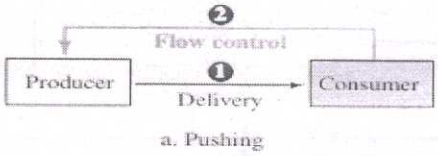
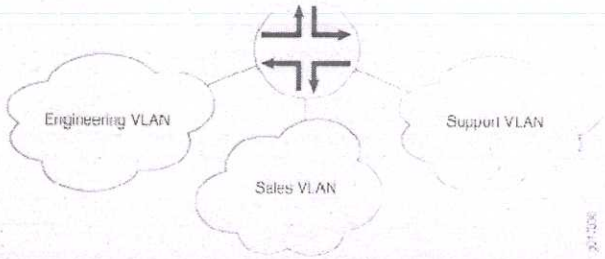
**Revision : 2015** **Course code(15) : 6131**

**Course Title : COMPUTER NETWORKS**

PART A

| Qst No. | Scoring Indicator | Split up score | Sub total | Total |
|---|---|---|---|---|
| I (1) | Interconnected collection of autonomous computers that intelligently share data,hardware& software | | 2 | |
| (2) | TCP/IP,ISO-OSI | | 2 | |
| (3) | Divide the addresses into several contiguous groups and assign each group to smaller networks( subnets) | | 2 | |
| (4) | If the sender delivers items whenever they are produced–without a prior request from the consumer–the delivery is referred to as pushing.   a. Pushing | 1 + 1 | 2 | |
| (5) | A repository of information in which the documents, called web pages, are distributed all over the world and related documents are linked together | | 2 | 10 |

PART B

| | | | | |
|---|---|---|---|---|
| II (1) | 1)LAN :connects computers that are physically close together ( < =1 km).high speed,owned by single organisation 2)WAN:connects computers that are physically far apart. "long-haul network".(100-1000km). 1)slower than a LAN. 3) MAN:it covers entire city,Larger than a LAN and smaller than a WAN(within 10km) | 3*2 | 6 | |
| (2) | domains based on logical groupings. Because the groupings are logical, the broadcast domains are not determined by the physical connectivity of the devices in the network. Hosts can be grouped according to a logical function, to limit the traffic broadcast within the VLAN to only the devices for which the traffic is intended. | 4 | 6 | |
| |   | 2 | | |
| (3) | Several mechanism have been used for this.1)Backpressure: It is a node to node congestion control that starts with a node and propagate,in the opposite direction of data flow,to the source 2) Choke pkt :In this warning is from router,which has encountered | 3 | 6 | |
| | 3)Implicit Signaling:In this there is no communication between the congested node or nodes and the source. The source guess that there is congestion somewhere in the network from other symptoms like delay of acknowledgement from receiver. 4)Explicit signaling:The node experiences congestion can explicitly send a signal to the source | 3 | | |

| | | | |
|---|---|---|---|
| (OR) | The HTTP protocol defines the format of the request and response messages.The first line in a request message is called a request line. There arethree fields in this line separated by one space and terminated by two characters (Carriage return and Line feed)The fields are called method, URL,and version.<br>The method field defines the request types. In version 1.1 of HTTP, several methods are defined. Most of the time, the client uses the GET method to send a request. In this case, the body of the message is empty. The<br>HEAD method is used when the client needs only some information about the web page from the server, such as the last time it was modified. It can also be used to test the validity of a URL. The response message in this case has only the header section; the body section is empty. The PUT method is the inverse of the GET method; it<br>allows the client to post a new web page on the serverThe POST method is similar to the PUT method, but it is used to send some information to the server to be added to the web page or to modify the web page. The TRACE method is used for debugging; the client asks the server to echo back the request to check<br>whether the server is getting the requests.The DELETE method allows the client todelete a web page on the server if the client has permission to do so. The CONNECT method was originally made as a reserve method; it may be used by proxy servers. | 3 | 6 |
| 7. | Message Transfer: After connection has been established between the SMTP client and server, a single message between a sender and one or more recipients can be exchanged. This phase involves eight steps. Steps 3 and 4 are repeated if there is more than one recipient.<br>1. The client sends the MAIL FROM message to introduce the sender of the message. It includes the mail address of the sender (mailbox and the domain name). This step is needed to give the server the return mail address for returning errors and reporting messages.<br>2. The server responds with code 250 or some other appropriate code.<br>3. The client sends the RCPT TO (recipient) message, which includes the mail address of the recipient.<br>4. The server responds with code 250 or some other appropriate code<br>5. The client sends the DATA message to initialize the message transfer.<br>6. The server responds with code 354 (start mail input) or some other appropriate message<br>7.The client sends the contents of the message in consecutive lines. Each line is terminated with two-character end-of-line token (carriage return and line feed). The message is terminated by a line containing just one period.<br>8. The server responds with code 250 (OK) or some other appropriate code | 6 | |

A wired LAN can be connected to another network or an internetwork such as the Inter-network

Figure 15.2 *Connection of a wired LAN and a wireless LAN to other networks*

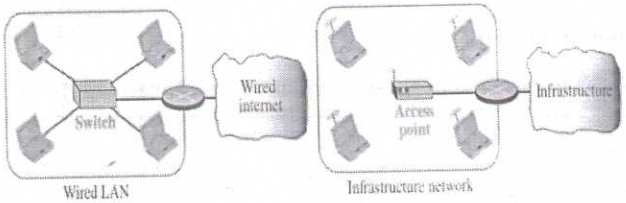
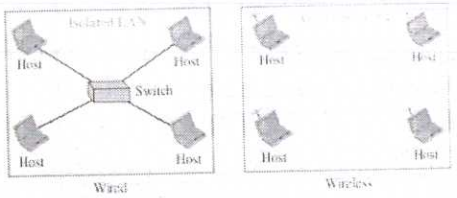



Figure 15.1 *Isolated LANs: wired versus wireless*




3

9

15

b Characteristics

6 6

1)Attenuation:Srength of signal decreases
2)Interference:Receive more than sender signal of same frequency
3)Multipath Propagation:electromagnetic waves can be reflected back
4)Error:Signal to noise ratio

V a Factors affecting network layer performance are:

***Delay***
***Throughput***
***Packet Loss***

***Delay :***

*four types.*

***Transmission Delay***
***Propagation Delay***
***Processing Delay***
***Queuing Delay***

9

b *OPEN LOOP CONGESTION:prevent congestion before it happens*

List of policies

**Retransmission Policy**
**Window Policy**
**Acknowledgment Policy**
**Discarding Policy**
**Admission Policy**

2+4

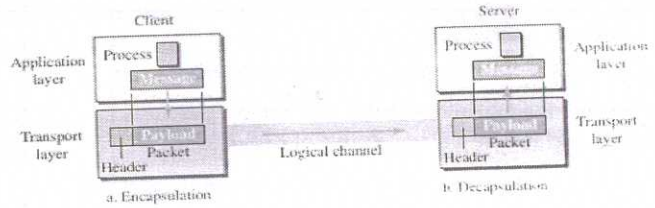
15

Encapsulation and Decapsulation

To send a message from one process to another, the transport layer protocol encapsulates and decapsulates messages. Encapsulation happens at the sender site. When a process has a message to send, it passes the message to the transport layer along with a pair of socket addresses and some other pieces of information, which depend on the transport-layer protocol. The transport layer receives the data and adds the transport-layer header. Decapsulation happens at the receiver site. When the message arrives at thedestinationtransport layer, the header is dropped and the transport layer delivers the messagee to the process running at the application layer. The sender socket address is passed to the process in case it needs to respond to the message received

2

Figure 23.7 *Encapsulation and decapsulation*




Multiplexing and Demultiplexing

Whenever an entity accepts items from more than one source, this is referred to as multiplexing (many to one); whenever an entity delivers items to more than one source, this is referred to as demultiplexing (one to many). The transport layer at the source performs multiplexing; the transport layer at the destination performs demultiplexing

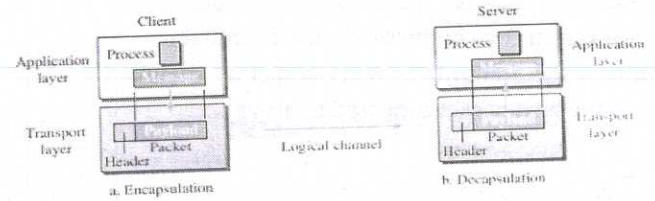
1

10

Flow Control

Whenever an entity produces items and another entity consumes them, there should be a balance between production and consumption rates.

Figure 23.7 *Encapsulation and decapsulation*




2

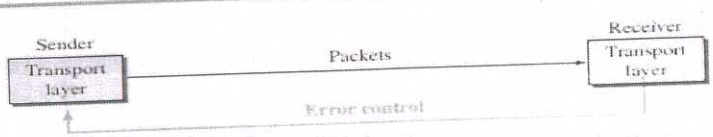
15

Error Control : It is responsible for

1. Detecting and discarding corrupted packets.
2. Keeping track of lost and discarded packets and resending them.
3. Recognizing duplicate packets and discarding them.
4. Buffering out-of-order packets until the missing packets arrive.

2

Figure 23.11 *Error control at the transport layer*



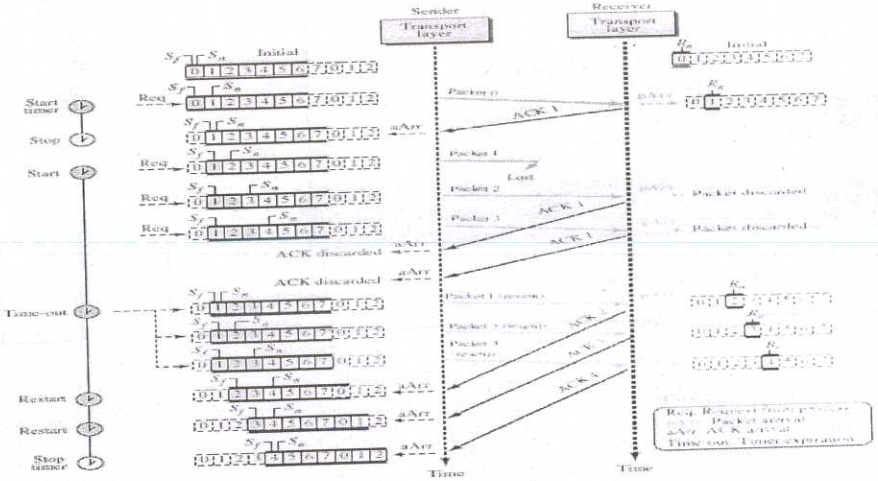
Send Window of Go Back - N :The send window at any time divides the possible sequence numbers into four regions. The first region, left of the window, defines the sequence numbers belonging to packets that are already acknowledged. The sender does not worry about these packets and keeps no copies of them. The second region, colored, defines the range
of sequence numbers belonging to the packets that have been sent, but have an unknown status. The sender needs to wait to find out if these packets have been received or were lost. We call these outstanding packets. The third range, white in the figure,defines the range of sequence numbers for packets that can be sent; however, the corresponding data have not yet been received from the application layer. Finally, the fourth region, right of the window, defines sequence numbers that cannot be used until the window slides. The maximum size of the window is 2m − 1

9

Receive window :In Go-Back-N, the size of the receive window isal ways 1. The receiver is always looking for the arrival of a specific packet. Any packet arriving out of order is discarded and needs to be resent.

Timers: Timers
Although there can be a timer for each packet that is sent, in our protocol we use onlyone. The reason is that the timer for the first outstanding packet always expires first. We resend all outstanding packets when this timer expires.
Resending packets When the timer expires, the sender resends all outstanding packets. For example, suppose
the sender has already sent packet 6 (Sn = 7), but the only timer expires. If Sf = 3,this means that packets 3, 4, 5, and 6 ave not been acknowledged; the sender goes back and resends packets 3, 4, 5, and 6. That is why the protocol is called Go-Back-N. On a time-out, the machine goes back N locations and resends all packets.

15

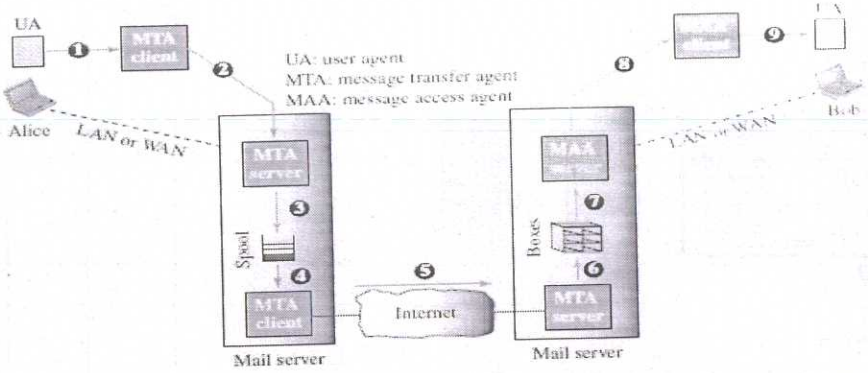


IX a In the common scenario, the sender and the receiver of the e-mail, Alice and Bob respectively, are connected via a LAN or a WAN to two mail servers. The administratorhas created one mailbox for each user where the received messages are stored. A mailbox is part of a server hard drive, a special file with permission restrictions. Only theowner of the mailbox has access to it. The administrator has also created a queue (spool) to store messages waiting to be sent. A simple e-mail from Alice to Bob takes nine different steps, as shown in the figure. Alice and Bob use three different agents: a user agent (UA), a message transfer agent (MTA), and a message access agent (MAA). When Alice needs to send a message to Bob, she runs a UA program to prepare the message and send it to her mail server. The mail server at her site uses a queue (spool) to store messages waiting to be sent. The message however, needs to be sent through the Internet from Alice's site to Bob's site sing an MTA. Here two message transfer agents are needed: one client and one server. Like most client-server programs on the Internet, the server needs to run all the time because it does not know when a client will ask for a connection. The client, on the other hand, can be triggered by the system when there is a message in the queue to be sent. The user agent at the Bob site allows Bob to read the received message. Bob later uses an MAA client to retrieve the message from an MAA server running on the second server. There are two important points we need to emphasize here. First, Bob cannot bypass the mail server and use the MTA server directly. To use the MTA server directly, Bob would need to run the MTA server all the time because he does not know when a message will arrive. This implies that Bob must keep his computer on all the time if he is connected to his system through a LAN. If he is connected through a WAN, he must keep the connection up all the time. Neither of these situations is feasible today. Second, note that Bob needs another pair of client-server programs: message access programs. This is because an MTA client-server program is a push program: the client pushes the message to the server. Bob needs a pull program. The client needs to pull the message from the server.

6

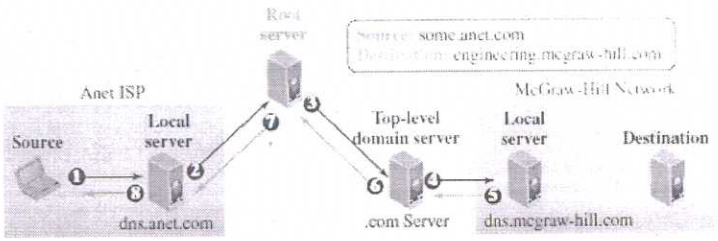
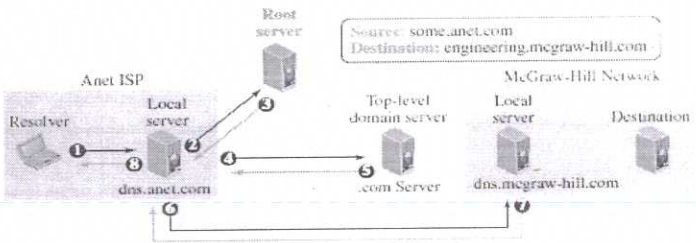
9

Figure 26.12 *Common scenario*




15

4

| | | | | |
|---|---|---|---|---|
| b | Mapping a name to an address is called name-address resolution. DNS is designed as a client-server application. A host that needs to map an address to a name or a name to an address calls a DNS client called a resolver. The resolver accesses the closest DNS server with a mapping request. If the server has the information, it satisfies 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 interprets the response to see if it is a real resolution or an error, and finally delivers the result to the process that requested it. A resolution can be either recursive or iterative. | 3 | 7 | 15 |
| | Figure 26.36 *Recursive resolution*<br>  <br><br>Figure 26.37 *Iterative resolution*<br>  <br> | 2 + 2 | | |