

Figure 7.9 *Detailed fragmentation example*

Example 7.5

A packet has arrived with an M bit value of 0. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

- **If the M bit is 0, it means that there are no more fragments; the fragment is the last one.**
- **However, we cannot say if the original packet was fragmented or not.**
- **A nonfragmented packet is considered the last fragment.**

D	0	The datagram can be fragmented if necessary
	1	The machine must not fragment the datagram.
M	0	It means this is the last or only fragment
	1	It means the datagram is not the last fragment; there are more fragments after this one.

Example 7.6

A packet has arrived with an M bit value of 1. Is this the first fragment, the last fragment, or a middle fragment? Do we know if the packet was fragmented?

Solution

- **If the M bit is 1, it means that there is at least one more fragment.**
- **This fragment can be the first one or a middle one, but not the last one.**
- **We don't know if it is the first one or a middle one; we need more information (the value of the fragmentation offset). See also the next example.**

Example 7.7

A packet has arrived with an M bit value of 1 and a fragmentation offset value of zero. Is this the first fragment, the last fragment, or a middle fragment?

Solution

Because the M bit is 1, it is either the first fragment or a middle one.

Because the offset value is 0, it is the first fragment.

Example 7.9

A packet has arrived in which the offset value is 100, the value of HLEN is 5 and the value of the total length field is 100. What is the number of the first byte and the last byte?

Solution

The first byte number is $100 \times 8 = 800$.

The total length is 100 bytes and the header length is 20 bytes (5×4), which means that there are 80 bytes in this datagram.

If the first byte number is 800, the last byte number must be 879.

7-4 OPTIONS

The header of the IP datagram is made of two parts: a fixed part and a variable part. The fixed part is 20 bytes long and was discussed in the previous section. The variable part comprises the options, which can be a maximum of 40 bytes.

Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging. Although options are not a required part of the IP header, option processing is required of the IP software.

Figure 7.10 *Option format*

Figure 7.11 *Categories of options*

7-5 CHECKSUM

- The error detection method used by most TCP/IP protocols is called the checksum.
- The checksum protects against the corruption that may occur during the transmission of a packet.
- It is redundant information added to the packet.
- The checksum is calculated at the sender and the value obtained is sent with the packet.
- The receiver repeats the same calculation on the whole packet including the checksum.
- If the result is satisfactory, the packet is accepted; otherwise, it is rejected.

Topics Discussed in the Section

- ✓ **Checksum Calculation at the Sender**
- ✓ **Checksum Calculation at the Receiver**
- ✓ **Checksum in the Packet**

Checksum Calculation at the Sender

To create the checksum the sender does the following:

- The packet is divided into *k* sections, each of *n* bits (usually 16 bits).
- All sections are added together using one's complement arithmetic.
- The final result is complemented to make the checksum.

Checksum Calculation at the Receiver

- The receiver divides the received packet into *k* sections and adds all sections.
- It then complements the result.
- If the final result is 0, the packet is accepted; otherwise, it is rejected.

Figure 7.22 *Checksum concept*

Figure 7.23 *Checksum in one's complement arithmetic*

Checksum Calculation in IP Packet

- The implementation of the checksum in the IP packet follows the same principles discussed above.
- First, the value of the checksum field is set to 0.
- Then, the entire header is divided into 16-bit sections and added together.
- The result (sum) is complemented and inserted into the checksum field.

Checksum in IP covers only the header, not the data.

Example 7.17

Figure 7.24 shows an example of a checksum calculation at the sender site for an IP header without options. The header is divided into 16-bit sections.

Sol:.

- **All the sections are added and the sum is complemented.**
- **The result is inserted in the checksum field.**

Figure 7.24 *Example of checksum calculation at the sender*

4, 5, and 0	01000101	00000000
28	00000000	00011100
1	00000000	00000001
0 and 0	00000000	00000000
4 and 17	00000100	00010001
0	00000000	00000000
10.12	00001010	00001100
14.5	00001110	00000101
12.6	00001100	00000110
7.9	00000111	00001001
Sum	01110100	01001110
Checksum	10001011	10110001

Example 7.18

Figure 7.25 shows the checking of checksum calculation at the receiver site (or intermediate router) assuming that no errors occurred in the header.

Sol:

- **The header is divided into 16-bit sections.**
- **All the sections are added and the sum is complemented.**
- **Since the result is 16 number of 0s, the packet is accepted.**

Figure 7.25 *Example of checksum calculation at the receiver*

4, 5, and 0	01000101	00000000
28	00000000	00011100
1	00000000	00000001
0 and 0	00000000	00000000
4 and 17	00000100	00010001
Checksum	10001011	10110001
10.12	00001010	00001100
14.5	00001110	00000101
12.6	00001100	00000110
7.9	00000111	00001001
Sum	1111 1111	1111 1111
Checksum	0000 0000	0000 0000

4	5	0	28	
1			0	0
4		17	35761	
10.12.14.5				
12.6.7.9				

Next is UDP…..

Chapter 14

User Datagram Protocol

Chapter Outline

14.1 Introduction

14.2 User Datagram

14.3 UDP Services

14.4 UDP Application

Figure 14.1 *Position of UDP in the TCP/IP protocol suite*

14-2 USER DATAGRAM

UDP packets, called user datagrams, have a fixed-size header of 8 bytes.

a. UDP user datagram

b. Header format

Figure 14.2 ***User datagram format***

14-2 USER DATAGRAM

- **Source port number.**
  - Used by the process running on the source host.
  - It is 16 bits long, which means that the port number can range from 0 to 65,535.
- **Destination port number.**
  - Used by the process running on the destination host.
  - It is 16 bits long.
- **Length.**
  - 16 bits can define a total length of 0 to 65,535 bytes.
  - However, the total length needs to be much less because a UDP user datagram is stored in an IP datagram with the total length of 65,535 bytes.
- **Checksum**
  - Used to detect errors over the entire user datagram.

UDP length = IP length – IP header's length

Well-known ports from client to server

Table 14.1 *Well-known Ports used with UDP*

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Domain	Domain Name Service (DNS)
67	Bootps	Server port to download bootstrap information
68	Bootpc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

Example 14.1

The following is a dump of a UDP header in hexadecimal format.

CB84000D001C001C

a. What is the source port number?
b. What is the destination port number?
c. What is the total length of the user datagram?
d. What is the length of the data?
e. Is the packet directed from a client to a server or vice versa?
f. What is the client process?

Example 14.1 *Continued*

Solution

CB84000D001C001C

a. The source port number is the first four hexadecimal digits $(CB84)_{16}$ or 52100.

b. The destination port number is the second four hexadecimal digits $(000D)_{16}$ or 13.

c. The third four hexadecimal digits $(001C)_{16}$ define the length of the whole UDP packet as 28 bytes.

d. The length of the data is the length of the whole packet minus the length of the header, or 28 – 8 = 20 bytes.

e. Since the destination port number is 13 (a well-known port), the packet is from the client to the server.

f. The client process is the Daytime (see Table 14.1).

Connectionless Service

- **Connectionless service means that each user datagram sent by UDP is an independent datagram.**
- **There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program.**
- **The user datagrams are not numbered.**
- **Also, there is no connection establishment and no connection termination as in the case for TCP.**
- **This means that each user datagram can travel on a different path.**

Pseudoheader is the part of the header of the IP packet in which the user datagram is encapsulated.

Figure 14.3 ***Pseudoheader for checksum calculation***

NOTE

If the checksum does not include the pseudo-header, a user datagram may arrive safe and sound.

However, if the IP header is corrupted, it may be delivered to the wrong host.

Example 14.2

Figure 14.4 shows the checksum calculation for a very small user datagram with only 7 bytes of data. Because the number of bytes of data is odd, padding is added for checksum calculation. The pseudoheader as well as the padding will be dropped when the user datagram is delivered to IP

153.18.8.105			
171.2.14.10			
All 0s	17	15	
1087		13	
15		All 0s	
T	E	S	T
I	N	G	Pad

Figure 14.4 *Checksum calculation for a simple UDP user datagram*

10011001 00010010	→	153.18
00001000 01101001	→	8.105
10101011 00000010	→	171.2
00001110 00001010	→	14.10
00000000 00010001	→	0 and 17
00000000 00001111	→	15
00000100 00111111	→	1087
00000000 00001101	→	13
00000000 00001111	→	15
00000000 00000000	→	0 (checksum)
01010100 01000101	→	T and E
01010011 01010100	→	S and T
01001001 01001110	→	I and N
01000111 00000000	→	G and 0 (padding)
10010110 11101011	→	Sum
01101001 00010100	→	Checksum

Figure 14.5 *Encapsulation and decapsulation*

Figure 14.6 *Queues in UDP*

If incoming queue is full, UDP send ICMP port not reachable message to sender and drop the packet

Figure 14.7 *Multiplexing and demultiplexing*

Note

UDP is an example of the connectionless simple protocol (without flow control, error control and congestion control mechanism) with the exception of an optional checksum added to packets for error detection.

14-4 UDP APPLICATION

Even though UDP is unreliable (No flow control, error control and congestion control mechanism), UDP is preferable for some applications.

Example 14.4

A client-server application such as DNS uses the services of UDP because

- **a client needs to send a short request to a server and to receive a quick response from it. The request and response can each fit in one user datagram. Since only one message is exchanged in each direction, the connectionless feature is not an issue; the client or server does not worry that messages are delivered out of order.**

Example 14.5

A client-server application such as SMTP. which is used in electronic mail, cannot use the services of UDP because a user can send a long e-mail message, which may include multimedia (images, audio, or video).

- **If the application uses UDP and the message does not fit in one single user datagram, the message must be split by the application into different user datagrams. Here the connectionless service may create problems. The user datagrams may arrive and be delivered to the receiver application out of order. The receiver application may not be able to reorder the pieces. This means the connectionless service has a disadvantage for an application program that sends long messages.**

Example 14.6

Assume we are downloading a very large text file from the Internet.

- **We definitely need to use a transport layer that provides reliable service. We don't want part of the file to be missing or corrupted when we open the file. The delay created between the delivery of the parts are not an overriding concern for us; we wait until the whole file is composed before looking at it. In this case, UDP is not a suitable transport layer.**

Example 14.7

Assume we are watching a real-time stream video on our computer. Such a program is considered a long file; it is divided into many small parts and broadcast in real time.

- The parts of the message are sent one after another. If the transport layer is supposed to resend a corrupted or lost frame, the synchronizing of the whole transmission may be lost. The viewer suddenly sees a blank screen and needs to wait until the second transmission arrives. This is not tolerable. However, if each small part of the screen is sent using one single user datagram, the receiving UDP can easily ignore the corrupted or lost packet and deliver the rest to the application program. That part of the screen is blank for a very short period of the time, which most viewers do not even notice. However, video cannot be viewed out of order, so streaming audio, video, and voice applications that run over UDP must reorder or drop frames that are out of sequence (using RTP).

The following shows some typical applications that can benefit more from the services of UDP than from those of TCP.

- UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control. It is not usually used for a process such as FTP that needs to send bulk data.
- UDP is suitable for a process with internal flow and error-control mechanisms.
- UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.
- UDP is used for management processes such as SNMP.
- UDP is used for some route updating protocols such as Routing Information Protocol (RIP)
- UDP is normally used for real-time applications that cannot tolerate uneven delay between sections of a received message.
- Examples: time series applications, online gaming, video conferencing, Voice over IP (VoIP), DNS

- Next is TCP…..

Chapter 15

Transmission Control Protocol (TCP)

Chapter Outline

15.1 TCP Services
15.2 TCP Features
15.3 Segment
15.4 A TCP Connection

Figure 15.1 *TCP/IP protocol suite*

Table 15.1 *Well-known Ports used by TCP*

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20 and 21	FTP	File Transfer Protocol (Data and Control)
23	TELNET	Terminal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol

C → S

Figure 15.2 *Stream delivery*

Figure 15.3 *Sending and receiving buffers*

Figure 15.4 *TCP segments*

15-3 SEGMENT

Before discussing TCP in more detail, let us discuss the TCP packets themselves. A packet in TCP is called a segment.

Topics Discussed in the Section

- ✓ **Format**
- ✓ **Encapsulation**

Figure 15.5 *TCP segment format*

Figure 15.6 *Control field*

URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection

Figure 15.7 *Pseudoheader added to the TCP segment*

Note

The use of the checksum in TCP is mandatory.

Note

The bytes of data being transferred in each connection are numbered by TCP.

The numbering starts with an arbitrarily generated number.

Example 15.1

Suppose a TCP connection is transferring a file of 5,000 bytes. The first byte is numbered 10,001. What are the sequence numbers for each segment if data are sent in five segments, each carrying 1,000 bytes?

Solution

The following shows the sequence number for each segment:

Segment 1	→	Sequence Number:	10,001	Range:	10,001	to	11,000
Segment 2	→	Sequence Number:	11,001	Range:	11,001	to	12,000
Segment 3	→	Sequence Number:	12,001	Range:	12,001	to	13,000
Segment 4	→	Sequence Number:	13,001	Range:	13,001	to	14,000
Segment 5	→	Sequence Number:	14,001	Range:	14,001	to	15,000

Note

The value in the sequence number field of a segment defines the number assigned to the first data byte contained in that segment.

Note

The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive.

The acknowledgment number is cumulative.

Figure 15.8 *Encapsulation*

15-4 A TCP CONNECTION

TCP is connection-oriented. It establishes a virtual path between the source and destination.

All of the segments belonging to a message are then sent over this virtual path. You may wonder how TCP, which uses the services of IP, a connectionless protocol, can be connection-oriented. The point is that a TCP connection is virtual, not physical.

TCP operates at a higher level. TCP uses the services of IP to deliver individual segments to the receiver, but it controls the connection itself. If a segment is lost or corrupted, it is retransmitted.

A TCP CONNECTION PHASES

- ✓ **Connection Establishment**
- ✓ **Data Transfer**
- ✓ **Connection Termination**
- ✓ **Connection Reset**

Figure 15.9 *Connection establishment using three-way handshake*

Means "no data" !

seq: 8001 if piggybacking

Note

A SYN segment cannot carry data, but it consumes one sequence number.

Note

A SYN + ACK segment cannot carry data, but does consume one sequence number.

Note

An ACK segment, if carrying no data, consumes no sequence number.

Figure 15.10 *Data Transfer*

Figure 15.11 *Connection termination using three-way handshake*

Note

The FIN segment consumes one sequence number if it does not carry data.

Note

The FIN + ACK segment consumes one sequence number if it does not carry data.

Figure 15.12 *Half-Close*

15-5 STATE TRANSITION DIAGRAM

To keep track of all the different events happening during connection establishment, connection termination, and data transfer, TCP is specified as the finite state machine shown in Figure 15.13.

Figure 15.13 *State transition diagram*

Table 15.2 *States for TCP*

State	*Description*
CLOSED	No connection exists
LISTEN	Passive open received; waiting for SYN
SYN-SENT	SYN sent; waiting for ACK
SYN-RCVD	SYN+ACK sent; waiting for ACK
ESTABLISHED	Connection established; data transfer in progress
FIN-WAIT-1	First FIN sent; waiting for ACK
FIN-WAIT-2	ACK to first FIN received; waiting for second FIN
CLOSE-WAIT	First FIN received, ACK sent; waiting for application to close
TIME-WAIT	Second FIN received, ACK sent; waiting for 2MSL time-out
LAST-ACK	Second FIN sent; waiting for ACK
CLOSING	Both sides decided to close simultaneously

15-6 WINDOWS IN TCP

TCP uses two windows (send window and receive window) for each direction of data transfer, which means four windows for a bidirectional communication.

To make the discussion simple, we make an assumption that communication is only unidirectional;

Figure 15.22 *Send window in TCP*

a. Send window

b. Opening, closing, and shrinking send window

Figure 15.23 *Receive window in TCP*

a. Receive window and allocated buffer

b. Opening and closing of receive window

ICMP
&
IGMP

9-1 INTRODUCTION

- **The IP protocol has no error-reporting or error-correcting mechanism.**
- **What happens if something goes wrong?**
- **What happens if a router must discard a datagram because it cannot find a router to the final destination, or because the time-to-live field has a zero value?**
- **These are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.**

Figure 9.1 *Position of ICMP in the network layer*

Internet Control Message Protocol (ICMP)

Figure 9.2 *ICMP encapsulation*

9-2 MESSAGES

- **ICMP messages are divided into two broad categories: error-reporting messages and query messages.**
- **The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.**
- **The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host.**
- **Also, hosts can discover and learn about routers on their network and routers can help a node redirect its messages.**

Figure 9.3 *General format of ICMP messages*

Table 9.1 *ICMP messages*

Category	*Type*	*Message*
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

ICMP Parameter Message Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Type								Code								Checksum															
0								1								2								3							
Pointer								Unused																							
4								5								6								7							
Internet Header + 8 bytes of Original Data Datagram																															
8																															

Type	Code	Meaning
0	0	Echo Reply
3	0	Net Unreachable
	1	Host Unreachable
	2	Protocol Unreachable
	3	Port Unreachable
	4	Frag needed and DF set
	5	Source route failed
	6	Dest network unknown
	7	Dest host unknown
	8	Source host isolated
	9	Network admin prohibited
	10	Host admin prohibited
	11	Network unreachable for TOS
	12	Host unreachable for TOS
	13	Communication admin prohibited
4	0	Source Quench (Slow down/Shut up)

Type	Code	Meaning
5	0	Redirect datagram for the network
	1	Redirect datagram for the host
	2	Redirect datagram for the TOS & Network
	3	Redirect datagram for the TOS & Host
8	0	Echo
9	0	Router advertisement
10	0	Router selection
11	0	Time To Live exceeded in transit
	1	Fragment reassemble time exceeded
12	0	Pointer indicates the error (Parameter Problem)
	1	Missing a required option (Parameter Problem)
	2	Bad length (Parameter Problem)
13	0	Time Stamp
14	0	Time Stamp Reply
15	0	Information Request
16	0	Informaiton Reply
17	0	Address Mask Request
18	0	Address Mask Reply
30	0	Traceroute (Tracert)

Type	Code	Meaning
0	0	Echo Reply
3	0	Net Unreachable
	1	Host Unreachable
	2	Protocol Unreachable
	3	Port Unreachable
	4	Frag needed and DF set
	5	Source route failed
	6	Dest network unknown
	7	Dest host unknown
	8	Source host isolated
	9	Network admin prohibited
	10	Host admin prohibited
	11	Network unreachable for TOS
	12	Host unreachable for TOS
	13	Communication admin prohibited
4	0	Source Quench (Slow down/Shut up)

Type	Code	Meaning
5	0	Redirect datagram for the network
	1	Redirect datagram for the host
	2	Redirect datagram for the TOS & Network
	3	Redirect datagram for the TOS & Host
8	0	Echo
9	0	Router advertisement
10	0	Router selection
11	0	Time To Live exceeded in transit
	1	Fragment reassemble time exceeded
12	0	Pointer indicates the error (Parameter Problem)
	1	Missing a required option (Parameter Problem)
	2	Bad length (Parameter Problem)
13	0	Time Stamp
14	0	Time Stamp Reply
15	0	Information Request
16	0	Informaiton Reply
17	0	Address Mask Request
18	0	Address Mask Reply
30	0	Traceroute (Tracert)

Figure 9.4 *Error-reporting messages*

The following are important points about ICMP error messages:

- No ICMP error message will be generated in response to a datagram carrying an ICMP error message.
- No ICMP error message will be generated for a fragmented datagram that is not the first fragment.
- No ICMP error message will be generated for a datagram having a multicast address.
- No ICMP error message will be generated for a datagram having a special address such as 127.0.0.0 or 0.0.0.0.

12-3 IGMP: Internet Group Management Protocol

- **Multicast communication means that a sender sends a message to a group of recipients that are members of the same group.**
- **Each multicast router needs to know the list of groups that have at least one loyal member related to each interface.**
- **Collection of this type of information is done at two levels: locally and globally.**
- **The first task is done by the IGMP protocol; the second task is done by the multicast routing protocols.**

Figure 12.6 *Position of IGMP in the network layer*

Internet Group Management Protocol (IGMP)

Note

IGMP is a group management protocol. It helps a multicast router create and update a list of loyal members related to each router interface.

Figure 12.7 *IGMP messages*

MQ is sent by the router to find active group members in the network.

Figure 12.8 *Membership query message format*

S: suppress flag
QRV: querier's robustness variable
QQIC: querier's query interval code

Figure 12.9 *Three forms of query messages*

Figure 12.10 *Membership report message format*

Table 12.3 *Record Type*

Category	*Type*	*Type Value*
Current-State-Record	Mode_Is_Include	1
	Mode_Is_Exclude	2
Filter-Mode-Change-Record	Change_To_Include_Mode	3
	Change_To_Exclude_Mode	4
Source-List-Change-Record	Allow_New_Sources	5
	Block_Old_Sources	6

Host Configuration: Dynamic Host Configuration Protocol (DHCP)

18-1 INTRODUCTION

- **Each computer that uses the TCP/IP protocol suite needs to know its IP address.**
- **If the computer uses classless addressing or is a member of a subnet, it also needs to know its subnet mask.**
- **Most computers today need two other pieces of information:**
  - **the address of a default router to be able to communicate with other networks and**
  - **the address of a name server to be able to use names instead of addresses.**

18-2 DHCP OPERATION

The DHCP client and server can either be on the same network or on different networks. Let us discuss each situation separately.

Figure 18.1 *Client and server on the same network*

Same Network

Figure 18.2 *Client and server on two different networks*

Different Network

Figure 18.3 *Use of UDP ports*

Figure 18.4 *DHCP packet format*

Figure 18.5 *Flag format*

Figure 18.6 *Option format*

Table 18.1 *Options for DHCP*

Tag	*Length*	*Value*	*Description*
0			Padding
1	4	Subnet mask	Subnet mask
2	4	Time of the day	Time offset
3	Variable	IP addresses	Default router
4	Variable	IP addresses	Time server
5	Variable	IP addresses	IEN 16 server
6	Variable	IP addresses	DNS server
7	Variable	IP addresses	Log server
8	Variable	IP addresses	Quote server
9	Variable	IP addresses	Print server
10	Variable	IP addresses	Impress
11	Variable	IP addresses	RLP server
12	Variable	DNS name	Host name
13	2	Integer	Boot file size
53	1	Discussed later	Used for dynamic configuration
128–254	Variable	Specific information	Vendor specific
255			End of list

Routing

11-1 INTRODUCTION

An internet is a combination of networks connected by routers. When a datagram goes from a source to a destination, it will probably pass through many routers until it reaches the router attached to the destination network.

- **Cost/Metrics**
- **Static and Dynamic routing**

11-2 INTER- AND INTRA-DOMAIN ROUTING

- **Today, the internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers.**
- **For this reason, the internet is divided into autonomous systems.**
- **An autonomous system (AS) is a group of networks and routers under the authority of a single administration.**
- **Routing inside an autonomous system is called intra-domain routing.**
- **Routing between autonomous systems is called inter-domain routing.**

Figure 11.1 *Autonomous systems*

Figure 11.2 *Popular routing protocols*

11-3 DISTANCE VECTOR ROUTING

- **This method sees an AS, with all routers and networks, as a graph, a set of nodes and lines (edges) connecting the nodes.**
- **A router can normally be represented by a node and a network by a link connecting two nodes, as an edge.**
- **Bellman-Ford is used to find the shortest path between nodes in a graph given the distance between nodes.**

Figure 11.3 *A graph for Bellman-Ford algorithm*

The algorithm is based on the fact that if all neighbors of node *i* know the shortest distance to node *j*, then the shortest distance between node *i* and *j* can be found by adding the distance between node *i* and each neighbor to the neighbor's shortest distance to node *j* and then select the minimum,

Figure 11.4 *The fact behind Bellman-Ford algorithm*

$$D_{ij} = \text{minimum } \{(c_{i1} + D_{1j}), (c_{i2} + D_{2j}), \ldots (c_{iN} + D_{Nj})$$

1. The shortest distance and the cost between a node and itself is initialized to 0.
2. The shortest distance between a node and any other node is set to infinity. The cost between a node and any other node should be given (can be infinity if the nodes are not connected).
3. The algorithm repeat as shown in Figure 11.4 until there is no more change in the shortest distance vector.

Table 11.1 *Bellman-Ford Algorithm*

```
Bellman_Ford ( )
{
   // Initialization
   for (i = 1 to N; for j = 1 to N)
   {
       if(i == j)  D_ij = 0   c_ij = 0
       else        D_ij = ∞   c_ij = cost between i and j
   }
   // Updating
   repeat
   {
      for (i = 1 to N; for j = 1 to N)
      {
          D_ij ← minimum [(c_i1 + D_1j)  ...  (c_iN + D_Nj)]
      } // end for
   } until (there was no change in previous iteration)
} // end Bellman-Ford
```

Table 11.2 *Distance Vector Algorithm Used by Each Router*

```
Distance_Vector_Algorithm ( )
{
     // At startup
    for (i = 1 to N)            // N is number of ports
    {
        Table_i.dest = address of the attached network
        Table_i.cost = 1
        Table_i.next = —        // Means at home
        Send a record R about each row to each neighbor
    } // end for loop

      // Updating
    repeat (forever)
    {
        Wait for arrival of a record R from a neighbor
        Update (R, T)           // Call update module
        for (i = 1 to N)        // N is the current table size
```

Table 11.2 *Distance Vector Algorithm Used by Each Router (continued)*

```
        {
                Send a record R about each row to each neighbor
        }
      }  // end repeat

}  // end Distance_Vector
Update (R, T)            // Update module
{
    Search T for a destination matching the one in R
    if (destination is found in row i)
    {
        if (R.cost + 1 < Ti.cost    or    R.next == Ti.next)
        {
                Ti.cost = R.cost + 1
                Ti.next = Address of sending router
        }
```

Address of the sender of R

Table 11.2 *Distance Vector Algorithm Used by Each Router (continued)*

```
        else  discard  the  record      // No change is needed
        }
        else
         // Insert the new router
        {
            T_{N+1}.dest = R.dest
            T_{N+1}.cost = R.cost + 1
            T_{N+1}.next = Address of sending router
            Sort the table according to destination address
        }
}  // end of Update module
```

Updating Routing Table

- **If the next-node entry is different**
  - The receiving node chooses the row with the smaller cost
  - If there is a tie, the old one is kept
- **If the next-node entry is the same**
  - i.e. the sender of the new row is the provider of the old entry
  - The receiving node chooses the new row, even though the new value is infinity.

When to Share ?

- **Periodic Update**
  - A node sends its routing table, normally 30 seconds, in a periodic update
- **Triggered Update**
  - A node sends its routing table to its neighbors any time when there is a change in its routing table
    - 1. After updating its routing table, or
    - 2. Detects some failure in the neighboring links

Example 11.1

Figure 11.5 shows the initial routing table for an AS. Note that the figure does not mean that all routing tables have been created at the same time; each router creates its own routing table when it is booted.

Figure 11.5 *Example 11.1*

Example 11.2

Now assume router A sends four records to its neighbors, routers B, D, and C. Figure 11.6 shows the changes in B's routing table when it receives these records. We leave the changes in the routing tables of other neighbors as exercise.

Figure 11.6 *Example 11.2*

Routing Table B

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net6	1	—

After receiving record 1

Routing Table B

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net6	1	—

After receiving record 2

Routing Table B

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net4	2	A
Net6	1	—

After receiving record 3

Routing Table B

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net4	2	A
Net5	2	A
Net6	1	—

After receiving record 4

Example 11.3

Figure 11.7 shows the final routing tables for routers in Figure 11.5.

Figure 11.7 *Example 11.3*

A

Dest	Cost	Next
Net1	1	—
Net2	1	—
Net3	2	B
Net4	1	—
Net5	1	—
Net6	2	C
Net7	2	C

B

Dest	Cost	Next
Net1	2	A
Net2	1	—
Net3	1	—
Net4	2	A
Net5	2	A
Net6	1	—
Net7	2	C

C

Dest	Cost	Next
Net1	2	A
Net2	2	A
Net3	2	B
Net4	2	A
Net5	1	—
Net6	1	—
Net7	1	—

D

Dest	Cost	Next
Net1	2	A
Net2	2	A
Net3	3	A
Net4	1	—
Net5	1	A
Net6	3	A
Net7	3	A

E

Dest	Cost	Next
Net1	3	C
Net2	3	C
Net3	3	C
Net4	3	C
Net5	2	C
Net6	2	C
Net7	1	—

Figure 11.8 *Two-node instability*

Count to infinity problem

A decrease in cost (good news) propagates quickly, but any increase in cost (bad news) propagates slowly.

Two-Node Instability (1)

- Defining Infinity
  - Most implementations define 16 as infinity
- Split Horizon
  - Instead of flooding the table through each interface, each node sends only part of its table through each interface
  - E.g. node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A

11-5 LINK STATE ROUTING

Link state routing has a different philosophy from that of distance vector routing. In link state routing, if each node in the domain has the entire topology of the domain—the list of nodes and links, how they are connected including the type, cost (metric), and the condition of the links (up or down)—the node can use the Dijkstra algorithm to build a routing table.

Figure 11.17 *Concept of Link state routing*

Figure 11.18 *Link state knowledge*

L S

A

Node	Cost
B	3
C	2
D	5

A → D → E

	0	3	2	4	5	3
Next	A	B	C	D	E	F
A	$\boxed{0}$	∞	∞	∞	∞	∞
C		3	$\boxed{2}$	5	∞	∞
B		$\boxed{3}$		4	∞	3
F				4	7	$\boxed{3}$
D				$\boxed{4}$	5	
E						

Shortest path tree for "A"

Dest	Cost	Next
A	0	—
B	3	B
C	2	C
D	4	C
E	5	C
F	3	C

Building Routing Tables

- Creation of the states of the links by each node, called the link state packets (LSP)
- Dissemination of LSPs to every other routers, called flooding (efficiently)
- Formation of a shortest path tree for each node
- Calculation of a routing table based on the shortest path tree

Creation of LSP

- LSP data: E.g. the node ID, the list of links, a sequence number, and age.
- LSP Generation
  - When there is a change in the topology of the domain
  - On a periodic basis
    - There is no actual need for this type of LSP, normally 60 minutes or 2 hours

Table 11.3 *Dijkstra's Algorithm*

```
Dijkstra ( )
{
    // Initialization
    Path = {s}              // s means self
    for (i = 1 to N)
    {
        if(i is a neighbor of s and i ≠ s)    D_i = c_si
        if (i is not a neighbor of s)         D_i = ∞
    }
    D_s = 0

} // Dijkstra
```

Continued

```
// Iteration
Repeat
{
    // Finding the next node to be added
 Path = Path ∪ i   if D_i is minimum among all remaining nodes

    // Update the shortest distance for the rest
 for (j = 1 to M)     // M number of remaining nodes
 {
        D_j = minimum (D_j ,  D_j  + c_ij)
  }
} until (all nodes included in the path, M = 0)
```

Figure 11.19 *Forming shortest path three for router A in a graph*

Topology

Legend

Initialization

Figure 11.19 *Continued*

Iteration 1

Iteration 2

Iteration 3

Figure 11.19 *Continued*

Iteration 4

Iteration 5

Iteration 6

Example 11.6

To show that the shortest path tree for each node is different, we found the shortest path tree as seen by node C (Figure 11.20). We leave the detail as an exercise.

Figure 11.20 *Example 11.6*

Table 11.4 *Routing Table for Node A*

Destination	*Cost*	*Next Router*
A	0	—
B	2	—
C	7	B
D	3	—
E	6	B
F	8	B
G	9	B

Routing-BF

22 November 2021 22:40

DVR → Routing info of every node in the N/W.
- → N/W Addr. →
- → Next hop →
- → cost →
- → no of hop → hop count

A → A

$2+3+(-6) = -1$

BF →

→ $d(i) + c(i,j) < d(j)$

replace

$d(j) = d(i) + c(i,j)$.

(*)

edge	cost
S→A	4
S→B	6
A→C	3
~~B→A~~	-5
B→D	1
C→D	2

Initial

Vertex	S	A	B	C	D
Dist	0	∞	∞	∞	∞
Prev	-	-	-	-	-

First iteration.

Vertex	S	A	B	C	D
Dist	0	4	6	∞	∞
Prev	-	S	S	-	-

Second iteration

V	S	A	B	C	D
D	0	1	6	7	7
P	-	B	S	A	B

Third iteration

V	S	A	B	C	D
D	0	1	6	4	6
P	-	B	S	A	C

V	S	A	B	C	D
D	0	1	6	4	6
P	-	B	S	A	C

4th

4th

V	S	A	B	C	D
D	0	1	6	4	6
Pn	-	B	S	A	C

Shortest path Tree

Cost & Shortest path tree

S

V	Shortest Dist	Path.
S→A	1	S→B→A
S→B	6	S→B
S→C	4	S→B→A→C
S→D	6	S→B→A→C→D

S → shortest path tree + cost table

do the same for all nodes