CBCS SCHEME

USN | | | | | | | | | |

20MCA13

First Semester MCA Degree Examination, Feb./Mar. 2022
Computer Networks

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

Module-1

1 a. What are the perspective requirements of Computer Network? Explain. **(06 Marks)**
b. Briefly explain various requirements of Computer Network. **(10 Marks)**
c. Define Link, Nodes, Switches and Raster. **(04 Marks)**

OR

2 a. With a neat diagram, explain Internal Architecture of Computer Network. **(10 Marks)**
b. What is socket programming? Explain various methods used in server side. **(06 Marks)**
c. Discuss Bandwidth and Latency. **(04 Marks)**

Module-2

3 a. With neat diagram of frame format explain BISYNC and HDLC framing. **(10 Marks)**
b. Explain steps of Internet Check Sum and suppose that the sender sends the following four frames of eight bits. Check whether the frames is accepted or not using internet check sum.
11001100 10101010 11110000 11000011 **(10 Marks)**

OR

4 a. Explain ethernet frame format and transmission algorithm. **(10 Marks)**
b. Explain the following :
(i) Stop and wait protocol
(ii) 802.11 wifi **(10 Marks)**

Module-3

5 a. What is a Datagram network? Explain its characteristics. **(10 Marks)**
b. In detail, explain IPv4 packet header format **(10 Marks)**

OR

6 a. Explain class A, class B, class C of IP Addresses. **(10 Marks)**
b. With neat diagram, explain (i) ARP (ii) DHCP. **(10 Marks)**

Module-4

7 a. Explain 3-way handshaking in TCP. **(08 Marks)**
b. Explain simple Demultiplexer and its header format (UDP). **(06 Marks)**
c. Give the difference between UDP and TCP. **(06 Marks)**

OR

8 a. What is congestion? Explain Leaky Bucket algorithm. **(10 Marks)**
b. Explain the following :
(i) Queuing Discipline
(ii) TCP Congestion Control **(10 Marks)**

Module-5

9 a. Define Cipher. Explain Symmetric key cipher. **(08 Marks)**

b. Write a note on :
(i) SMTP
(ii) DNS **(08 Marks)**

c. What is a firewall? Explain its strength and weakness of a firewall. **(04 Marks)**

OR

10 a. Explain how public key authentication work. **(10 Marks)**

b. What are the security threats in Internet working? **(06 Marks)**

c. Write a note on www. **(04 Marks)**

* * * * *

MODULE-1

1. a. What are the perspective requirements of Computer Network? Explain?

   In perspective requirements we focused on the perspective of someone who would design networking equipment and protocols. We continue to focus on this perspective. We also want to cover two additional groups that are of increasing importance: those who develop networked applications and those who manage or operate networks. Let's consider how these three groups might list their requirements for a network:

   i. An application programmer would list the services that his or her application needs—for example, a guarantee that each message the application sends will be delivered without error within a certain amount of time or the ability to switch gracefully among different connections to the network as the user moves around.

   ii. A network operator would list the characteristics of a system that is easy to administer and manage—for example, in which faults can be easily isolated, new devices can be added to the network and configured correctly, and it is easy to account for usage.

   iii. A network designer would list the properties of a cost-effective design—for example, that network resources are efficiently utilized and fairly allocated to different users. Issues of performance are also likely to be important. This section attempts to distill these different perspectives into a high-level introduction to the major considerations that drive network design.

   b. Briefly explain various requirements of Computer Network.

a. **Perspectives.**
b. **Scalable Connectivity.**
c. **Cost – Effective Resource Sharing.**
d. **Support for Common Services.**
e. **Manageability**

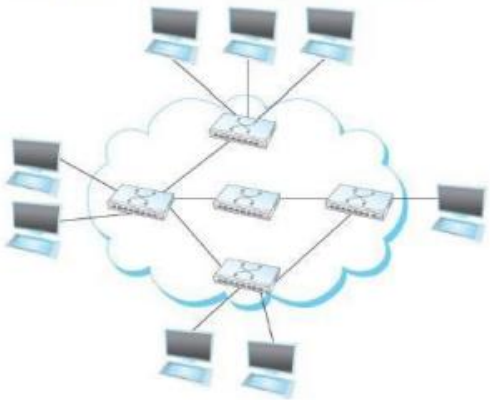
Perspectives

- An *application programmer* list the services based on application needs. For example, a guarantee that each message will be delivered without error or within a certain time or to allow graceful switching in a mobile environment.
- A *network operator* lists the characteristics of a system that is easy to administer and manage. For example, fault isolation, adding new devices, easy to account for usage, etc.
- A *network designer* lists the properties of a cost-effective design. For example, efficient utilization of network resources, fair allocation to users, etc.

Scalable Connectivity

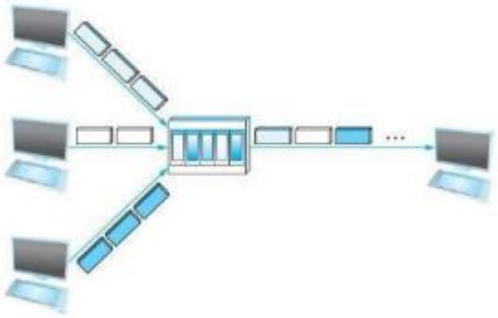
- A system that is designed to support growth to an arbitrarily large size is *scalable*.
- Physical medium is referred to as *link*, and devices that connect to the link are *nodes*.
- Link could be either *dedicated* point-to-point between nodes or *shared* amongst nodes with multiple access.
- End nodes can be connected through a set of forwarding nodes called *switches*. Switching could be either circuit or packet switching.
- Packet switching networks uses *store-and-forward* method, i.e. the switch receives a packet, stores in its buffer and later forwards onto another link.

- Independent networks are connected to form *internetwork* or internet. A node that connects two or more networks is known as *router*.
- The process of forwarding frames from source to destination is known as *routing*.
- A node can also send messages to a group of nodes (*multicasting*) or to all nodes on the network (*broadcasting*).
- Each node on the network is assigned a unique address.



- Switch decides which packet is to be transmitted from the packets queued up, according to queuing discipline such as FIFO.

- Switch decides which packet is to be transmitted from the packets queued up, according to queuing discipline such as FIFO.



Support for Common Services

- Since applications have common services in, it is apt for the network designer to identify and implement a common set of services for the application designer to build upon.
- Network provide logical channels and set of services required for process-to-process communication.
- Functionalities may include guaranteed delivery, in-order delivery, privacy, etc.
- File access program such as FTP / NFS or sophisticated digital library application require read and write operation performed either by client / server.
- Two types of communication channels that could be provided are *request/reply* and *message stream* channel
- Request/reply channel guarantees delivery of message and ensures privacy and integrity of data required in case of FTP or digital library.
- Message stream channel does not guarantee delivery of all data but assures in-order delivery, required in applications like video conferencing.

Reliability

- Reliability is an important characteristic to be provided by the network, i.e., it should be possible for the network to recover from errors.
- Single bit/ burst errors may occur during data transmission due to interference. Such errors can be detected and retransmission sought for.
- Packets can be dropped due to congestion or wrongly routed.

- Packets can be dropped due to congestion or wrongly routed.
- Links can fail or node can crash. In case of failed link, it should be possible to route the packet along alternate path.

Cost-Effective Resource Sharing

- Hosts can share network resources using the concept of *multiplexing*. For example, multiple flows can be multiplexed onto a single physical link.
- Synchronous time-division multiplexing (*STDM*) divides time into equal slots and flows use the slots in a round-robin manner.
- Frequency-division multiplexing (*FDM*) transmits each flow at different frequency.
- In statistical multiplexing, link is shared over time as in STDM, but packets are transmitted from each flow on demand, rather than on predetermined slot.
- Packets multiplexed at one end, is demultiplexed at the other switch.

Manageability

- Network needs troubleshooting to adapt to increase in traffic or to improve performance.
- Managing network devices on the internet to work correctly is a challenging one.
- Automating network management tasks is needed for scalability and cost-effectiveness.
- Network nowadays is common and could be managed by consumers with little skill level.

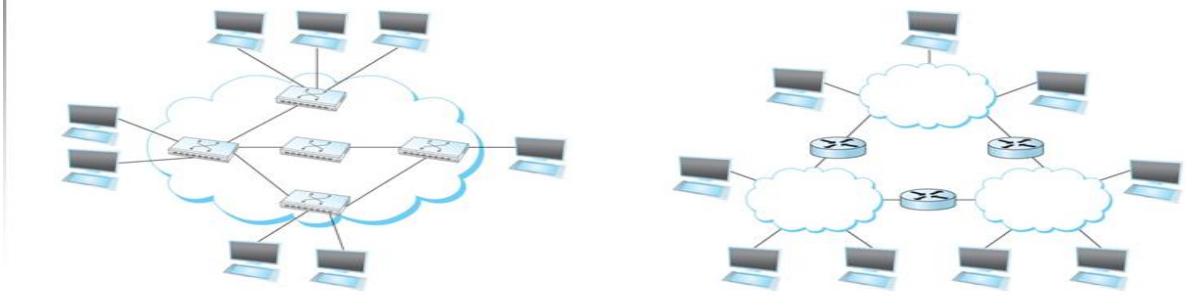
c. Define Link, Nodes, Switches and Raster.

Link: Physical medium that connects nodes

Nodes: a device (could be a computer, switch, etc) on the network

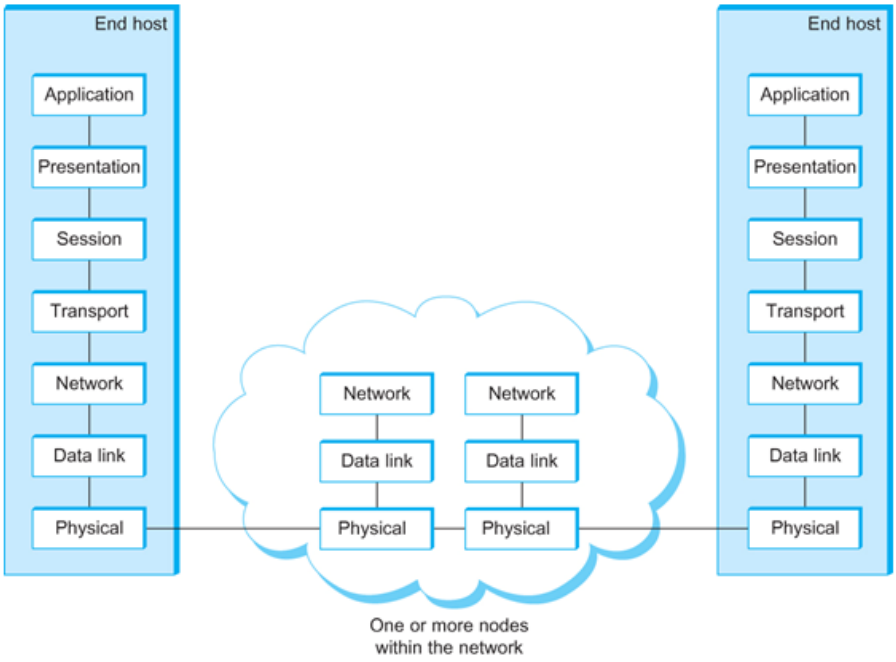
Switch: nodes inside the cloud(network) and implement the network (store and forward packets)

Router: connects two or more networks (plays much the same role as a switch—stores and forwards)



2. a. With a neat diagram explain Internal Architecture of Computer Network.

OSI Architecture



The OSI 7-layer Model

- Physical Layer
  - Handles the transmission of raw bits over a communication link
- Data Link Layer
  - Collects a stream of bits into a larger aggregate called a *frame*
  - Network adaptor along with device driver in OS implement the protocol in this layer
  - Frames are actually delivered to hosts
- Network Layer
  - Handles routing among nodes within a packet-switched network
  - Provides Host-to-Host connectivity
  - Unit of data exchanged between nodes in this layer is called a *packet*

  The lower three layers are implemented on all network nodes
- Transport Layer
  - Implements a process-to-process channel
  - Unit of data exchanges in this layer is called a message
- There is a disagreement on the top three
  - Session, Presentation, and Application
  - Mainly because the are not always present

- Session Layer
  - Provides a name space that is used to tie together the potentially different transport streams that are part of a single application
  - Ex., tie the audio and video together in videoconference
- Presentation Layer
  - Concerned about the format of data exchanged between peers
  - Ex., integer formats, audio/video format, most/least significant
- Application Layer
  - Include things like the Hypertext Transfer Protocol
    - Basis the world wide web
    - Used by web browsers
- The transport layer and the higher layers typically run only on end-hosts and not on the intermediate switches and routers

b. What is socket programming? Explain various methods used in server side.

- What is a socket?
  - The point where a local application process attaches to the network
  - An interface between an application and the network
  - An application creates the socket
- The interface defines operations for
  - Creating a socket
  - Attaching a socket to the network
  - Sending and receiving messages through the socket
  - Closing the socket

int sockfd = socket(protocol_family, type, protocol);

- Protocol Family
  - PF_INET denotes the Internet family
  - PF_PACKET denotes direct access to the network interface (i.e., it bypasses the TCP/IP protocol stack)
- The socket number returned is the socket descriptor for the newly created socket

int sockfd = socket(protocol_family, type, protocol);

- Socket Type
  - SOCK_STREAM is used to denote a byte stream (TCP)
  - SOCK_DGRAM is an alternative that denotes a message oriented service, such as that provided by UDP
- int sockfd = socket (PF_INET, SOCK_STREAM, 0);
- int sockfd = socket (PF_INET, SOCK_DGRAM, 0);

  The combination of PF_INET and SOCK_STREAM implies TCP

Server

- Passive open
- Prepares to accept connection, does not actually establish a connection

Server invokes

```
int bind (int socket, struct sockaddr *address,
                                        int addr_len)
int listen (int socket, int backlog)
int accept (int socket, struct sockaddr *address,
                                    int *addr_len)
```

Bind

- Binds the newly created socket to the specified address i.e. the network address of the local participant (the server)
- Address is a data structure which combines IP and port

Listen

- Defines how many connections can be pending on the specified socket

Accept

- Carries out the passive open
- Blocking operation
  - Does not return until a remote participant has established a connection
  - When it does, it returns a new socket that corresponds to the new established connection and the address argument contains the remote participant's address

- Once a connection is established, the application process invokes two operation
-
```
int send (int socket, char *msg, int msg_len,
                    int flags)
int recv (int socket, char *buff, int buff_len,
                     int flags)
```

d. Discuss bandwidth and Latency.

<u>Bandwidth of a Network</u>

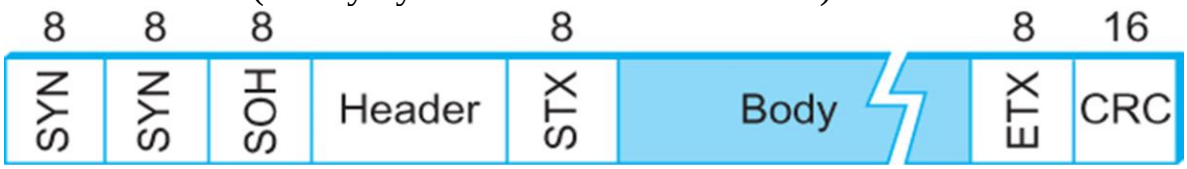
- Number of bits per second that can be transmitted over a communication link
- Throughput vs. bandwidth (from the most confusing terms in computer networks. Bandwidth: the **maximum** data rate (bits per second)
  - Throughput: number of bits per second that we actually transmit over the link **in practice**
- 1 Mbps: 1×10^6 bits/second = $1x2^{20}$ bits/sec

- 1×10^{-6} seconds to transmit each bit or imagine that a timeline, now each bit occupies 1 micro second space.
- On a 2 Mbps link the width is 0.5 micro second.

- Latency: How long it takes a message to travel from one end of a network to the other.
  - Measured in time
- Latency = Propagation + transmit + queue

Module-2

3. a. With neat diagram of frame format explain BISYNC and HDLC framing.
   - BISYNC (Binary Synchronous Communication) Protocol



BISYNC Frame Format

- BISYNC – Byte Oriented
  - Beginning of a frame is denoted by sending a special SYN (synchronize) character
  - Data portion of the frame is contained between special sentinel character STX (start of text) and ETX (end of text)
  - SOH : Start of Header
  - DLE : Data Link Escape (character stuffing-Extra characters are inserted in the data portion of the frame)
  - CRC: Cyclic Redundancy Check to detect transmission errors.

HDLC –Bit oriented

- **HDLC : High Level Data Link Control**
  - **Beginning and Ending Sequences**

0 1 1 1 1 1 1 0

HDLC Frame Format

- **HDLC Protocol-bit stuffing**
  - **On the sending side, any time five consecutive 1's have been transmitted from the body of the message (i.e. excluding when the sender is trying to send the distinguished 01111110 sequence)**
    - **The sender inserts 0 before transmitting the next bit**
- **HDLC Protocol-**
  - **On the receiving side**
    - **5 consecutive 1's**
      - **Next bit 0 : Stuffed, so discard it**
        1 : Either End of the frame marker
        Or Error has been introduced in the bitstream

Look at the next bit
If 0 (01111110) → End of the frame marker
If 1 (01111111) → Error, discard the whole frame
The receiver needs to wait for next 01111110 before it can start receiving again

b. Explain steps of Internet check sum and suppose that the sender sends the following four frames of eight bits. Check whether the frames is accepted or not using internet check sum.
11001100 10101010 11110000 11000011.

I. Error Detection by Checksums

For error detection by checksums, data is divided into fixed sized frames or segments.

- **Sender's End** − The sender adds the segments using 1's complement arithmetic to get the sum. It then complements the sum to get the checksum and sends it along with the data frames.

- **Receiver's End** − The receiver adds the incoming segments along with the checksum using 1's complement arithmetic to get the sum and then complements it.

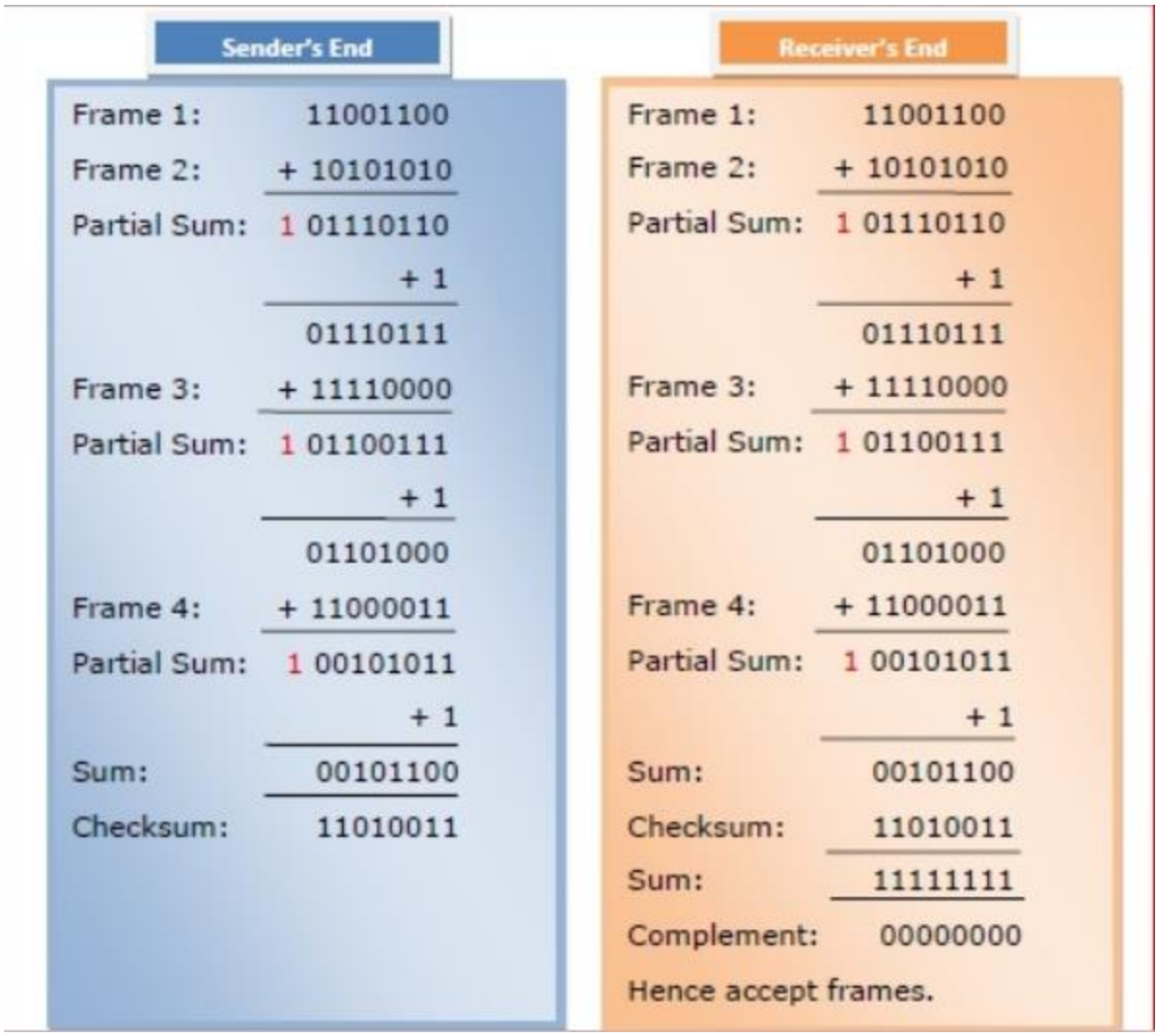
If the result is zero, the received frames are accepted; otherwise they are discarded.

Suppose that the sender wants to send 4 frames each of 8 bits, where the frames are 11001100, 10101010, 11110000 and 11000011.

The sender adds the bits using 1s complement arithmetic. While adding two numbers using 1s complement arithmetic, if there is a carry over, it is added to the sum.

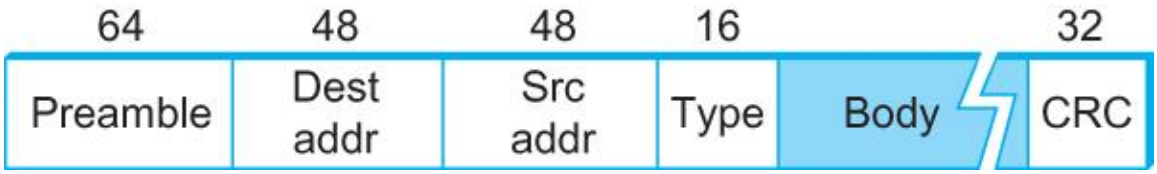
After adding all the 4 frames, the sender complements the sum to get the checksum, 11010011, and sends it along with the data frames.

The receiver performs 1s complement arithmetic sum of all the frames including the checksum. The result is complemented and found to be 0. Hence, the receiver assumes that no error has occurred.



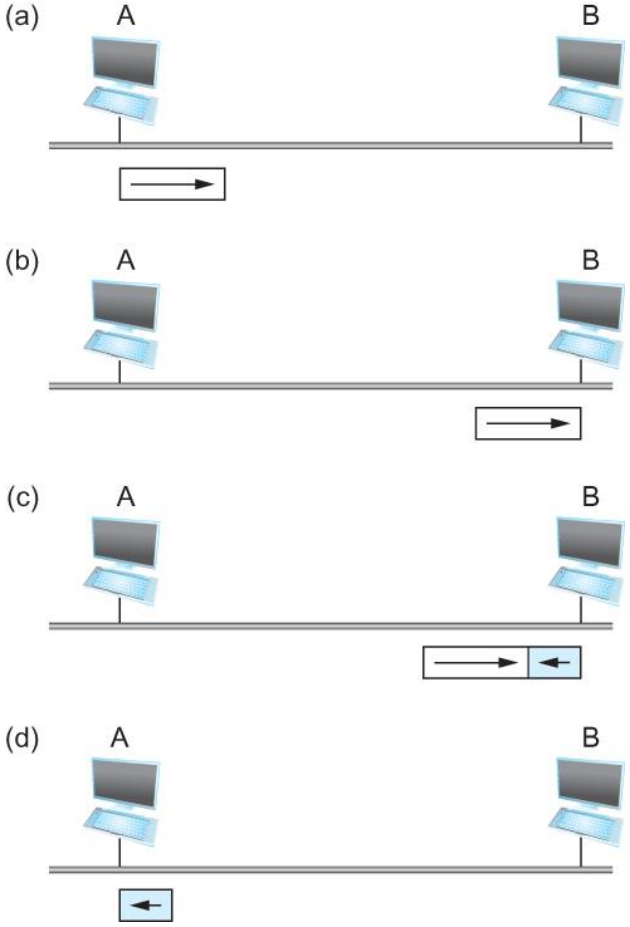
4. a. Explain Ethernet frame format and transmission algorithm.
   Frame format:

- Frame format
  - Preamble (64bit): allows the receiver to synchronize with the signal (sequence of alternating 0s and 1s).
  - Host and Destination Address (48bit each).
  - Packet type (16bit): acts as demux key to identify the higher level protocol.
  - Data (up to 1500 bytes)
    - Minimally a frame must contain at least 46 bytes of data.
    - Frame must be long enough to detect collision.
  - CRC (32bit)



- When the adaptor has a frame to send and the line is idle, it transmits the frame immediately.
  - The upper bound of 1500 bytes in the message means that the adaptor can occupy the line for a fixed length of time.
- When the adaptor has a frame to send and the line is busy, it waits for the line to go idle and then transmits immediately.
- The Ethernet is said to be 1-persistent protocol because an adaptor with a frame to send transmits with probability 1 whenever a busy line goes idle.
- Since there is no centralized control it is possible for two (or more) adaptors to begin transmitting at the same time,
  - Either because both found the line to be idle,
  - Or, both had been waiting for a busy line to become idle.
- When this happens, the two (or more) frames are said to be *collide* on the network.
- Since Ethernet supports collision detection, each sender is able to determine that a collision is in progress.
- At the moment an adaptor detects that its frame is colliding with another, it first makes sure to transmit a 32-bit jamming sequence and then stops transmission.
  - Thus, a transmitter will minimally send 96 bits in the case of collision
    - 64-bit preamble + 32-bit jamming sequence
- One way that an adaptor will send only 96 bit (called a *runt frame*) is if the two hosts are close to each other.
- Had they been farther apart,
  - They would have had to transmit longer, and thus send more bits, before detecting the collision.
- The worst case scenario happens when the two hosts are at opposite ends of the Ethernet.
- To know for sure that the frame its just sent did not collide with another frame, the transmitter may need to send as many as 512 bits.
  - Every Ethernet frame must be at least 512 bits (64 bytes) long.
    - 14 bytes of header + 46 bytes of data + 4 bytes of CRC
- A begins transmitting a frame at time t
- d denotes the one link latency
- The first bit of A's frame arrives at B at time $t + d$

- Suppose an instant before host A's frame arrives, host B begins to transmit its own frame
- B's frame will immediately collide with A's frame and this collision will be detected by host B
- Host B will send the 32-bit jamming sequence
- Host A will not know that the collision occurred until B's frame reaches it, which will happen at $t + 2 * d$
- Host A must continue to transmit until this time in order to detect the collision
  - Host A must transmit for $2 * d$ to be sure that it detects all possible collisions



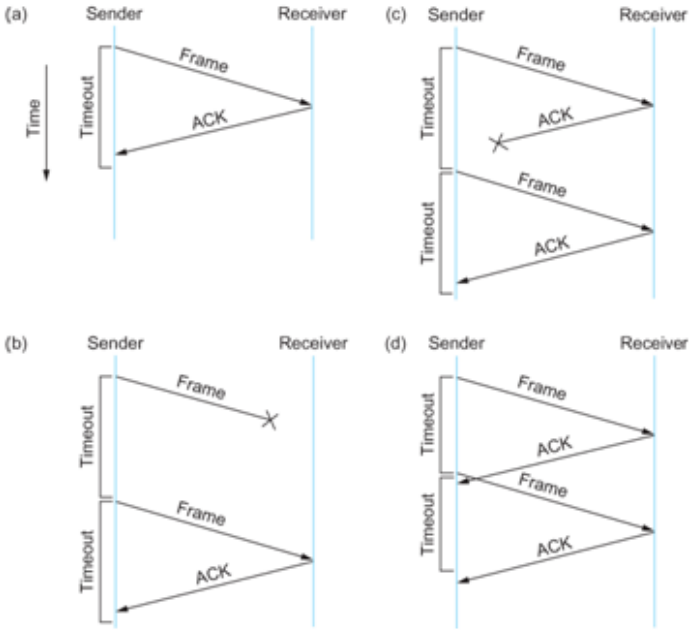
Worst-case scenario: (a) A sends a frame at time *t; (b) A's frame arrives* at B at time $t + d$; *(c) B begins transmitting at time $t + d$ and collides with A's frame;* (d) B's runt (32-bit) frame arrives at A at time $t + 2d$.

b. Explain the following:

i) Stop and wait protocol

- Idea of stop-and-wait protocol is straightforward
  - After transmitting one frame, the sender waits for an acknowledgement before transmitting the next frame.
  - If the acknowledgement does not arrive after a certain period of time, the sender times out and retransmits the original frame

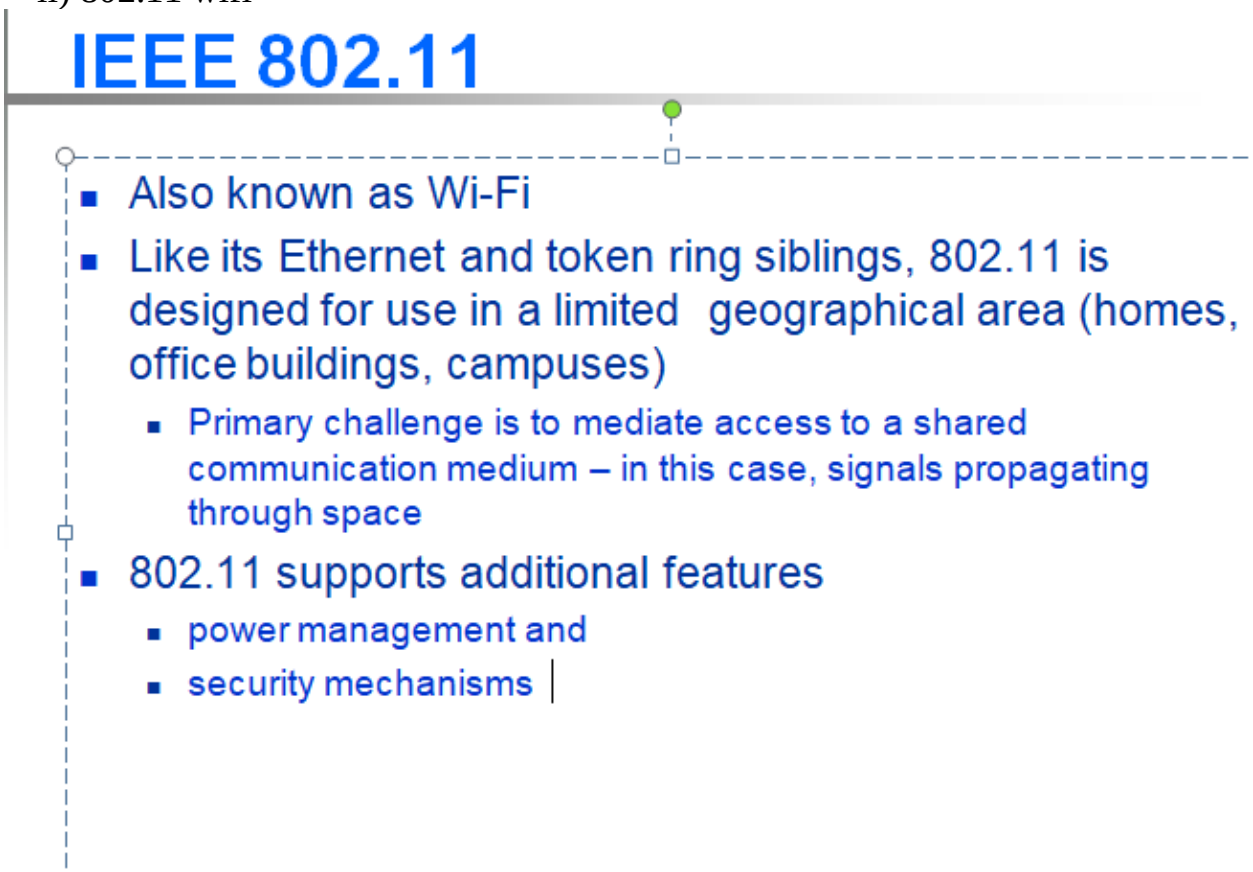
Stop and Wait Protocol



Timeline showing four different scenarios for the stop-and-wait algorithm.
(a) The ACK is received before the timer expires; (b) the original frame is lost; (c) the ACK is lost; (d) the timeout fires too soon

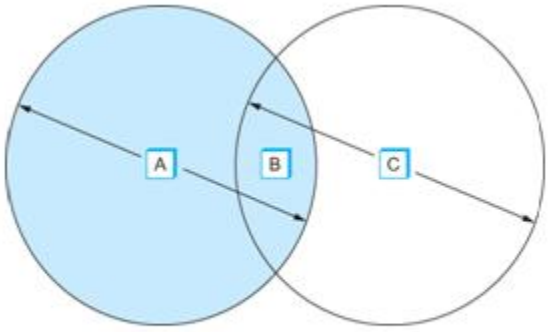
- If the acknowledgment is lost or delayed in arriving
  - The sender times out and retransmits the original frame, but the receiver will think that it is the next frame since it has correctly received and acknowledged the first frame
  - As a result, duplicate copies of frames will be delivered
  - How to solve this
  - Use 1 bit sequence number (0 or 1)
  - When the sender retransmits frame 0, the receiver can determine that it is seeing a second copy of frame 0 rather than the first copy of frame 1 and therefore can ignore it (the receiver still acknowledges it, in case the first acknowledgement was lost)

ii) 802.11 wifi



- Original 802.11 standard defined two radio-based physical layer standard
  - One using the frequency hopping
    - Over 79 1-MHz-wide frequency bandwidths
  - Second using direct sequence
    - Using 11-bit chipping sequence
  - Both standards run in the 2.4-GHz and provide up to 2 Mbps
- Then physical layer standard 802.11b was added
  - Using a variant of direct sequence 802.11b provides up to 11 Mbps
  - Uses license-exempt 2.4-GHz band
- Then came 802.11a which delivers up to 54 Mbps using OFDM
  - 802.11a runs on license-exempt 5-GHz band
- Most recent standard is 802.11g which is backward compatible with 802.11b
  - Uses 2.4 GHz band, OFDM and delivers up to 54 Mbps
- Consider the situation in the following figure where each of four nodes is able to send and receive signals that reach just the nodes to its immediate left and right
  - For example, B can exchange frames with A and C, but it cannot reach D

- C can reach B and D but not A



Example of a wireless network

- Source and Destinations addresses: each 48 bits
- Data: up to 2312 bytes
- CRC: 32 bit
- Control field: 16 bits
  - Contains three subfields (of interest)
    - 6 bit **Type** field: indicates whether the frame is an RTS or CTS frame or being used by the scanning algorithm

A pair of 1 bit fields : called **ToDS** and **FromDS**

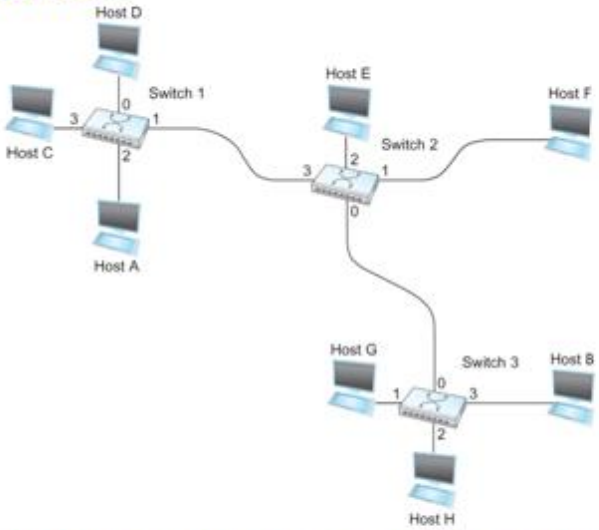
Frame Format

module-3

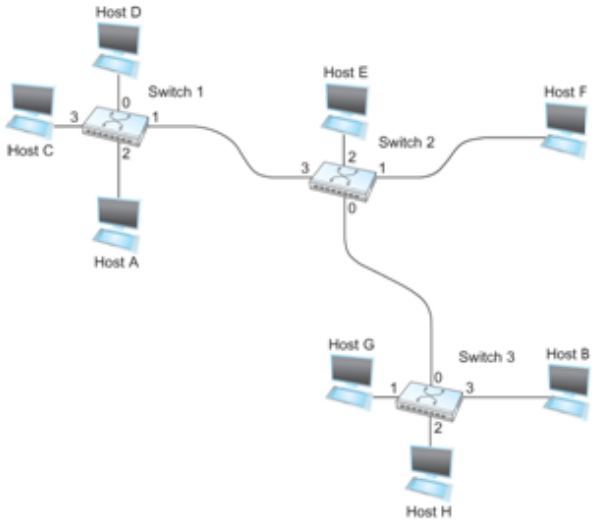
5. a. What is datagram network? Explain its characteristics.

- Datagrams
  - Key Idea
    - Every packet contains enough information to enable any switch to decide how to get it to destination
      - Every packet contains the complete destination address

An example network



- To decide how to forward a packet, a switch consults a *forwarding table* (sometimes called a *routing table*)



Destination	Port
A	3
B	0
C	3
D	3
E	2
F	1
G	0
H	0

Forwarding Table for Switch 2

Characteristics of Connectionless (Datagram) Network

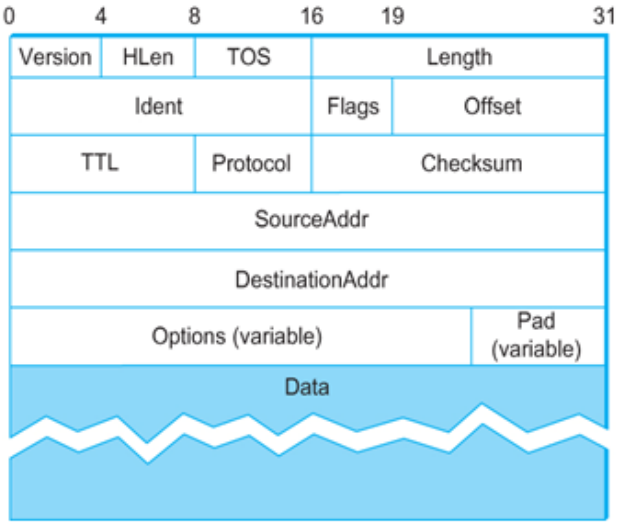
- A host can send a packet anywhere at any time, since any packet that turns up at the switch can be immediately forwarded using the **forwarding table**

- When a host sends a packet, **it does NOT know** if the network is capable of delivering it or if the destination host is even up and running
- Each packet is **forwarded independently** of previous packets that might have been sent to the same destination.
  - Thus two successive packets from host A to host B may follow completely different paths

A switch or link failure **might not have any serious effect** on communication if it is possible to find an alternate route around the failure and update the forwarding table accordingly.

b. In detail. Explain IPv4 Packet header format.

- **Version (4 bits):**
  - currently 4 or 6.
  - Also called IPv4 and IPv6
- **Hlen (4 bits):**
  - number of **32-bit words** in header
  - usually 5 32-bit words with no options
- **TOS (8 bits):**
  - type of service (not widely used)
- **Length (16 bits):**
  - number of **bytes** in this datagram including the header
- **Ident (16 bits) and Flags/Offset (16 bits):**
  - used by fragmentation



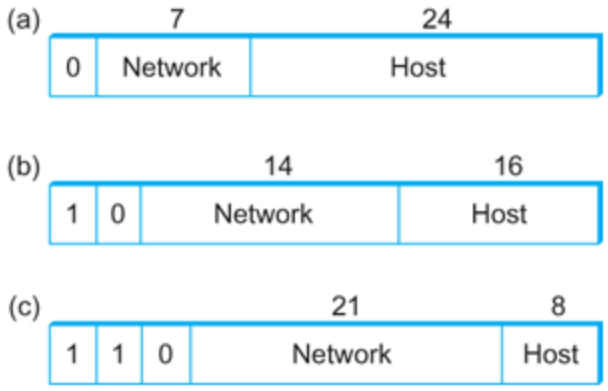
- **TTL (8 bits):**
  - number of hops/routers this packet can travel
    - discard the looping packets
  - Originally based on time, but changed to a hop-count based
  - Each router decrements it by 1
  - Discard the packet when it becomes 0
  - Default is 64
  - Problems
    - Setting it too high the packet will loop a lot

        - Setting it too low the packet will not reach the destination
- **Protocol (8 bits):**
    - demux key (TCP=6, UDP=17)
- **Checksum (16 bits):**
    - of the header only
- **DestAddr & SrcAddr (32 bits)**
    - The key for datagram delivery
    - Every packet contains a full destination address
    - Forwarding/routing decisions are made at each router
    - The source address is for the destination to know the sender and if it wants to reply to it

6. a. Explain Class A, Class B, Class C of IP Addresses.

- IP addresses Properties
    - globally unique
    - hierarchical: network + host
        - Network part: identifies the network the host is attached to
        - Host: identifies a unique host on that network
        - Ethernet addresses, even globally unique, are flat (no structure and thus no meaning) and can not be use for routing
    - Note that a router is attached to at least **two** networks, so it must have an IP address on each port/interface
        - Thus it is more precise to think of IP addresses as belonging to interfaces rather than to hosts

- Approximately, 4 Billion IP address, half are A type, ¼ is B type, and 1/8 is C type



(a) Class A (b) Class B (c) Class C

- Class A was intended for Wide Area Networks
  - Thus there should a very few of them
- Class B was intended for a modest size networks (like a campus)
- Class C is for the large number of LANs
- However, these classifications are not flexible and today's IP addresses are normally "**classless**" as we will see
- Format
  - 4 bytes, each byte is represented by a decimal number
  - Dot notation
    - 10.3.2.4
    - 128.96.33.81
    - 192.12.69.77

b. With neat diagram, explain i)ARP ii)DHCP.

ARP:

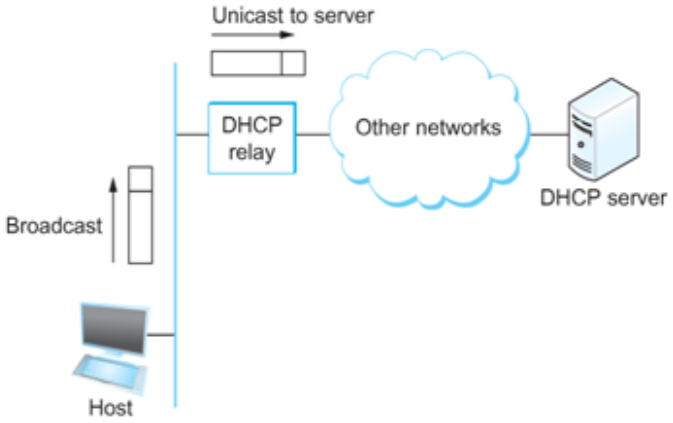
- Map IP addresses into physical addresses
- ARP (Address Resolution Protocol)
  - table of IP to physical address bindings

- The router broadcasts a request (who-has / tell) if the required IP address not in the ARP table
  - Ex., who-has 192.168.0.29 tell 192.168.0.1
- target machine (with IP 192.168.0.29 in the example) responds with its physical address (its MAC)

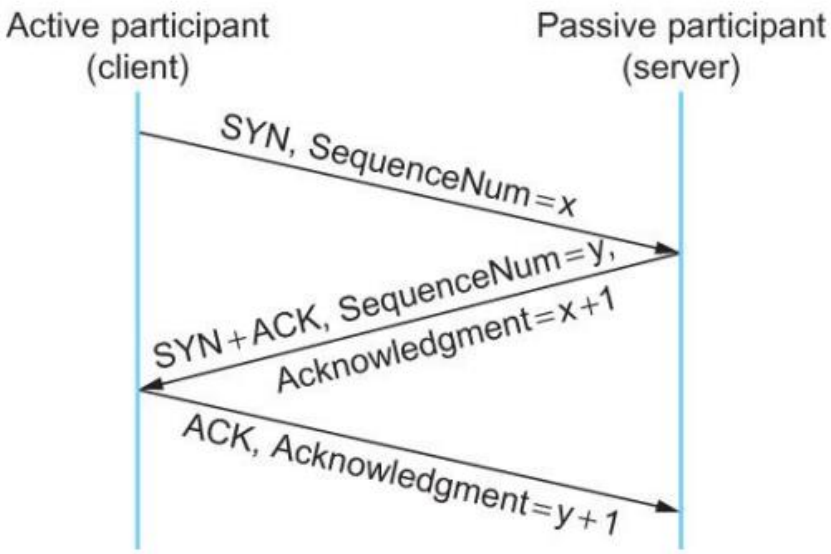
DHCP:

- Most host Operating Systems provide a way to manually configure the IP information for the host
- Drawbacks of manual configuration
  - A lot of work to configure all the hosts in a large network
  - Configuration process is error-prune
- Automated Configuration Process is required
  - Using the DHCP protocol
- DHCP server is responsible for providing configuration information to hosts
- There is at least one DHCP server for an administrative domain
- DHCP server maintains a pool/set of available addresses

- Newly booted or attached host sends DHCPDISCOVER message to a special IP address (255.255.255.255)
- DHCP relay agent unicasts the message to DHCP server and waits for the response



7. a. Explain 3-way handshaking in TCP.



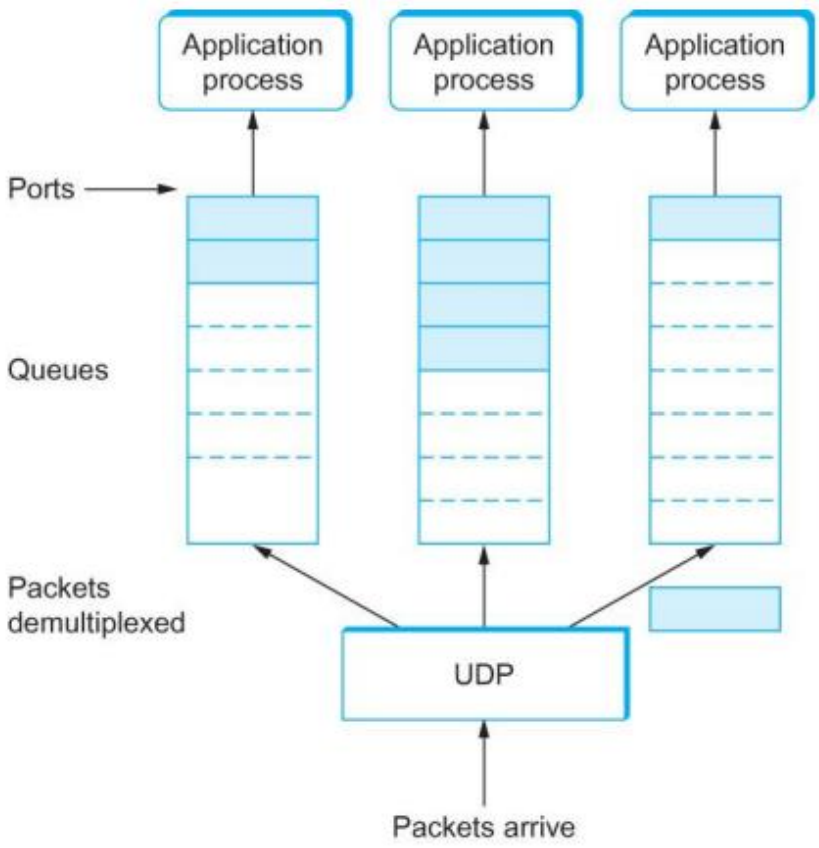
Timeline for three-way handshake algorithm

b. Explain simple demultiplexer and its header format(UDP)

Extends host-to-host delivery service of the underlying network into a process-to-process communication servic

Simple Demultiplexer (UDP)

Format for UDP header (Note: length and checksum fields should be switched)



UDP Message Queue

8. a. What is congestion? Explain Leaky Bucket algorithm.

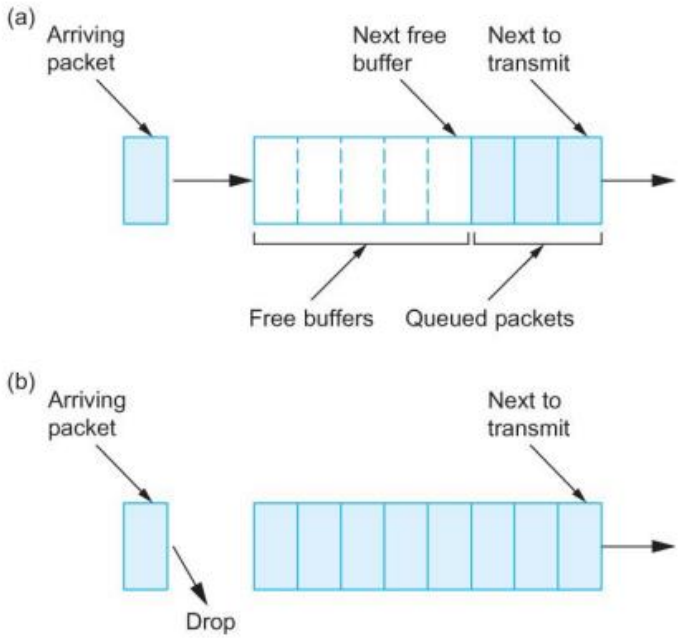
When too many packets are contending for the same link ■ The queue overflows ■ Packets get dropped ■ Network is congested! ■ Network should provide a congestion control mechanism to deal with such a situation.

b. Explain the following:
i)Queuing Discipline

The idea of FIFO queuing, also called first-come-firstserved (FCFS) queuing, is simple: ■ The first packet that arrives at a router is the first packet to be transmitted ■ Given that the amount of buffer space at each

router is finite, if a packet arrives and the queue (buffer space) is full, then the router discards that packet ■ This is done without regard to which flow the packet belongs to or how important the packet is. This is sometimes called tail drop, since packets that arrive at the tail end of the FIFO are dropped ■ Note that tail drop and FIFO are two separable ideas. FIFO is a scheduling discipline—it determines the order in which packets are transmitted. Tail drop is a drop policy—it determines which packets get dropped

Queuing Disciplines

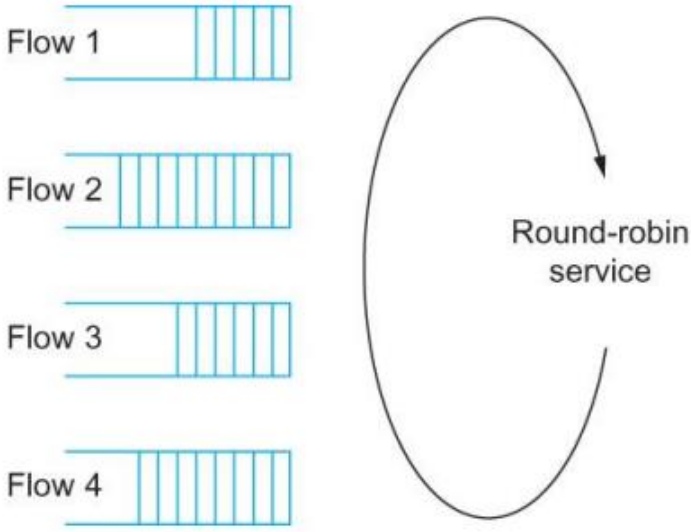


(a) FIFO queuing; (b) tail drop at a FIFO queue.

A simple variation on basic FIFO queuing is priority queuing. The idea is to mark each packet with a priority; the mark could be carried, for example, in the IP header. ■ The routers then implement multiple FIFO queues, one for each priority class. The router always transmits packets out of the highest-priority queue if that queue is nonempty before moving on to the next priority queue. ■ Within each priority, packets are still managed in a FIFO manner. A simple variation on basic FIFO queuing is priority queuing. The idea is to mark each packet with a priority; the mark could be carried, for

example, in the IP header. ■ The routers then implement multiple FIFO queues, one for each priority class. The router always transmits packets out of the highest-priority queue if that queue is nonempty before moving on to the next priority queue. ■ Within each priority, packets are still managed in a FIFO manner.

- Fair Queuing



Round-robin service of four flows at a router

ii)TCP Congestion Control

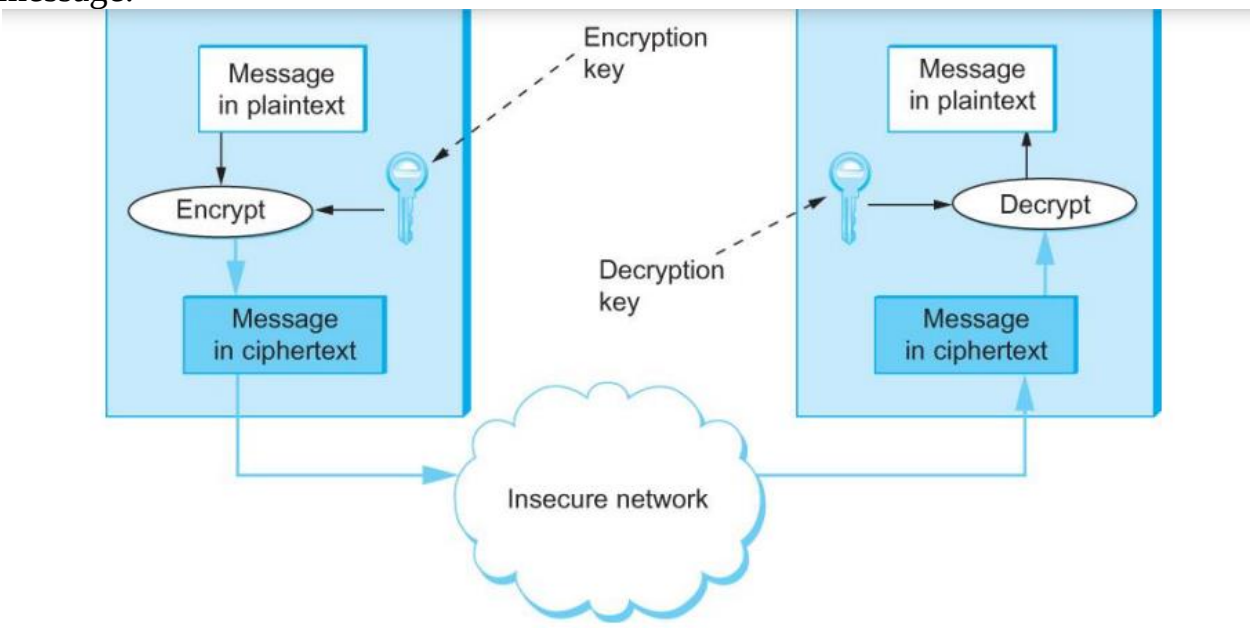
TCP congestion control was introduced into the Internet in the late 1980s by Van Jacobson, roughly eight years after the TCP/IP protocol stack had become operational. ■ Immediately preceding this time, the Internet was suffering from congestion collapse— ■ hosts would send their packets into the Internet as fast as the advertised window would allow, congestion would occur at some router (causing packets to be dropped), and the hosts would time out and retransmit their packets, resulting in even more congestion The idea of TCP congestion control is for each source to determine how much capacity is available in the network, so that it knows how many packets it can safely have in transit. ■ Once a given source has this many packets in transit, it uses the arrival of an ACK as a signal that one of its packets has left the network, and that it is therefore safe to insert a new packet into the

network without adding to the level of congestion. ■ By using ACKs to pace the transmission of packets, TCP is said to be self-clocking.
Additive Increase Multiplicative Decrease
Slow Start

Module-5

9. a. Define cipher. Explain Symmetric key cipher.

Symmetric Key Ciphers ■ In a symmetric-key cipher, both participants in a communication share the same key. In other words, if a message is encrypted using a particular key, the same key is required for decrypting the message.



Symmetric-key encryption and decryption

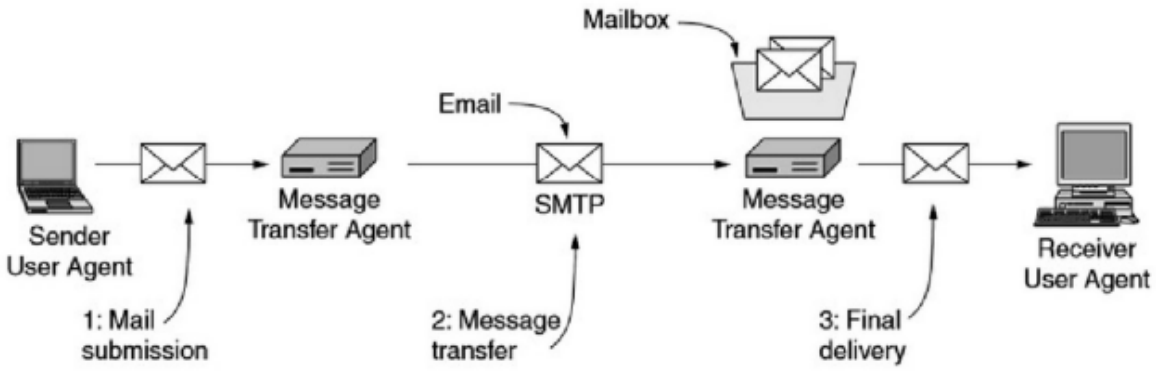
Symmetric Key Ciphers ■ Data Encryption Standard (DES) was the first, and it has stood the test of time in that no cryptanalytic attack better than brute force search has been discovered. ■ Brute force search, however, has gotten faster. DES's keys (56 independent bits) are now too small given current processor speeds Symmetric Key Ciphers ■ Advanced Encryption

Standard (AES) standard issued by NIST in 2001. ■ AES supports key lengths of 128, 192, or 256 bits, and the block length is 128 bits.

b. Write a note on:

i)SMTP –Simple Mail Transfer Protocol

Uses TCP to reliably transfer email message from client to server, port 25•SMTP [RFC 2821]: Messages must be in 7-bit ASCI• Direct transfer: sending server to receiving server. •



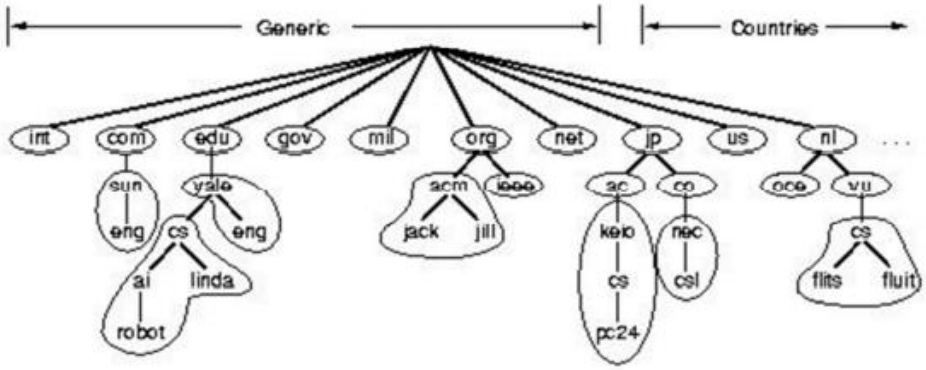
The user agent is a program that provides a graphical interface, or sometimes a text- and commandbased interface that lets users interact with the email system. It includes a means to compose messages and replies to messages, display incoming messages, and¬ organize messages by filing, searching, and discarding them. The act of sending new messages into the mail system for delivery is called mail submission.¬ The message transfer agents are typically system processes. They run in the background on mail¬ server machines and are intended to be always available. Their job is to automatically move email through the system from the originator to the recipient with SMTP (Simple Mail Transfer Protocol). Message transfer agents also implement mailing lists, in which an identical¬This is the message transfer step. copy of a message is delivered to everyone on a list of email addresses. Other advanced features are carbon copies, blind carbon copies, high-priority email, secret (i.e., encrypted) email, alternative recipients if the primary one is not currently available, and the ability for assistants to read and answer their bosses' email. Linking user agents and message transfer agents are the concepts of mailboxes and a standard¬ format for email messages. Mailboxes store the email that is received for a user. They are maintained by mail servers. User agents simply present users with a view of the contents of their mailboxes. To do this, the user agents send the mail servers commands to manipulate the mailboxes,¬ inspecting

their contents, deleting messages, and so on. The retrieval of mail is the final delivery As noted above, it is important (1) to distinguish the user interface (i.e., your mail reader) from the underlying message transfer protocols (such as SMTP or IMAP), and (2) to distinguish between this transfer protocol and a companion protocol (RFC 822 and MIME) that defines the format of the messages being exchanged.

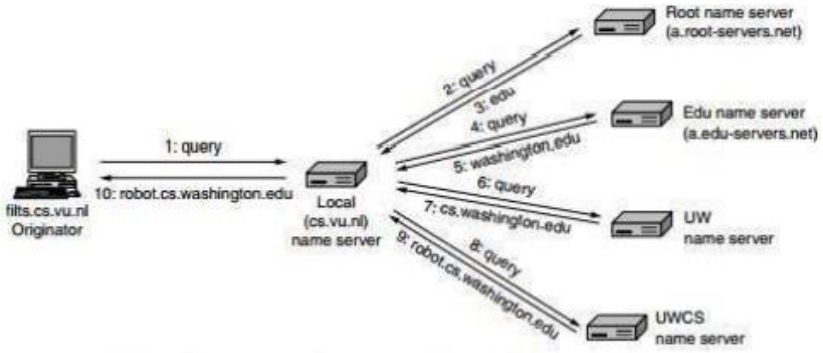
ii)DNS – Domain Name System

The “Domain Name System” is the mechanism by which Internet software translates names to attributes such as addresses and vice versa. An Internet service that translates domain names into IP addresses. Because domain names are• alphabetic, they're easier to remember. The Internet however, is really based on IP addresses. Every time you use a domain name, therefore, a DNS service must translate the name into the corresponding IP address. Name Servers: Server programs which hold information about the structure and the names.• Resolvers: Client programs that extract information from Name Servers. • Name Space: Specifications for a structured name space and data associated with the names •T here are 3 components: DNS namespace is divided into non overlapping zones. One possible way to divide the name• Each node has a label: The root node has a null label, written as “ / ” • The name space is the structure of the DNS database •DNS Name Space: space is shown below. Each circled zone contains some part

of the tree.



DNS Resolvers: DNS resolver is a server that uses the DNS protocol to query for information from DNS servers. DNS resolvers communicate with either remote DNS servers or the DNS servers or the DNS server program running on the local computer.



Example of a resolver looking up a remote name in 10 steps.

c. What is a firewall? Explain its strength and weakness of a firewall.

A firewall is a system that typically sits at some point of connectivity between a site it protects and the rest of the network, as illustrated in Figure below.

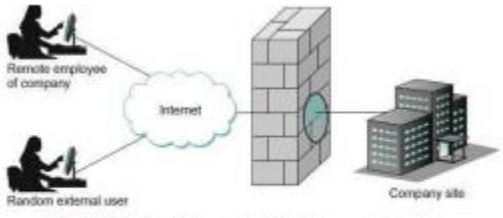


FIGURE 5.5 A firewall filters packets flowing between a site and the rest of the Internet.

It is usually implemented as an "appliance" or part of a router, although a "personal firewall" may be implemented on an end user machine. Firewall-based security depends on

the firewall being the only connectivity to the site from outside; there should be no way to bypass the firewall via other gateways, wireless connections, or dial-up connections. One way to think of a firewall is that by default it blocks traffic unless that traffic is specifically allowed to pass through. For example, it might filter out all incoming messages except those addresses to a particular set of IP addresses or to particular TCP port numbers. A firewall divides a network into a more-trusted zone internal to the firewall and a less-trusted zone external to the firewall. This is useful if you do not want external users to access a particular host or service within your site. Much of the complexity comes from the fact that you want to allow different kinds of access to different external users, ranging from the general public, to business partners, to remotely located members of your organization. A firewall may also impose restrictions on outgoing traffic to prevent certain attacks and to limit losses if an adversary succeeds in getting access inside the firewall. Firewalls may be used to create multiple zones of trust, such as a hierarchy of increasingly trusted zones. A common arrangement involves three zones of trust: the internal network, the DMZ ("demilitarized zone"); and the rest of the Internet. The DMZ is used to hold services such as DNS and email servers that need to be accessible to the outside. Both the internal network and the outside world can access the DMZ, but hosts in the DMZ cannot access the internal network; therefore, an adversary who succeeds in compromising a host in the exposed DMZ still cannot access the internal network. The DMZ can be periodically restored to a clean state. Firewalls filter based on IP, TCP, and UDP information, among other things. They are configured with a table of addresses that characterize the packets they will, and will not, forward. By addresses, we mean more than just the destination's IP address, although that is one possibility. Generally, each entry in the table is a 4-tuple: It gives the IP address and TCP (or UDP) port number for both the source and destination. For example, a firewall might be configured to filter out (not forward) all packets that match the following description: (192.12.13.14, 1234, 128.7.6.5, 80) This pattern says to discard all packets from port 1234 on host 192.12.13.14 addressed to port 80 on host 128.7.6.5. (Port 80 is the well known TCP port for HTTP.) Of course, it's often not practical to name every source host whose packets you want to filter, so the patterns can include wildcards. For example, (*. *, 128.7.6.5, 80) says to filter out all packets addressed to port 80 on 128.7.6.5, regardless of what source host or port sent the packet. Stateless firewalls are designed to protect networks based on static information such as source and destination. Whereas stateful firewalls filter packets based on the full context of a given network connection, stateless firewalls filter packets based on the individual

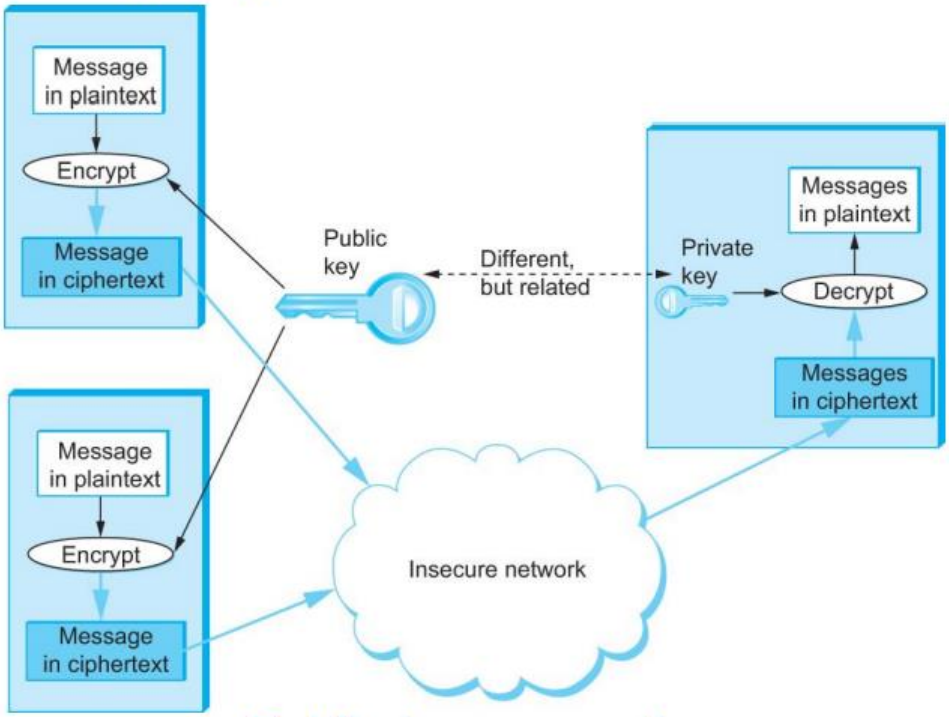
packets themselves. A stateful firewall is a firewall that monitors the full state of active network connections. This means that stateful firewalls are constantly analyzing the complete context of traffic and data packets, seeking entry to a network rather than discrete traffic and data packets in isolation. Once a certain kind of traffic has been approved by a stateful firewall, it is added to a state table and can travel more freely into the protected network. Traffic and data packets that don't successfully complete the required handshake will be blocked. By taking multiple factors into consideration before adding a type of connection to an approved list, such as TCP stages, stateful firewalls are able to observe traffic streams in their entirety. However, this method of protection does come with a few vulnerabilities. For example, stateful firewalls can fall prey to DDoS attacks due to the intense compute resources and unique softwarenetwork relationship necessary to verify connections. Many client/server applications dynamically assign a port to the client. If a client inside a firewall initiates access to an external server, the server's response would be addressed to the dynamically assigned port. This poses a problem: How can a firewall be configured to allow an arbitrary server's response packet but disallow a similar packet for which there was no client request? This is not possible with a stateless firewall, which evaluates each packet in isolation. It requires a stateful firewall, which keeps track of the state of each connection. An incoming packet addressed to a 5.2.1 STRENGTHS AND WEAKNESSES OF FIREWALLS dynamically assigned port would then be allowed only if it is a valid response in the current state of a connection on that port. Modern firewalls also understand and filter based on many specific application-level protocols such as HTTP, Telnet, or FTP. They use information specific to that protocol, such as URLs in the case of HTTP, to decide whether to discard a message. While a firewall is an integral part of an organization's security architecture and plays a vital role in protection of assets, it has strengths and weaknesses too. Restricting access to specified services. Access can even be granted selectively based on• Helping to enforce security and safety policies of an organization. •The strengths of a firewall: Their singularity of purpose which means that companies need not make any compromises•authentication functionality. Its appraisal capacity which results in an organization getting to know and monitor all the•between usability and security. Being a notification system which can alert people concerned about specific events.•traffic that sifts through their networks. An inability to fend off attacks from within the system that it is meant to protect. This could•The weaknesses of a firewall: take the form of people

granting unauthorized access to other users within the network or It cannot circumvent poorly structured security policies or bad administrative practices. For• It can only stop the intrusions from the traffic that actually passes through them. •social engineering assaults or even an authorized user intent on malafide use of the network. instance, if a company has a very loosely knit policy on security and over-permissive rules, As long as a communication or transaction has been permitted, a firewall has no ability to•then a firewall cannot protect data or the network. protect the system against it. For instance, if a firewall has been built to allow emails to come through, it cannot detect a virus or a Trojan within that email

10.a. Explain how public key authentication work.

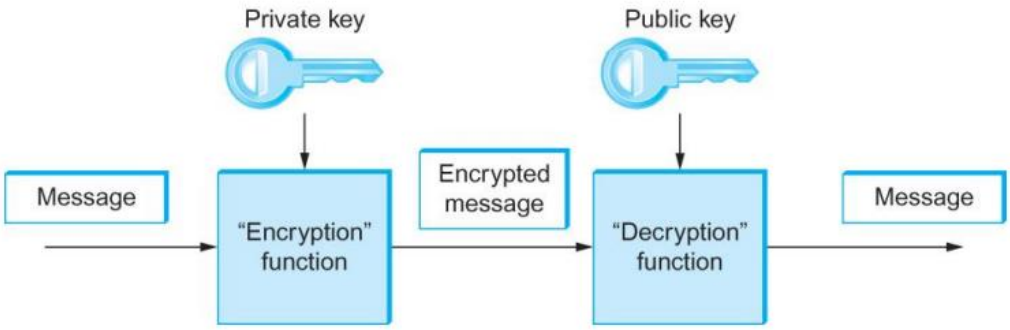
Public Key Ciphers ■ An alternative to symmetric-key ciphers is asymmetric, or public-key, ciphers. ■ Instead of a single key shared by two participants, a public-key cipher uses a pair of related keys, one for encryption and a different one for decryption. ■ The pair of keys is “owned” by just one participant. ■ The owner keeps the decryption key secret so that only the owner can decrypt messages; that key is called the private key. Public Key Ciphers ■ The owner makes the encryption key public, so that anyone can encrypt messages for the owner; that key is called the public key. ■ Obviously, for such a scheme to work it must not be possible to deduce the private key from the public key. ■ Consequently any participant can get the public key and send an encrypted message to the owner of the keys, and only the owner has the private key necessary to decrypt it.

- Public Key Ciphers



Public-key encryption

- Public Key Ciphers



Authentication using public keys

Public Key Ciphers ■ An important additional property of public-key ciphers is that the private key can be used with the encryption algorithm to encrypt messages so that they can only be decrypted using the public "encryption" key. ■ This property clearly wouldn't be useful for confidentiality since anyone with the public key could decrypt such a message. ■ This property is, however, useful for authentication since it tells the receiver of such a message that it could only have been created by the owner of the keys

b. What are the security threats in Internet working?

Computer networks are typically a shared resource used by many applications representing different interests. ■ The Internet is particularly widely shared, being used by competing businesses, mutually antagonistic governments, and opportunistic criminals. ■ Unless security measures are taken, a network conversation or a distributed application may be compromised by an adversary. Consider some threats to secure use of, for example, the World Wide Web. ■ Suppose you are a customer using a credit card to order an item from a website. ■ An obvious threat is that an adversary would eavesdrop on your network communication, reading your messages to obtain your credit card information. ■ It is possible and practical, however, to encrypt messages so as to prevent an adversary from understanding the message contents. A protocol that does so is said to provide confidentiality. Even with confidentiality there still remain threats for the website customer. ■ An adversary who can't read the contents of your encrypted message might still be able to change a few bits in it, resulting in a valid order for, say, a completely different item or perhaps 1000 units of the item. ■ There are techniques to detect, if not prevent, such tampering. ■ A protocol that detects such message tampering provides data integrity. Another threat to the customer is unknowingly being directed to a false website. ■ This can result from a DNS attack, in which false information is entered in a Domain Name Server or the name service cache of the customer's computer. ■ This leads to translating a correct URL into an incorrect IP address—the address of a false website. ■ A protocol that ensures that you really are talking to whom you think you're talking is said to provide authentication. ■ Authentication entails integrity since it is meaningless to say that a message came from a certain participant if it is no longer the same message The owner of the website can be attacked as well.

Some websites have been defaced; the files that make up the website content have been remotely accessed and modified without authorization. ■ That is an issue of access control: enforcing the rules regarding who is allowed to do what. ■ Websites have also been subject to Denial of Service (DoS) attacks, during which would-be customers are unable to access the website because it is being overwhelmed by bogus requests. ■ Ensuring a degree of access is called availability. In addition to these issues, the Internet has notably been used as a means for deploying malicious code that exploits vulnerabilities in end-systems. ■ Worms, pieces of self-replicating code that spread over networks, have been known for several decades and continue to cause problems, as do their relatives, viruses, which are spread by the transmission of "infected" files. ■ Infected machines can then be arranged into botnets which can be used to inflict further harm, such as launching DoS attacks.

c. Write a note on WWW.

World Wide Web ■ The World Wide Web has been so successful and has made the Internet accessible to so many people that sometimes it seems to be synonymous with the Internet. ■ In fact, the design of the system that became the Web started around 1989, long after the Internet had become a widely deployed system. ■ The original goal of the Web was to find a way to organize and retrieve information, drawing on ideas about hypertext—interlinked documents—that had been around since at least the 1960s. The core idea of hypertext is that one document can link to another document, and the protocol (HTTP) and document language (HTML) were designed to meet that goal. ■ One helpful way to think of the Web is as a set of cooperating clients and servers, all of whom speak the same language: HTTP. ■ Most people are exposed to the Web through a graphical client program, or Web browser, like Safari, Chrome, Firefox or Internet Explorer Clearly, if you want to organize information into a system of linked documents or objects, you need to be able to retrieve one document to get started. ■ Hence, any Web browser has a function that allows the user to obtain an object by "opening a URL." ■ URLs (Uniform Resource Locators) are so familiar to most of us by now that it's easy to forget that they haven't been around forever. ■ They provide information that allows objects on the Web to be located, and they look like the following: ■ http://www.cs.princeton.edu/index.html

If you opened that particular URL, your Web browser would open a TCP connection to the Web server at a machine called www.cs.princeton.edu and immediately retrieve and display the file called index.html. ■ Most files on the Web contain images and text and many have other objects such as audio and video clips, pieces of code, etc. ■ They also frequently include URLs that point to other files that may be located on other machines, which is the core of the “hypertext” part of HTTP and HTML