IAT3_solution

Subject:Computer Communication Networks(15EC64)

Faculty:-Dr.Sharmila K P/Jyoti M R

1a)Explain with neat diagram TCP segment format.

The segment consists of a header of 20 to 60 bytes, followed by data from the application program. The header is 20 bytes if there are no options and up to 60 bytes if it contains options.

Source port address. This is a 16-bit field that defines the port number of the application program in the host that is sending the segment.

Destination port address. This is a 16-bit field that defines the port number of the application program in the host that is receiving the segment.

Sequence number. This 32-bit field defines the number assigned to the first byte of data contained in this segment. As we said before, TCP is a stream transport protocol. To ensure connectivity, each byte to be transmitted is numbered. The sequence number tells the destination which byte in this sequence is the first byte in the segment. During connection establishment (discussed later) each party uses a random number generator to create an **initial sequence number** (ISN), which is usually different in each direction.

Acknowledgment number. This 32-bit field defines the byte number that the receiver of the segment is expecting to receive from the other party. If the receiver of the segment has successfully received byte number *x* from the other party, it returns *x* □□1 as the acknowledgment number. Acknowledgment and data can be piggybacked together.

Header length. This 4-bit field indicates the number of 4-byte words in the TCP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field is always between 5 (5 □□4 □□20) and 15 (15 □□4 □□60).

Control. This field defines 6 different control bits or flags, as shown in Figure 24.8. One or more of these bits can be set at a time. These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP. A brief description of each bit is shown in the figure. We will discuss them further when we study the detailed operation of TCP later in the chapter.

Window size. This field defines the window size of the sending TCP in bytes. Note that the length of this field is 16 bits, which means that the maximum size of the window is 65,535 bytes. This value is normally referred to as the receiving window (*rwnd*) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.

Checksum. This 16-bit field contains the checksum. The calculation of the checksum for TCP follows the same procedure as the one described for UDP. However, the use of the checksum in the UDP datagram is optional, whereas the use of the checksum for TCP is mandatory.

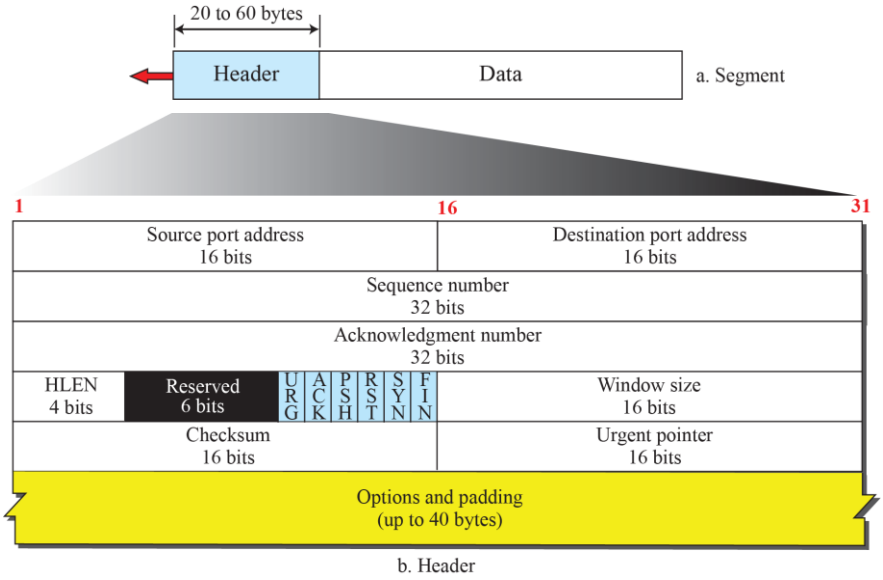
Urgent pointer. This 16-bit field, which is valid only if the urgent flag is set, is used when the segment contains urgent data. It defines a value that must be added

to the sequence number to obtain the number of the last urgent byte in the data section of the segment. This will be discussed later in this chapter.

Options. There can be up to 40 bytes of optional information in the TCP header. We will discuss some of the options used in the TCP header later in the section.

Yhe TCP segment is shown in the below figure

TCP segment format



b)Explain TCP sending and receiving buffers with neat diagrams.

Sending and reciving buffers in TCP

- At the transport layer writing and reading not happen at same time so TCP needs buffers for storage.
- There are two buffers, the sending buffer and the receiving buffer, one for each direction.
- . we have shown two buffers of 20 bytes each; normally the buffers are hundreds or thousands of bytes, depending on the implementation.
- The figure shows the movement of the data in one direction. At the sender, the buffer

has three types of chambers. The white section contains empty chambers that can be

filled by the sending process (producer). The colored area holds bytes that have been

sent but not yet acknowledged. The TCP sender keeps these bytes in the buffer until it

receives an acknowledgment. The shaded area contains bytes to be sent by the sending
TCP.

- TCP may be able to send only part of this shaded section. This could be due to the slowness of the receiving process or to congestion in the network. Also note that, after the bytes in the colored chambers are acknowledged, the chambers are recycled and available for use by the sending process.

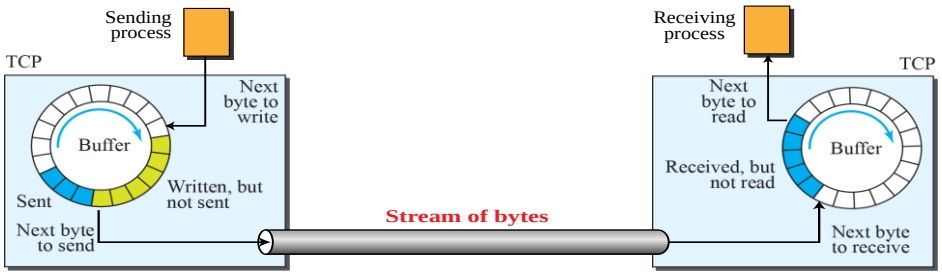
.

- The operation of the buffer at the receiver is simpler. The circular buffer is divided

into two areas (shown as white and colored). The white area contains empty chambers
to be filled by bytes received from the network.

- The colored sections contain received bytes that can be read by the receiving process. When a byte is read by the receiving process, the chamber is recycled and added to the pool of empty chambers.

Sending and receiving buffers



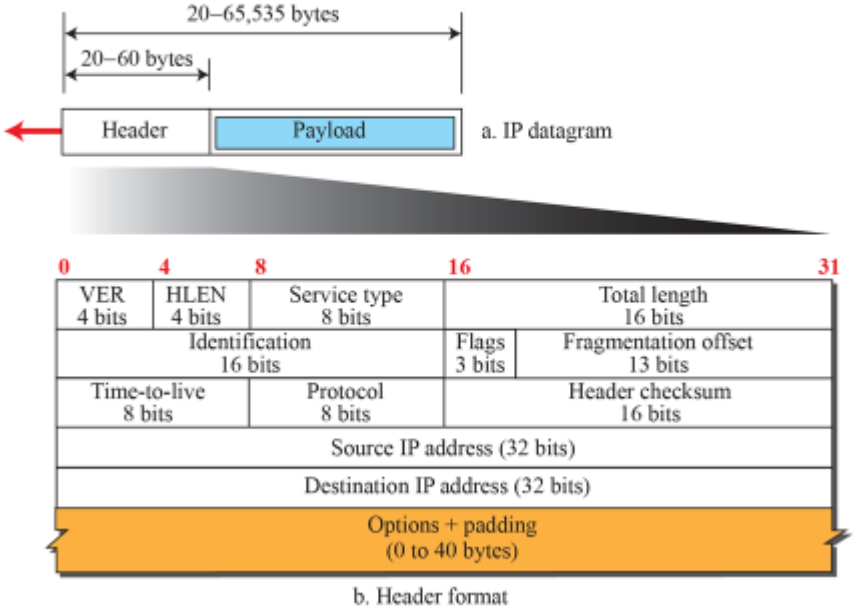
TCP needs buffers for storage.

2a)Explain IPv4 datagram format with neat diagram
b) In an IPv4 packet HLEN is$(1000)_2$.How many bytes of options are being carried by this packet?

- Ans: *Version (VER). This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4.*
- *Header length (HLEN). This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes).*
- *Services. This field, previously called service type, is now called differentiated services.*

- *The total length field defines the total length of the datagram including the header.*
- *Identification,Flags, Fragmentation offset.- used in fragmentation*
- *Time to live. A datagram has a limited lifetime in its travel through an internet.*
- *This field was originally designed to hold a timestamp, which was decremented by each visited router.*
- *The datagram was discarded when the value became zero*
- *Protocol. This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer. An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IPv4 datagram is delivered*
- *Checksum. The 16 bit checksum is used for error correction.*
- *Source address. This 32-bit field defines the IPv4 address of the source. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.*
- *Destination address. This 32-bit field defines the IPv4 address of the destination. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.The IP datagram format is shown in below figure,*

IP datagram



b)1000-→8*4=32
32-20bytes header
22bytes option

3) With neat diagram explain internal and external Border Gateway Protocol(BGP).

- The **Border Gateway Protocol version 4 (BGP4)** is the only interdomain routing protocol used in the Internet today.
- BGP4 is based on the path-vector algorithm.
- BGP4, is a complex protocol. In this section, Figure 3.1 shows an example of an internet with four autonomous systems. AS2, AS3, and AS4 are *stub* autonomous systems;
- AS1 is a *transient* one. In our example,data exchange between AS2, AS3, and AS4 should pass through AS1.
- To enable each router to route a packet to any network in the internet, we first
  install a variation of BGP4, called *external BGP* (*eBGP*), on each *border router* (the
- one at the edge of each AS which is connected to a router at another AS).
- We then install the second variation of BGP, called *internal BGP* (*iBGP*), on all routers. This means that the border routers will be running three routing protocols (intradomain,
  eBGP, and iBGP), but other routers are running two protocols (intradomain and iBGP).
- We can say that BGP is a kind of point-to-point protocol. When the software is installed
  on two routers, they try to create a TCP connection.
- In other words, a pair of client and server processes continuously communicate with each
  other to exchange messages. The two routers that run the BGP processes are called
  BGP peers or *BGP speakers*.
- The eBGP variation of BGP allows two physically connected border routers in two
  different ASs to form pairs of eBGP speakers and exchange messages.
- The routers thatare eligible in our example in Figure 3.2 form three pairs: R1-R5, R2-R6, and R4-R9. The connection between these pairs is established over three physical WANs (N5,N6, and N7).
- However, there is a need for a logical TCP connection to be created over the physical connection to make the exchange of information possible.

Fig 3.1

A sample internet with four ASs

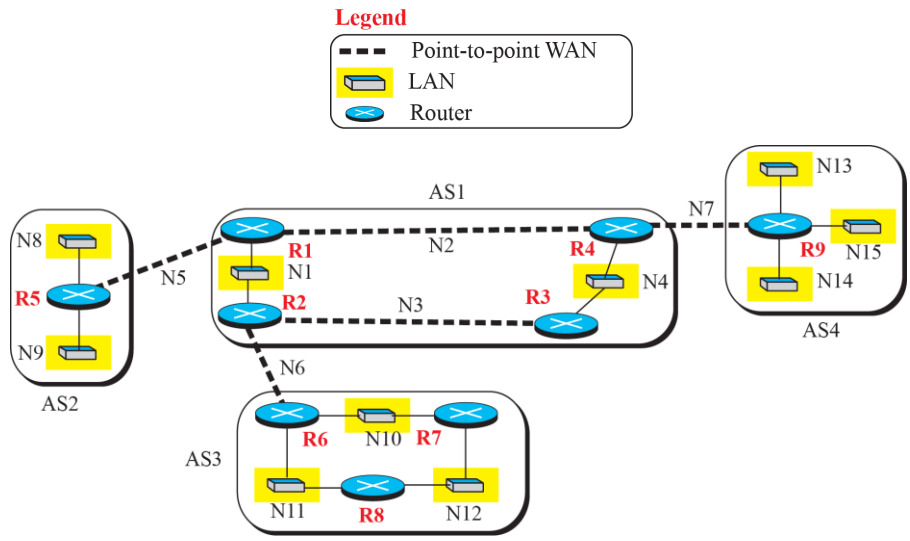


Fig 3.2

eBGP operation

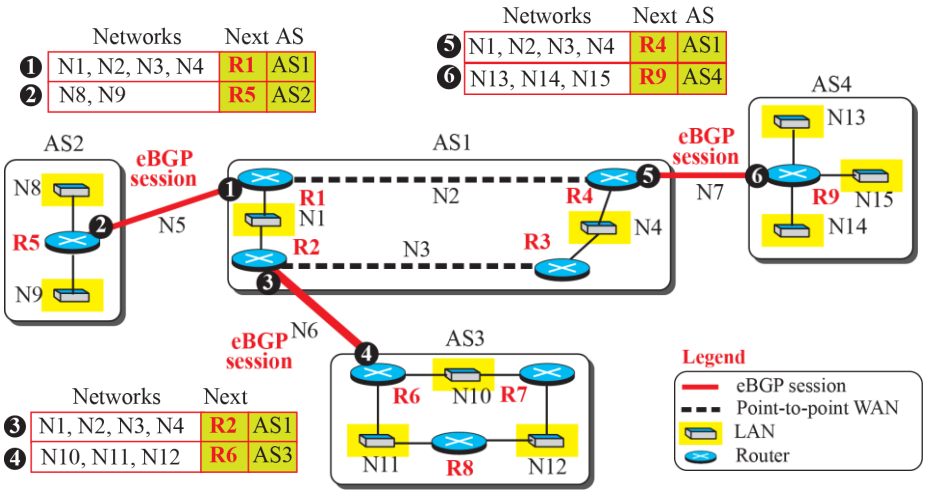


Fig 3.3

4)Explain Goback N protocol with send ,receive window and flow diagram.

The Go-Back-N protocol simplifies the process at the receiver. The receiver keeps track of only one variable, and there is no need to buffer out-of-order packets; they are simply discarded.

However, this protocol is inefficient if the underlying network protocol loses a lot of packets. Each time a single packet is lost or corrupted, the sender resends all outstanding packets, even though some of these packets may have been received safe and

sound but out of order.Ack.no. is the seq. no. of the error free packet received. To overcome the problem of go back N protocol selective repeat protocol is used.

To improve the efficiency of transmission (to fill the pipe), multiple packets must be in transition while the sender is waiting for acknowledgment.

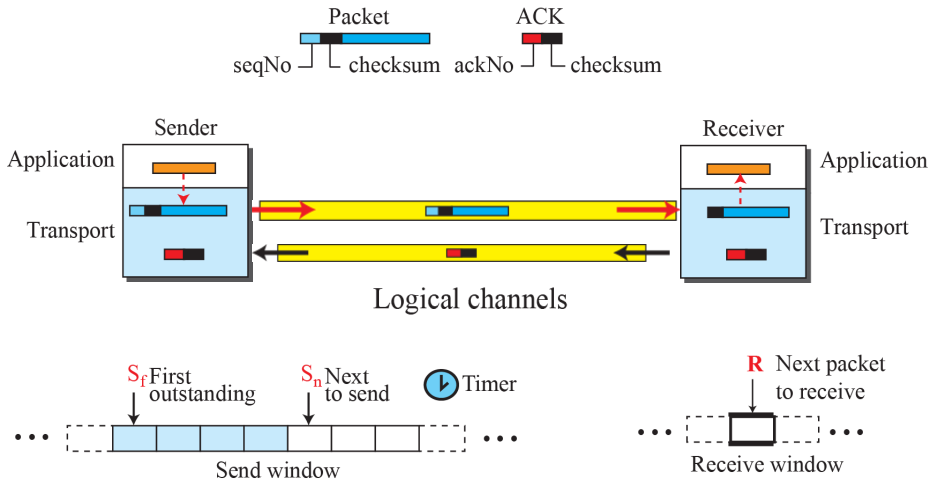
In other words, we need to let more than one packet be outstanding to keep the channel busy while the sender is waiting for acknowledgment.

Ack.no. defines the seq.no. of the next packet expected

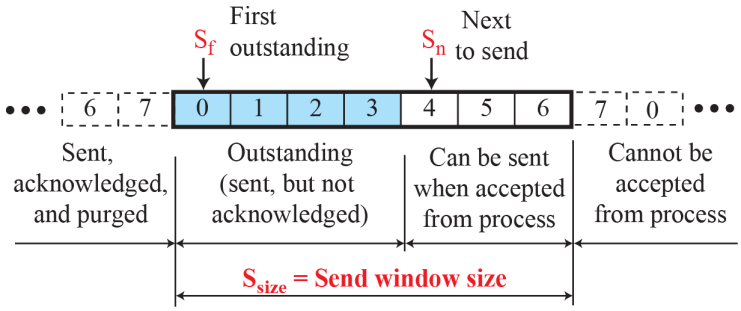
the sequence numbers are modulo 2^m, where m is the size of the sequence number field in bits.

Send window is an imaginary box of max. size= 2^m - 1
Receive window is an imaginary box of size 1.

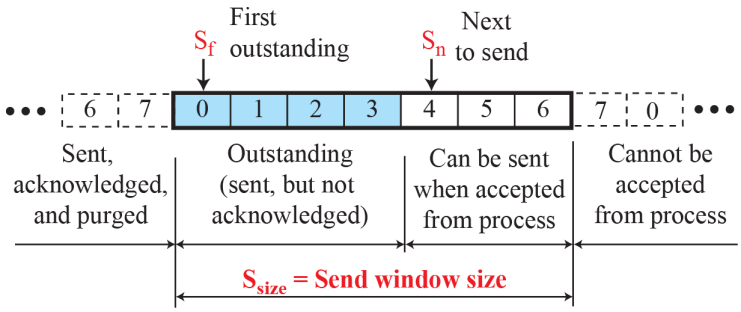
Go-Back-N protocol



Send window for Go-Back-N



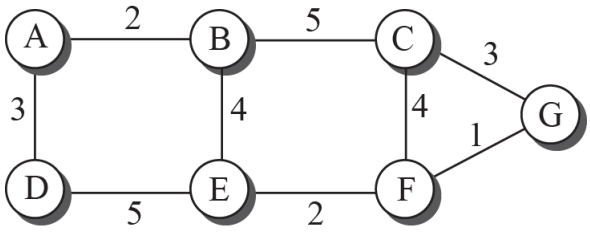
Send window for Go-Back-N



5. Explain Link state Routing and also explain Dijkstra algorithm to design least cost tree.

A routing algorithm that directly follows our discussion for creating least-cost trees and forwarding tables is link-state (LS) routing. This method uses the term link-state to define the characteristic of a link (an edge) that represents a network in the internet. In this algorithm the cost associated with an edge defines the state of the link. Links with lower costs are preferred to links with higher costs; if the cost of a link is infinity, it means that the link does not exist or has been broken.

Example of a link-state database



a. The weighted graph

	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

b. Link state database

To create a least-cost tree for itself, using the shared LSDB, each node needs to run the famous Dijkstra Algorithm. This iterative algorithm uses the following steps:

1. The node chooses itself as the root of the tree, creating a tree with a single node, and sets the total cost of each node based on the information in the LSDB.
2. The node selects one node, among all nodes not in the tree, which is closest to the root, and adds this to the tree. After this node is added to the tree, the cost of all other nodes not in the tree needs to be updated because the paths may have been changed.
3. The node repeats step 2 until all nodes are added to the tree. We need to convince ourselves that the above three steps.

6)Explain TCP connection establishment and connection termination with neat flow diagrams

- TCP transmits data in full-duplex mode. When two TCPs in two machines are connected,
  they are able to send segments to each other simultaneously.
- Before data transfer connection will be established between sender and receiver.
- TCP uses three phases for data transfer
  1. Connection establishment
  2. Data transfer
  3. Connection termination

Connection establishment

The process starts with the server. The server program tells its TCP that it is ready to accept a connection. This request is called a passive open.
Although the server TCP is ready to accept a connection from any machine in the world, it cannot make the connection itself.
The client program issues a request for an active open. A client that wishes to connect to an open server tells its TCP to connect to a particular server.
TCP can now start the three-way handshaking process, as shown in below Figure7.1
The three steps in this phase are as follows.

1)

- The client sends the first segment, a SYN segment, in which only the SYN flag is set. This segment is for synchronization of sequence numbers.
- The client in our example chooses a random number as the first sequence number and sends this number to the server. This sequence number is called the initial sequence number (ISN). Note that this segment does not contain an acknowledgment number.
- It does not define the window size either; a window size definition makes sense only when a segment includes an acknowledgment.
- SYN segment is a control segment and carries no data. but, it consumes one sequence number because it needs to be acknowledged.

2.

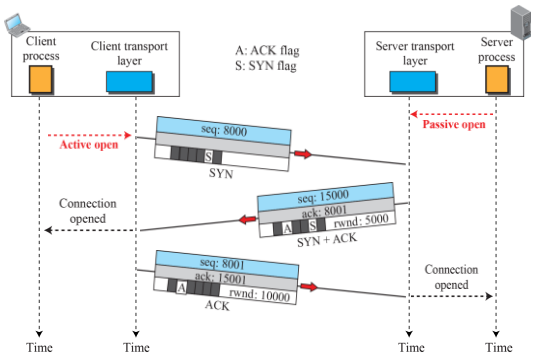
- The server sends the second segment, a SYN + ACK segment with two flag bits set as: SYN and ACK.
- This segment has a dual purpose. First, it is a SYN segment for communication in the other direction.
- The server uses this segment to initialize a sequence number for numbering the bytes sent from the server to the client.
- The server also acknowledges the receipt of the SYN segment from the client by setting the ACK flag and displaying the next sequence number it expects to receive from the client.
- Because the segment contains an acknowledgment, it also needs to define the receive window size, rwnd (to be used by the client), It consumes one sequence number.

3)

- The server sends the second segment, a SYN + ACK segment with two flag bits set as: SYN and ACK. T
- his segment has a dual purpose. First, it is a SYN segment for communication in the other direction. The server uses this segment to initialize a sequence number for numbering the bytes sent from the server to the client.
- The server also acknowledges the receipt of the SYN segment from the client by setting the ACK flag and displaying the next sequence number it expects to receive from the client.
- Because the segment contains an acknowledgment, it also needs to define the receive window size, rwnd (to be used by the client), It, therefore, consumes one sequence number.

Fig 7.1

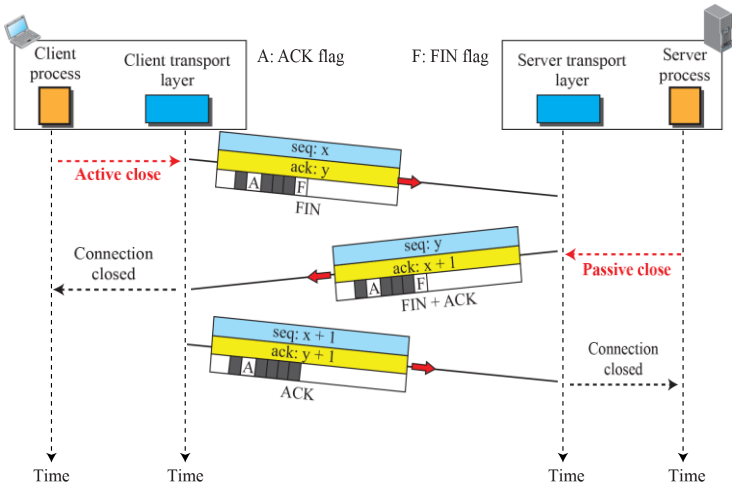
Connection establishment using three-way handshaking



Connection termination

Fig 7.3

Connection termination using three-way handshaking



Either of the two parties involved in exchanging data (client or server) can close the
connection, although it is usually initiated by the client.

Three-Way Handshaking

Most implementations today allow *three-way handshaking* for connection termination,

as shown in Figure 7.3

- In this situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set.
- FIN segment can include the last chunk of data sent by the client or it can be

just a control segment. If it is only a control segment, it consumes only one sequence number because it needs to be acknowledged

- The server TCP, after receiving the FIN segment, informs its process

and sends the second segment, a FIN □□ACK segment.

- FIN segment from the client announce the closing of the connection

in the other direction.

- This segment can also contain the last chunk of data from the server. If it does not carry data, it consumes only one sequence number because it needs to be acknowledged.
- The client TCP sends the last segment, an ACK segment, to confirm the receipt of

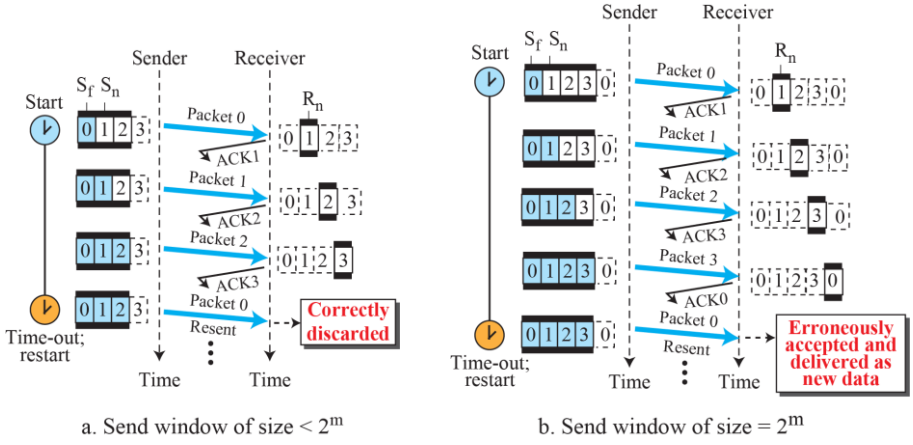
the FIN segment from the TCP server. This segment contains the acknowledgment

number, which is one plus the sequence number received in the FIN segment from

the server. This segment cannot carry data and consumes no sequence numbers.

7)Explain with neat flow diagrams why send and receive window size of selective repeat protocl must be equal to $2^m/2$.

Send window size for Go-Back-N



a. Send window of size < 2^m

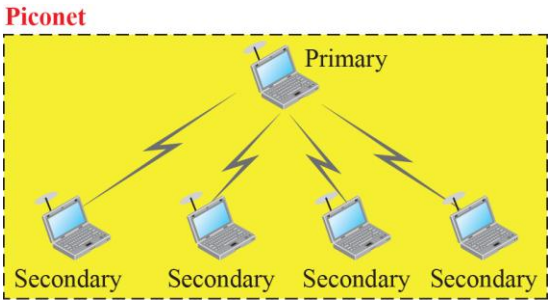
b. Send window of size = 2^m

- Send Window Size We can now show why the size of the send window must be less than 2^m.
- As an example, we choose m = 2, which means the size of the window can be $2^m - 1$, or 3.above Figure compares a window size of 3 against a window size of 4.
- If the size of the window is 3 (less than 2m) and all three acknowledgments are lost, the only timer expires and all three packets are resent.
- The receiver is now expecting packet 3, not packet 0, so the duplicate packet is correctly discarded. On the other hand, if the size of the window is 4 (equal to 2^2) and all acknowledgments are lost, the sender will send a duplicate of packet 0.
- However, this time the window of the receiver expects to receive packet 0 (in the next cycle), so it accepts packet 0, not as a duplicate, but as the first packet in the next cycle.
- This is an error. This shows that the size of the send window must be less than 2^m.

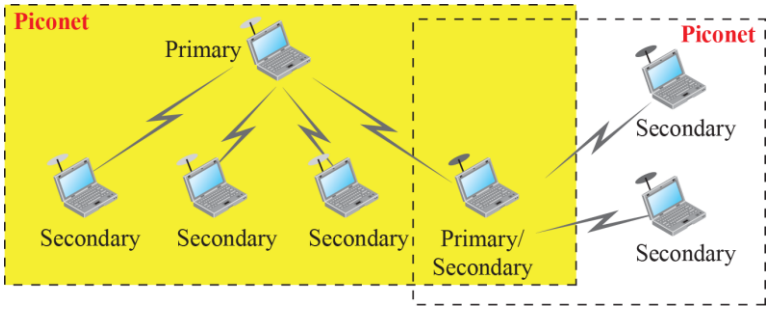
8)Explain Bluetooth different networks also explain Bluetooth layers.

Bluetooth defines two types of networks: piconet and scatternet.

Piconet



Scatternet



- A Bluetooth network is called a ***piconet,*** or a small net. A piconet can have up to eight

stations, one of which is called the *primary;* the rest are called *secondaries.*

- All the secondary stations synchronize their clocks and hopping sequence with the primary. Note

that a piconet can have only one primary station. The communication between the primary and secondary stations can be one-to-one or one-to-many.

- Although a piconet can have a maximum of seven secondaries, additional secondaries

can be in the *parked state*.

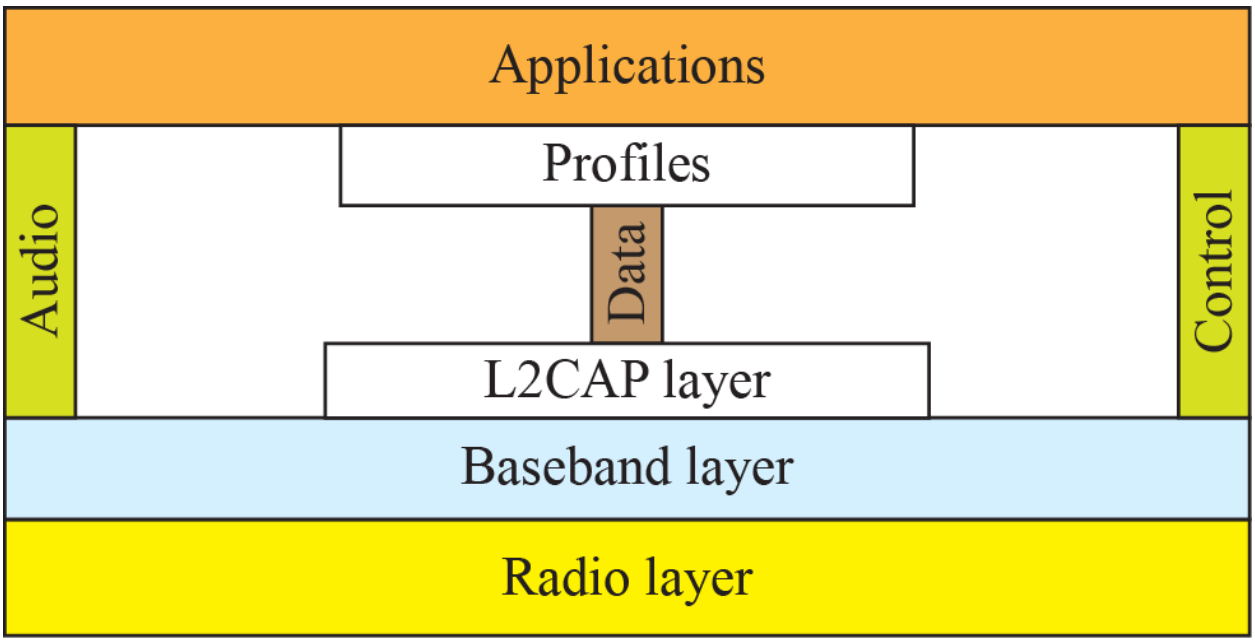
- A secondary in a parked state is synchronized with the primary, but cannot take part in communication until it is moved from the parked state to the active state. Because only eight stations can be active in a piconet, activating a station from the parked state means that an active station must go to the parked state.

Piconet

- Piconets can be combined to form what is called a ***scatternet.*** A secondary station in

one piconet can be the primary in another piconet.

- This station can receive messages from the primary 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.



L2CAP

The **Logical Link Control and Adaptation Protocol,** or **L2CAP** (L2 here means LL),

is roughly equivalent to the LLC sublayer in LANs. It is used for data exchange on an

The 16-bit length field defines the size of the data, in bytes, coming from the upper

layers. Data can be up to 65,535 bytes. The channel ID (CID) defines a unique identifier

for the virtual channel created at this level (see below).

The L2CAP has specific duties: multiplexing, segmentation and reassembly, quality
of service (QoS), and group management.

Multiplexing

The L2CAP can do multiplexing. At the sender site, it accepts data from one of the
upper-layer protocols, frames them, and delivers them to the baseband layer. At the
receiver site, it accepts a frame from the baseband layer, extracts the data, and delivers
them to the appropriate protocol layer. It creates a kind of virtual channel that we will
discuss in later chapters on higher-level protocols.

Segmentation and Reassembly

The maximum size of the payload field in the baseband layer is 2774 bits, or 343 bytes.
This includes 4 bytes to define the packet and packet length. Therefore, the size of the
packet that can arrive from an upper layer can only be 339 bytes. However, application
layers sometimes need to send a data packet that can be up to 65,535 bytes (an Internet
packet, for example). The L2CAP divides these large packets into segments and adds
extra information to define the location of the segments in the original packet. The
L2CAP segments the packets at the source and reassembles them at the destination.

QoS

Bluetooth allows the stations to define a quality-of-service level. We discuss quality of
service in Chapter 30. For the moment, it is sufficient to know that if no quality-of-service
level is defined, Bluetooth defaults to what is called *best-effort* service; it will do its
best under the circumstances.

Group Management

Another functionality of L2CAP is to allow devices to create a type of logical addressing between themselves. This is similar to multicasting. For example, two or three secondary devices can be part of a multicast group to receive data from the primary.

Baseband Layer

The baseband layer is roughly equivalent to the MAC sublayer in LANs. The access
method is TDMA . The primary and secondary stations communicate
with each other using time slots. The length of a time slot is exactly the same as the
dwell time, 625micro seconds.

This means that during the time that one frequency is used, a primary
sends a frame to a secondary, or a secondary sends a frame to the primary. Note that the communication is only between the primary and a secondary; secondaries cannot
communicate directly with one another.