USN

Internal Assessment Test 2 – December 2024

Sub:	Computer Networks				Sub Code:	BCS502	Branch:	CSE
Date:	14.12.2024	Duration:	90 mins	Max Marks:	50	Sem / Sec:	V (A, B & C)	OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1 (a)	Explain classful addressing system in IPV4 with a neat diagram.	6	CO3	L2
1 (b)	Illustrate the working of DHCP with suitable diagram.	4	CO3	L3
2 (a)	With a neat diagram explain TCP connection establishment and Connection termination Phase using three-way handshake.	6	CO3	L2
2 (b)	Describe the general services provided by UDP. With a neat diagram explain sending and receiving buffers in TCP.	2 + 2	CO3	L2
3(a)	Write a short note on DNS recursive and Iterative resolutions.	6	CO4	L2
3(b)	Discuss the IPV6 packet format.	4	CO4	L2

CI **CCI** **HOD**

USN

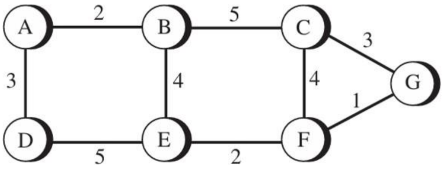
Internal Assessment Test 2 – December 2024

Sub:	Computer Networks				Sub Code:	BCS502	Branch:	CSE
Date:	14.12.2024	Duration:	90 mins	Max Marks:	50	Sem / Sec:	V (A, B & C)	OBE

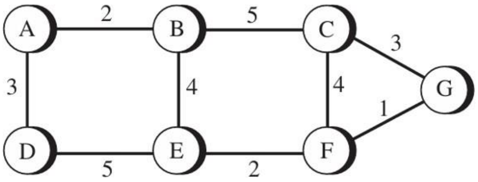
Answer any FIVE FULL Questions

		MARKS	CO	RBT
1 (a)	Explain classful addressing system in IPV4 with a neat diagram.	6	CO3	L2
1 (b)	Illustrate the working of DHCP with suitable diagram.	4	CO3	L3
2 (a)	With a neat diagram explain TCP connection establishment and Connection termination Phase	6	CO3	L2
2 (b)	Describe the general services provided by UDP. With a neat diagram explain sending and receiving buffers in TCP	2 + 2	CO3	L2
3(a)	Write a short note on DNS recursive and Iterative resolutions.	6	CO4	L2
3(b)	Discuss the IPV6 packet format.	4	CO4	L2

CI **CCI** **HOD**

4(a)	Explain Persistent and non-persistent http in detail with suitable diagram.	5	CO4	L2
4(b)	Demonstrate socket implementation for a client and server using TCP.	5	CO4	L3
5	Apply the distance vector algorithm for the given graph below to find the path from source node A to all other nodes. Mention the Bellmanford Equation.  A 2 B 5 C 3 3 4 4 G 1 D 5 E 2 F	10	CO4	L3
6(a)	Differentiate client server paradigm and peer-to-peer paradigm.	4	CO3	L2
6 (b)	Suppose Alice, with a Web based E-mail account and sends a message to Bob, who accesses his mail from his mail server using POP3. Discuss how the message gets from Alice's host to Bob's Host with suitable diagram.	6	CO3	L2

CI **CCI** **HOD**

4(a)	Explain Persistent and non-persistent http in detail with suitable diagram.	5	CO4	L2
4(b)	Demonstrate socket implementation for a client and server using TCP.	5	CO4	L3
5	Apply the distance vector algorithm for the given graph below to find the path from source node A to all other nodes. Mention the Bellmanford Equation.  A 2 B 5 C 3 3 4 4 G 1 D 5 E 2 F	10	CO4	L3
6(a)	Differentiate client server paradigm and peer-to-peer paradigm.	4	CO3	L2
6 (b)	Suppose Alice, with a Web based E-mail account and sends a message to Bob, who accesses his mail from his mail server using POP3. Discuss how the message gets from Alice's host to Bob's Host with suitable diagram.	6	CO3	L2

CI **CCI** **HOD**

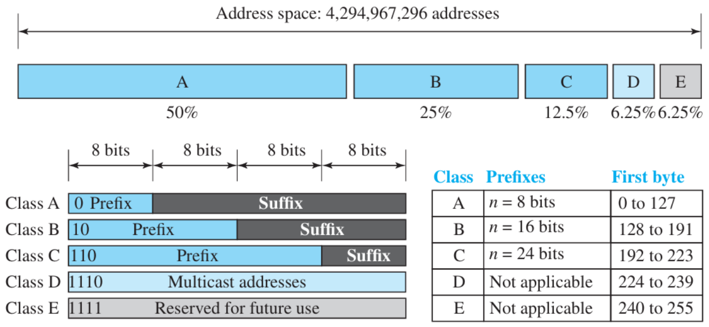
USN □□□□□□□□□□

Internal Assessment Test 2 – December 2024

Sub:	Computer Networks				Sub Code:	BCS502	Branch:	CSE
Date:	14.12.2024	Duration:	90 mins	Max Marks:	50	Sem / Sec:	V (A, B & C)	OBE

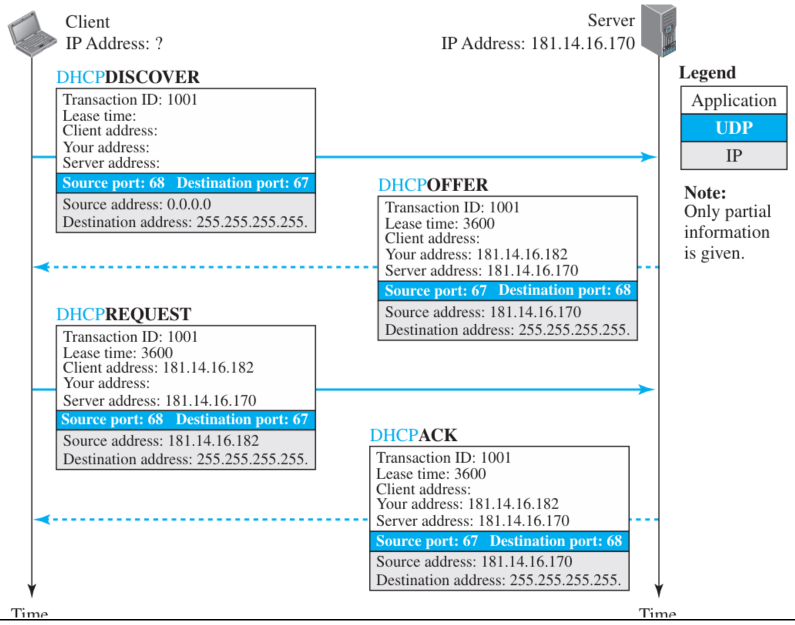
	Answer any FIVE FULL Questions	MARKS	CO	RBT
1 (a)	Explain classful addressing system in IPV4 with a neat diagram.	6	CO3	L2

Figure 18.18 *Occupation of the address space in classful addressing*



Class	Prefixes	First byte
A	n = 8 bits	0 to 127
B	n = 16 bits	128 to 191
C	n = 24 bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

1 (b)	Illustrate the working of DHCP with suitable diagram.	4	CO3	L3

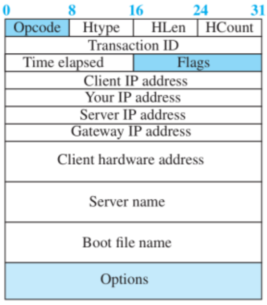
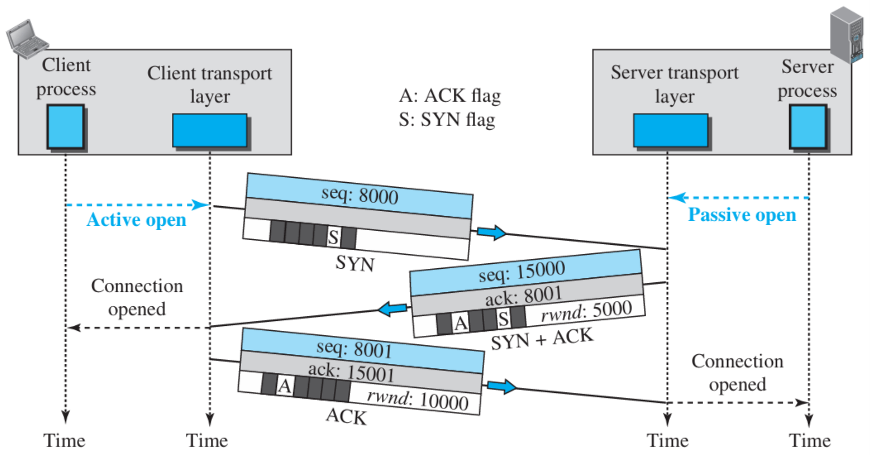
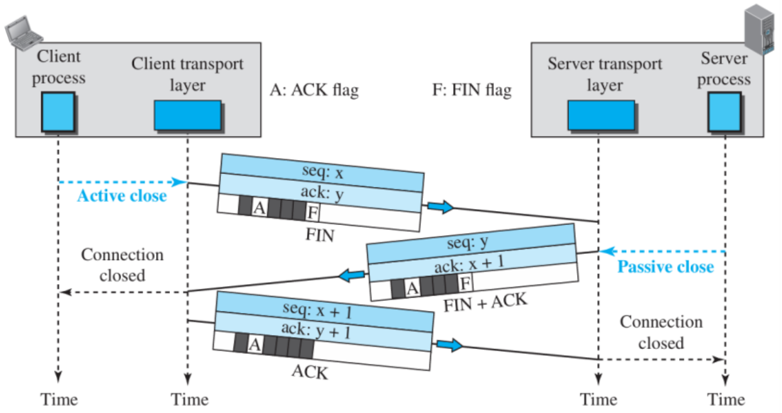


DHCPDISCOVER
Transaction ID: 1001
Lease time:
Client address:
Your address:
Server address:
Source port: 68 Destination port: 67
Source address: 0.0.0.0
Destination address: 255.255.255.255.

DHCPOFFER
Transaction ID: 1001
Lease time: 3600
Client address:
Your address: 181.14.16.182
Server address: 181.14.16.170
Source port: 67 Destination port: 68
Source address: 181.14.16.170
Destination address: 255.255.255.255.

DHCPREQUEST
Transaction ID: 1001
Lease time: 3600
Client address: 181.14.16.182
Your address:
Server address: 181.14.16.170
Source port: 68 Destination port: 67
Source address: 181.14.16.182
Destination address: 255.255.255.255.

DHCPACK
Transaction ID: 1001
Lease time: 3600
Client address:
Your address: 181.14.16.182
Server address: 181.14.16.170
Source port: 67 Destination port: 68
Source address: 181.14.16.170
Destination address: 255.255.255.255.

	Figure 18.25 *DHCP message format*  0 8 16 24 31 Opcode Htype HLen HCount Transaction ID Time elapsed Flags Client IP address Your IP address Server IP address Gateway IP address Client hardware address Server name Boot file name Options Fields: Opcode: Operation code, request (1) or reply (2) Htype: Hardware type (Ethernet, ...) HLen: Length of hardware address HCount: Maximum number of hops the packet can travel Transaction ID: An integer set by the client and repeated by the server Time elapsed: The number of seconds since the client started to boot Flags: First bit defines unicast (0) or multicast (1); other 15 bits not used Client IP address: Set to 0 if the client does not know it Your IP address: The client IP address sent by the server Server IP address: A broadcast IP address if client does not know it Gateway IP address: The address of default router Server name: A 64-byte domain name of the server Boot file name: A 128-byte file name holding extra information Options: A 64-byte field with dual purpose described in text • DHCP is often called as ***plug-and-play protocol***. • A network manager can configure DHCP to assign **permanent IP** addresses to the host and routers. DHCP can also be con figured to provide **temporary,** on demand, IP addresses to hosts			
2 (a)	With a neat diagram explain TCP connection establishment and Connection termination Phase using three-way handshake. Figure 24.10 *Connection establishment using three-way handshaking*  Client process Client transport layer A: ACK flag S: SYN flag Server transport layer Server process seq: 8000 S SYN Active open Passive open seq: 15000 ack: 8001 A S rwnd: 5000 SYN + ACK Connection opened seq: 8001 ack: 15001 A rwnd: 10000 ACK Connection opened Time Time Time Time Figure 24.12 *Connection termination using three-way handshaking*  Client process Client transport layer A: ACK flag F: FIN flag Server transport layer Server process seq: x ack: y A F FIN Active close seq: y ack: x + 1 A F FIN + ACK Passive close Connection closed seq: x + 1 ack: y + 1 A ACK Connection closed Time Time Time Time	6	CO3	L2
2 (b)	Describe the general services provided by UDP. With a neat diagram explain	2 + 2	CO3	L2

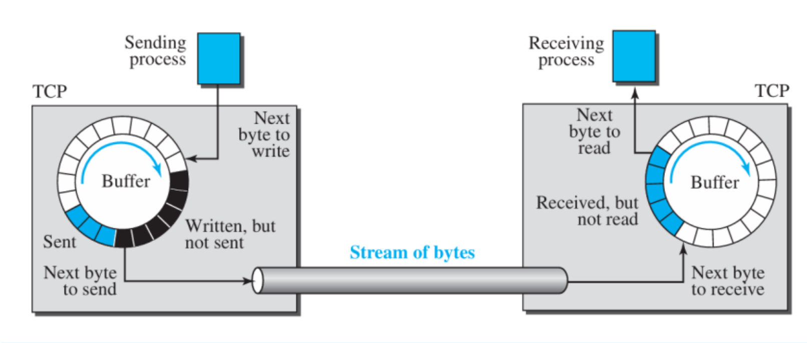
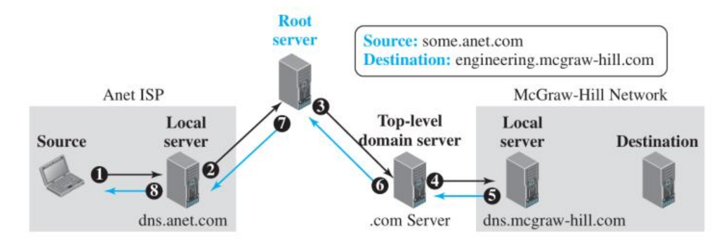
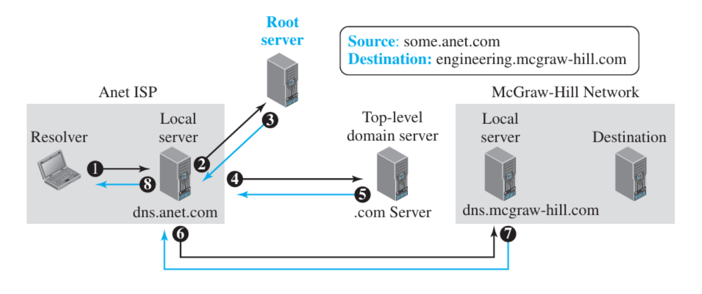
	sending and receiving buffers in TCP. General services provided by UDP: • Does not provide congestion control. • Performs encapsulation at the sender side and decapsulation at the receiver's side. • Some implementations create incoming and outgoing queues associated with each process. UDP provides multiplexing and demultiplexing **Figure 24.5** *Sending and receiving buffers*  Sending process TCP Next byte to write Buffer Written, but not sent Sent Next byte to send Stream of bytes Receiving process TCP Next byte to read Received, but not read Buffer Next byte to receive			
3(a)	Write a short note on DNS recursive and Iterative resolutions. • If the server has the information, it provides to the resolver. • otherwise, it either refers the resolver to other servers or asks other servers to provide the information. • After the resolver receives the mapping, it delivers the result to the process that requested it. Resolution can be of two types: • Recursive Resolution • Iterative Resolution **Figure 26.36** *Recursive resolution*  Root server Source: some.anet.com Destination: engineering.mcgraw-hill.com Anet ISP McGraw-Hill Network Source Local server Top-level domain server Local server Destination 1 2 3 4 5 6 7 8 dns.anet.com .com Server dns.mcgraw-hill.com **Figure 26.37** *Iterative resolution*  Root server Source: some.anet.com Destination: engineering.mcgraw-hill.com Anet ISP McGraw-Hill Network Resolver Local server Top-level domain server Local server Destination 1 2 3 4 5 6 7 8 dns.anet.com .com Server dns.mcgraw-hill.com	6	CO4	L2
3(b)	Discuss the IPV6 packet format.	4	CO4	L2

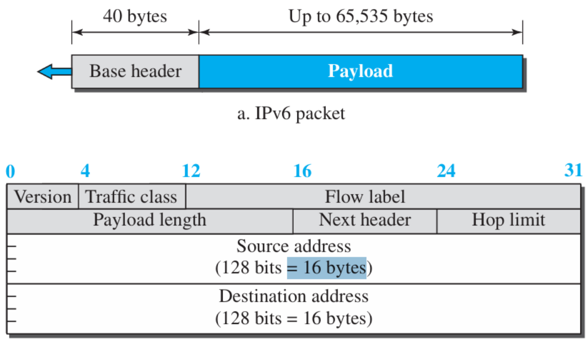
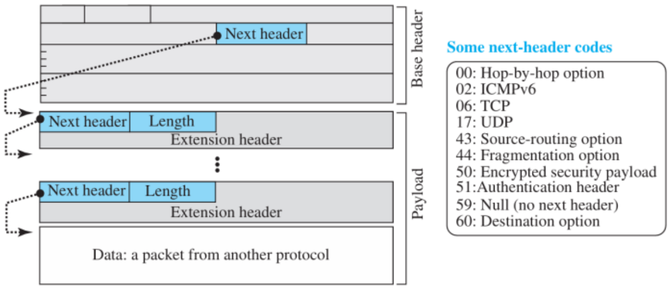
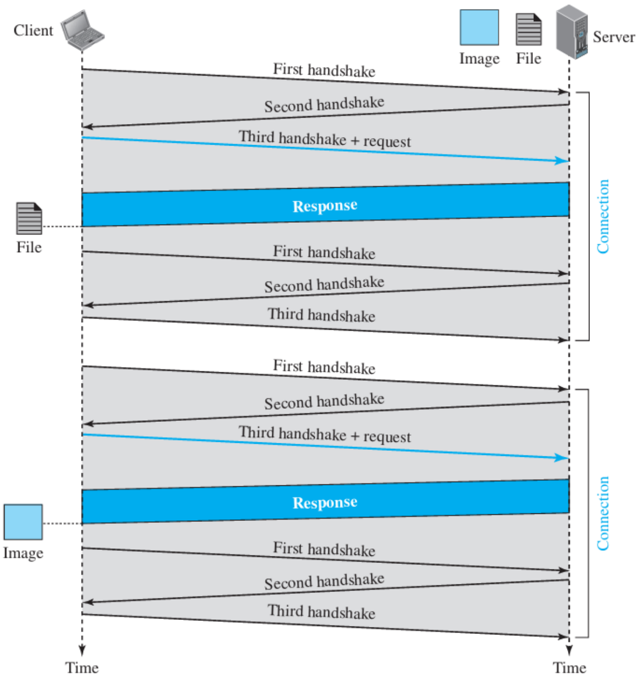
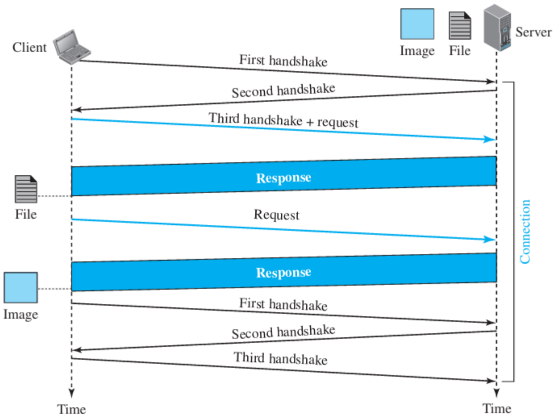
	Figure 22.6 *IPv6 datagram*  40 bytes Up to 65,535 bytes Base header Payload 0 4 12 16 24 31 Version Traffic class Flow label Payload length Next header Hop limit Source address (128 bits = 16 bytes) Destination address (128 bits = 16 bytes) a. IPv6 packet b. Base header **Figure 22.7** *Payload in an IPv6 datagram*  Next header Base header Next header Length Extension header Next header Length Extension header Payload Data: a packet from another protocol Some next-header codes 00: Hop-by-hop option 02: ICMPv6 06: TCP 17: UDP 43: Source-routing option 44: Fragmentation option 50: Encrypted security payload 51:Authentication header 59: Null (no next header) 60: Destination option • **Version.** The 4-bit version field defines the version number of the IP. For IPv6, the value is 6. • **Traffic Class**: Used to distinguish different payloads with different delivery requirements. • **Flow label**: Provides flow of the data. • **Payload Length**: Length of IP datagram excluding the header. • **Next header**: The next header describes the type of the data that follows the base header. • **Hop Limit.** Time to Live or hop count limit • **Source and Destination address** : The source address 16byte (128 bit) and destination address (128 bit) 16 bytes. • **Payload**: Describes the payload.			
4(a)	Explain persistent and non-persistent http in detail with suitable diagram. • To retrieve each object using a new TCP connection. (Non-Persistent Connection)	5	CO4	L2

Figure 26.3 *Example 26.3*



- To make a TCP connection and retrieve them all. (Persistent Connection)

Figure 26.4 *Example 26.4*



4(b)	Demonstrate socket implementation for a client and server using TCP.	5	CO4	L3

Program:

Server.java

```
import java.io.*;
import java.net.*;
public class server {
  public static void main(String[] args) {
     ServerSocket serverSocket = null;
     Socket clientSocket = null;
     try {
        // Create a server socket listening on port 9876
        serverSocket = new ServerSocket(9876);
        System.out.println("Server started. Waiting for a client...");
        // Accept client connection
```

```
            clientSocket = serverSocket.accept();
            System.out.println("Client connected!");
            // Input and output streams for client communication
            BufferedReader inFromClient = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
            PrintWriter outToClient = new PrintWriter(clientSocket.getOutputStream(),
true);
            // Read the file name requested by the client
            String fileName = inFromClient.readLine();
            System.out.println("Client requested file: " + fileName);
            // Attempt to open the requested file
            File file = new File(fileName);
            if (file.exists() && !file.isDirectory()) {
                // If the file exists, send back the file contents
                BufferedReader fileReader = new BufferedReader(new FileReader(file));
                String line;
                outToClient.println("FILE_FOUND");
                while ((line = fileReader.readLine()) != null) {
                    outToClient.println(line);
                }
                fileReader.close();
                System.out.println("File sent successfully.");
            } else {
                // If the file doesn't exist, inform the client
                outToClient.println("FILE_NOT_FOUND");
                System.out.println("Requested file not found.");
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            // Close the connections
            try {
                if (clientSocket != null) clientSocket.close();
                if (serverSocket != null) serverSocket.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Client.java

```
import java.io.*;
import java.net.*;
public class Client {
    public static void main(String[] args) {
        Socket socket = null;
        try {
            socket = new Socket("localhost", 9876);
            System.out.println("Connected to the server!");
          BufferedReader inFromServer = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
            PrintWriter outToServer = new PrintWriter(socket.getOutputStream(), true);
 BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in));
                    System.out.print("Enter the name of the file to request: ");
            String fileName = userInput.readLine();
            // Send the file name to the server
            outToServer.println(fileName);
            // Read the response from the server
            String serverResponse = inFromServer.readLine();
            if ("FILE_FOUND".equals(serverResponse)) {
                System.out.println("File found! Receiving content:");
                String line;
                while ((line = inFromServer.readLine()) != null) {
                    System.out.println(line);
                }
            } else if ("FILE_NOT_FOUND".equals(serverResponse)) {
                System.out.println("File not found on the server.");
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            // Close the connection
            try {
                if (socket != null) socket.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

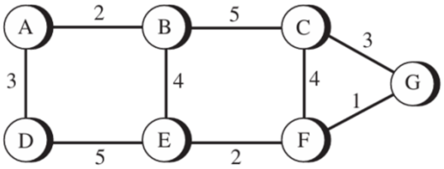
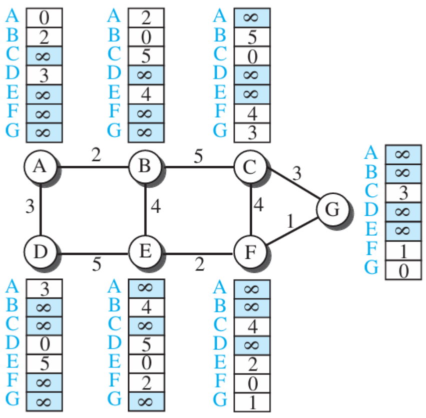
5	Apply the distance vector algorithm for the given graph below to find the path from source node A to all other nodes. Mention the Bellmanford Equation.  A B C G D E F 2 5 3 3 4 4 1 5 2 • Distance vector is a one-dimensional vector maintains name and distances of each node or router. • Distance vector is a one-dimensional vector maintains name and distances of each node or router.	10	CO4	L3

Figure 20.5 *The first distance vector for an internet*



	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C	∞	5	0	∞	∞	4	3
D	3	∞	∞	0	5	∞	∞
E	∞	4	∞	5	0	2	∞
F	∞	∞	4	∞	2	0	1
G	∞	∞	3	∞	∞	1	0

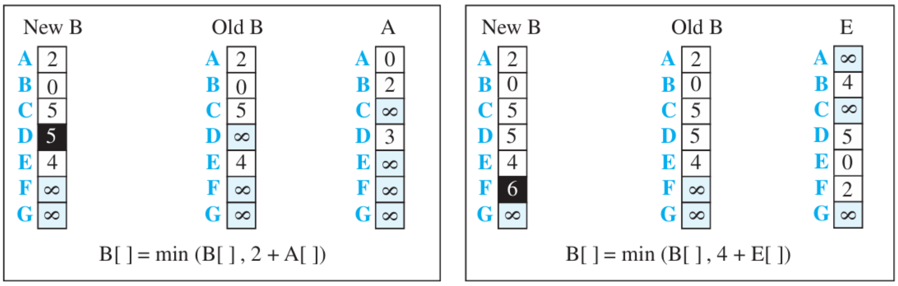
- After each node has created its vector, it sends a copy of vector to all its immediate neighbors.
- After a node receives a distance vector from its neighbor, it updates its distance vector using Bellman-Ford Equation.
- All the N nodes in the internet must be updated.

Bellman-Ford Equation:

$$D_{xy} = \min\{D_{xy}, (c_{xz} + D_{zy})\}$$

$$D_{xy} = \min\{(c_{xa} + D_{ay}), (c_{xb} + D_{by}), (c_{xc} + D_{cy}), \ldots\}$$

Figure 20.6 *Updating distance vectors*



	New B	Old B		A
A	2	2	A	0
B	0	0	B	2
C	5	5	C	∞
D	5	∞	D	3
E	4	4	E	∞
F	∞	∞	F	∞
G	∞	∞	G	∞

B[ ] = min (B[ ] , 2 + A[ ])

a. First event: B receives a copy of A's vector.

	New B	Old B		E
A	2	2	A	∞
B	0	0	B	4
C	5	5	C	∞
D	5	5	D	5
E	4	4	E	0
F	6	∞	F	2
G	∞	∞	G	∞

B[ ] = min (B[ ] , 4 + E[ ])

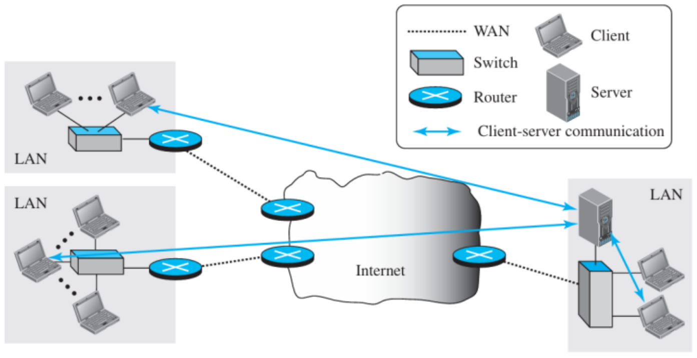
b. Second event: B receives a copy of E's vector.

Note:
X[ ]: the whole vector

6(a)	Differentiate client server paradigm and peer-to-peer paradigm. client server paradigm: • In this paradigm, the service provider is an application program called **server**. It runs continuously, waiting for another application program called **client**. • Some server process can provide a specific type of service to many clients. • The server must be running **all the time**.	4	CO3	L2

<table>
<tr>
<td></td>
<td>

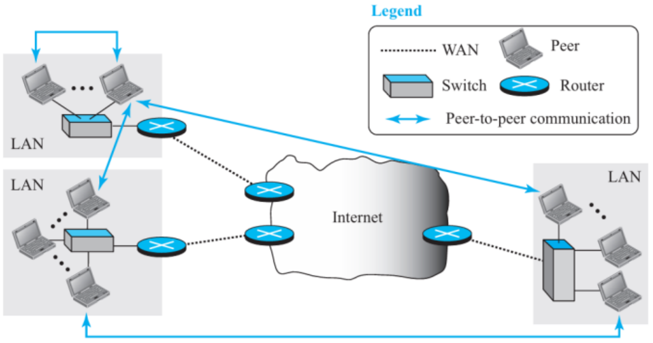
Figure 25.2 *Example of a client-server paradigm*



peer-to-peer paradigm:

- Often abbreviated as P2P paradigm, does not need for a server process to be running all the time.
- Responsibility is shared between the peers.

Figure 25.3 *Example of a peer-to-peer paradigm*



</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6 (b)</td>
<td>

Suppose Alice, with a Web based E-mail account and sends a message to Bob, who accesses his mail from his mail server using POP3. Discuss how the message gets from Alice’s host to Bob’s Host with suitable diagram.

- E-mail allows users to exchange messages.
- Considered as a one-way transaction.
- Uses some intermediate computers (servers).

The users run only client programs when they want and the intermediate servers apply the client/server paradigm

</td>
<td>6</td>
<td>CO3</td>
<td>L2</td>
</tr>
</table>

Figure 26.12 *Common scenario*

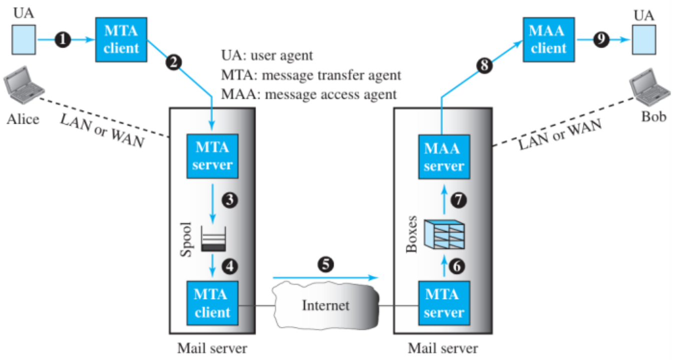


Figure 26.15 *Protocols used in electronic mail*

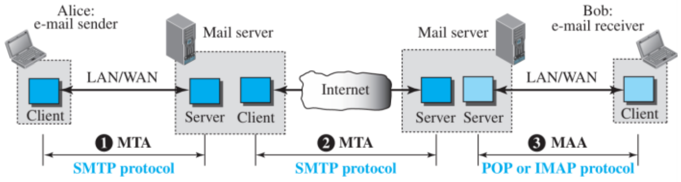
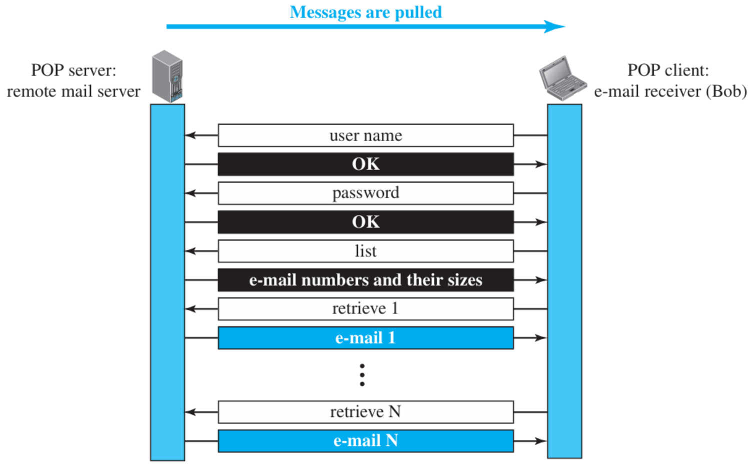


Figure 26.17 *POP3*


Messages are pulled
POP server: remote mail server
POP client: e-mail receiver (Bob)
user name
OK
password
OK
list
e-mail numbers and their sizes
retrieve 1
e-mail 1
retrieve N
e-mail N