DHANALAKSHMI SRINIVASAN ENGINEERING COLLEGE, PERAMBALUR - 621212

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CS6551 – COMPUTER NETWORKS

QUESTION BANK

UNIT I – FUNDAMENTALS AND LINK LAYER

PART – A

1. Distinguish between packet switched and circuit switched network *(Apr/May-17)*

Circuit Switching	Packet Switching(Datagram type)
Dedicated path	No Dedicated path
Path is established for entire conversation	Route is established for each packet
Call setup delay	packet transmission delay
Overload may block call setup	Overload increases packet delay
Fixed bandwidth	Dynamic bandwidth
No overhead bits after call setup	overhead bits in each packet

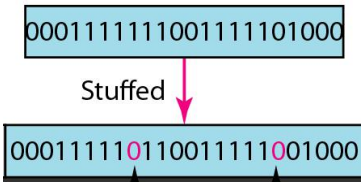
2. What is meant by bit stuffing? Give example *(Apr/may-17)*

Bit Stuffing

To prevent occurrence of bit pattern 01111110 as part of frame body, bit stuffing is used. In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added.

✓ This extra stuffed bit is eventually removed from the data by the receiver. The real flag 01111110 is not stuffed by the sender and is recognized by the receiver If a bit such as 01111111 arrives, then an error has occurred and the

✓ frame is discarded.



3. List the services provided by data link layer . *(Nov/Dec-16)*

- Framing, link access
- Reliable delivery between two physically connected
- devices:
- Flow Control:
- Error Detection
- Error Correction

4. Write the mechanism of stop and wait flow control. *(Nov/Dec-16)*

In this method of flow control, the sender sends a single frame to receiver & waits for an acknowledgment.

- frame is sent by sender only when acknowledgment of previous frame is received.
- This process of sending a frame & waiting for an acknowledgment continues as long as the sender has data to send.

- To end up the transmission sender transmits end of transmission (EOT) frame.

5. Define flow control? *(Apr/may-16)*

Flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver.
It prevents a fast sender from overwhelming a slow receiver with frames.

6. Write the parameters to measure network performance. *(May/Jun-16)*

- Bandwidth and Latency
- Delay × Bandwidth Product
- Effective end-to-end throughput
- Transfer time

7. Define protocol. *(Nov/Dec-15)*

The abstract objects that make up the layers of a network system are called *protocols*. Each protocol defines two different *interfaces*.

- *Service* interface that specifies the set of operations
- *Peer-to-peer* interface for messages to be exchanged amongst peers Protocol is a set of rules that govern communications between devices.

8. What do you mean by error control? *(Nov/Dec-15)*

The purpose of error control is to ensure that the information received by the receiver is exactly the information transmitted by the sender. As the communication channel is highly unreliable, the receiver must be able to deal with the received data, if it contains error. The term *error control* is defined as the process of identification or correction of error occurred in the transmitted data. There are two types of error control mechanisms

- Forward error control
- Feedback error control

9. What is Framing? *(Nov/Dec-13, Nov/Dec-14)*

- Break down a stream of bits into smaller, digestible chunks called frames
- Allows the physical media to be shared
- Provides manageable unit for error handling

- Wraps payload up with some additional information
- Basic unit of reception

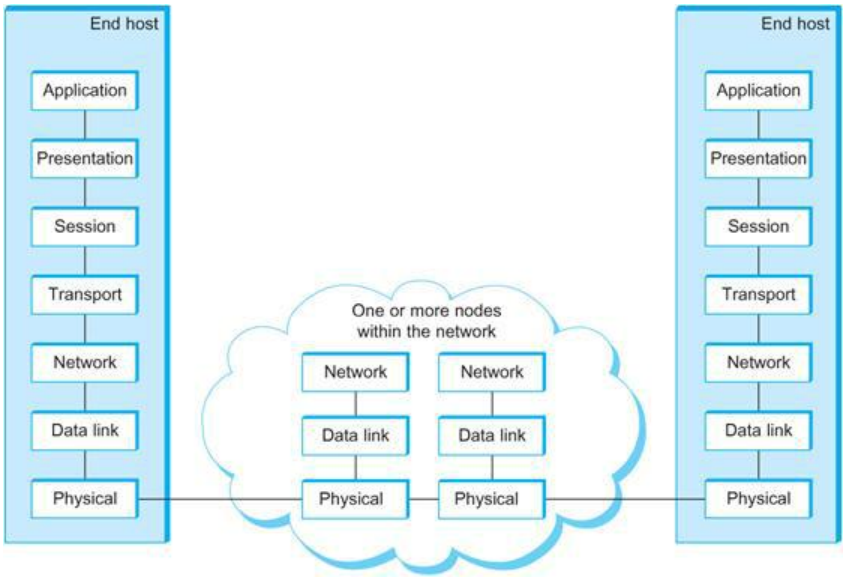
10. List the design issues of data link layer. *(Nov/Dec-12)*

- Framing—Breaking the bit stream into frames with well-defined frame boundary
- Error Control—Involves error detection and correction methods using redundant information to verify integrity of the data.
- Flow Control—Mechanisms that prevents fast sender from swamping a slow receiver with frames
- Access Control—Resolve conflicts and collisions that arise when multiple nodes compete to transmit data over a shared link.

PART –B (Answers as Hint)

1. Draw the OSI layer and explain the functionalities of each layer in detail. *(Nov/Dec-16, Nov/Dec-15, May/Jun-13)*

The ISO defined a common way to connect computers, called the Open Systems Interconnection (OSI) architecture. (eg. public X.25 network).
It defines partitioning of network functionality into seven layers.
The bottom three layers, i.e., physical, data link and network are implemented on all nodes on the network including switches.



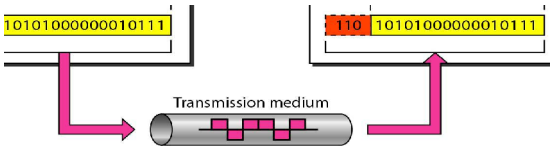
Physical Layer

It coordinates the functions required to carry a bit stream over a physical medium.

Encoding—To be transmitted, bits must be encoded into signals, electrical or optical. *Data rate*—It defines the transmission rate (number of bits sent per second).

Physical topology—It defines how devices are connected (mesh, star, ring, bus or hybrid) *Transmission mode*—Defines the direction of transmission between two devices:

simplex, half-duplex, or full-duplex



Data Link Layer

The data link layer transforms a raw transmission facility to a *reliable* link.

Framing—The bit stream is divided into manageable data units called *frames*. *Physical addressing*—Header contains physical address of sender and receiver

Flow control—If receiving rate is less than the transmission rate, flow control mechanism avoids sender overwhelming the receiver.

Error control—Redundant information is put as trailer to detect and retransmit damaged/lost frames and to recognize duplicate frames.

Access control—When two or more devices are connected to the same link, link layer protocols determines which device has control over the link at any given time.

Network Layer

It is responsible for source-to-destination delivery of a data unit called *packet*.

Logical addressing—A packet is identified across the network using logical addressing system provided by network layer and is used to identify the end systems.

Routing—Routers prepare routing table to send packets to their destination.

Transport Layer

Transport layer is responsible for *process-to-process* delivery of the entire message.

Port addressing—It includes a service-point or *port* address so that a process from one computer communicates to a specific process on the other computer.

Segmentation and reassembly—A message is divided into transmittable *segments*, each containing a sequence number. These numbers enable the transport layer to reassemble the message correctly at the destination and to identify which were lost/corrupt.

Connection control—Protocols can be either connectionless or connection-oriented.

Session Layer

It establishes, maintains, and synchronizes interaction among communicating systems. *Dialog control*—It allows two systems to enter into a dialog and communicate

Synchronization—Allows to add checkpoints to a stream of data. In case of a crash data is retransmitted from the last checkpoint.

Binding—binds together the different streams that are part of a single application. For example, audio and video stream are combined in a teleconferencing application.

Presentation Layer

It is concerned with syntax and semantics of the information exchanged between peers. *Translation*—Because different computers use different encoding systems, the

presentation layer is responsible for interoperability between these encoding methods.

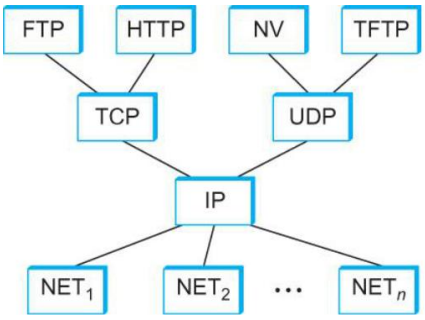
Encryption—To carry sensitive information, a system ensures privacy by encrypting the message before sending and decrypting at the receiver end.

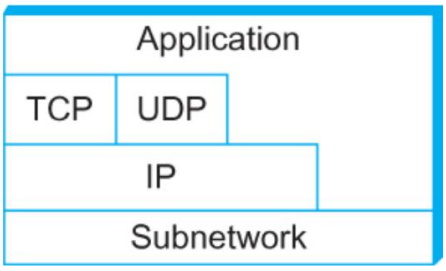
Compression—Data compression reduces the number of bits contained in the information. It is particularly important in multimedia transmission.

Application Layer

The application layer enables the user, whether human or software, to access the network. It provides user interface and support for services such as electronic mail, remote file
access, shared database management and several types of distributed
services. Composes a host of application protocols.

2. Explain the layers of TCP/IP (or) Internet architecture in detail. (April/May 15, April/May 17)





Features

Internet architecture is a four layered model, also known as TCP/IP architecture.
It evolved out of a packet-switched network called ARPANET.

TCP/IP does not enforce *strict layering*, i.e., applications are free to bypass transport layer and can directly use IP or any of the underlying networks.

IP layer serves as focal point in the architecture.

- Defines a common method for exchanging packets to any type of network o Segregates host-to-host delivery from process-to-process delivery.

For any protocol to be added to the architecture, it must also be accompanied by at least one working implementation of the specification. Thus efficiency is ensured.

Layers

Subnetwork TCP/IP does not define any specific protocol for the lowest level.

- All standard and proprietary protocols such as Ethernet, FDDI, etc are supported.
- The protocols are generally implemented by a combination of

hardware/software. *IP* The major protocol in TCP/IP is Internetworking Protocol (IP).

- It supports the interconnection of multiple networking technologies into a logical
  internetwork.
- It is an unreliable and connectionless protocol.
- IP sends data in packets called *datagrams,* each of which is transported separately and independently.
- Other protocols supported in this layer are ARP, RARP, ICMP and IGMP.

Transport layer is responsible for delivery of a message from one process to another process. The two protocols supported in this layer are:

- *Transmission Control Protocol* (TCP) for connection-oriented reliable byte-
  stream channel.
- *User Datagram Protocol* (UDP) for connectionless unreliable datagram delivery channel.

Application supports a wide range of protocols such as FTP, TFTP, Telnet (remote login), SMTP, etc., that enables the interoperation of popular applications.

3. *Discuss the factors that affect performance of the network. (Nov/Dec 16)*

Bandwidth and Latency

Performance of a network is measured in terms of *bandwidth* and *latency*.

Bandwidth refers to number of bits that can be transmitted over the network within a certain period of time (*throughput*).

Bandwidth also determines how *long* it takes to transmit each bit. For example, each bit on a 1-Mbps link is 1µs wide, whereas each bit on a 2-Mbps link is 0.5µs wide.

Latency refers to how long it takes for the message to travel to the other end (*delay*).

It is a factor of propagation delay, transmission time and queuing delay

Latency = Propagation + Transmit + Queue

Speed of light propagation depnds on medium (vaccum/copper cable/optical fiber) in which it travels and distance.

Propagation = Distance / SpeedOfLight

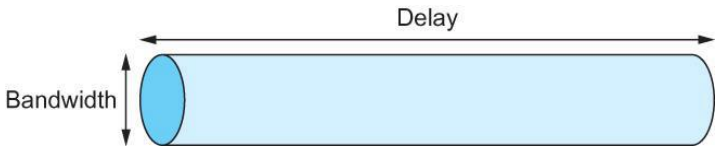
Transmission time depends upon bandwidth and packet size.

Transmit = Size / Bandwidth

Queuing delay occurs at switches and routers, since packets are stored before forwarded. Round Trip Time (RTT) is time taken for the message to travel to the other end and get back.

For applications that have minimal data transfer, latency dominates performance, whereas for bulk data transfers, bandwidth dominates performance.

Delay × Bandwidth Product



Consider a pipe, in which bandwidth is given by diameter and delay corresponds to length of the pipe.

The delay × bandwidth product specifies the number of bits in transit. It corresponds to how much the sender should transmit before the first bit is received at the other end.

If receiver signals the sender to stop, it would still receive RTT × bandwidth of data.

For example, for a cross-country fiber with 10 Gbps bandwidth, distance of 4000 km, the RTT is 40 ms and RTT × bandwidth is 400 Mb.

High Speed Networks

High speed networks enhances the bandwidth for applications but latency remains fixed. For example, when a 1 MB file is transmitted over a 1 Mbps link takes 80 RTTs, whereas the same file over a 1 Gbps links falls short of 1 RTT.

Effective end-to-end throughput that can be achieved is given as

Throughput = TransferSize / TransferTime

TransferTime includes latency as well as setup time. It is computed as

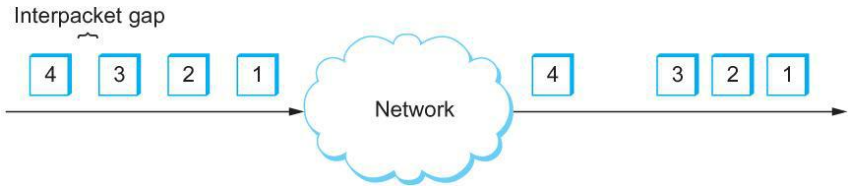
TransferTime = RTT + 1/Bandwidth × TransferSize

Application Performance Needs

Applications generally require as much bandwidth provided by the network.

Video streams are generally compressed but the flow rate varies due to details, compression algorithm used, etc. The average bandwidth could be determined, but instantaneous bursty traffic should be accounted for.

In some cases, the latency varies from packet to packet, known as *jitter*. Suppose that the packets being transmitted over the network contain video frames, the receiver will not be able to display, if a frame arrives late. If the receiver knows the latency that packets may experience, then it can delay playing first frame of the video. Thus jitter factor is smoothened out by buffering.



4. ***Explain error detection methods in detail with example (April/May 15, May/June 16)***

Error detection is only to see if data is corrupted or not. A single-bit or burst error is immaterial.

Sender adds *k* redundant bits for *n* data bits (*k* << *n*) to a frame, which is used by the receiver to determine if errors are there or not.

Two-Dimensional Parity

Data is divided into seven byte segments.

Even parity is computed for all bytes (Vertical Redundancy Check).

Even parity is also calculated for each bit position across each of the bytes (Longitudinal Redundancy Check).

Thus a parity byte for the entire frame, in addition to a parity bit for each byte is sent.

Receiver computes row and column parities for data bits. If all parity bits and parity byte match, then the frame is accepted else discarded.

Two-dimensional parity catches all 1, 2 and 3-bit errors, and most 4-bit errors.

Internet Checksum

16-bit Internet checksum is widely used by UDP and not in link layer.

Sender

Given data is divided into 16-bit words. Initial *checksum* value is 0.

All words are added using one's complement arithmetic. Carries (if any) are wrapped and added to the sum.

The complement of sum is known as *checksum* and is sent with data

Receiver

The message (including checksum) is divided into 16-bit words. All words are added using one's complement addition.

The sum is complemented and becomes the *new checksum*.

If the value of checksum is 0, the message is accepted, otherwise it is rejected.

7	0111		0111
11	1011		1011
12	1100		1100
6	0110		0110
Initial Checksum	0000	*Received Checksum*	**1001**
Sum	**10**0100	*Sum*	**10**1101
Carry	10	*Carry*	10
Sum	0110	*Sum*	1111
Checksum	**1001**	*New Checksum*	0000
Sender		*Receiver*	

Analysis

Checksum is well-suited for software implementation and is not strong as CRC.

If value of one word is incremented and another word is decremented by the same amount, the errors are not detected because sum and checksum remain the same.

Cyclic Redundancy Check (CRC)

CRC developed by IBM uses the concept of finite fields.

A *n* bit message is represented as a polynomial of degree *n* - 1.

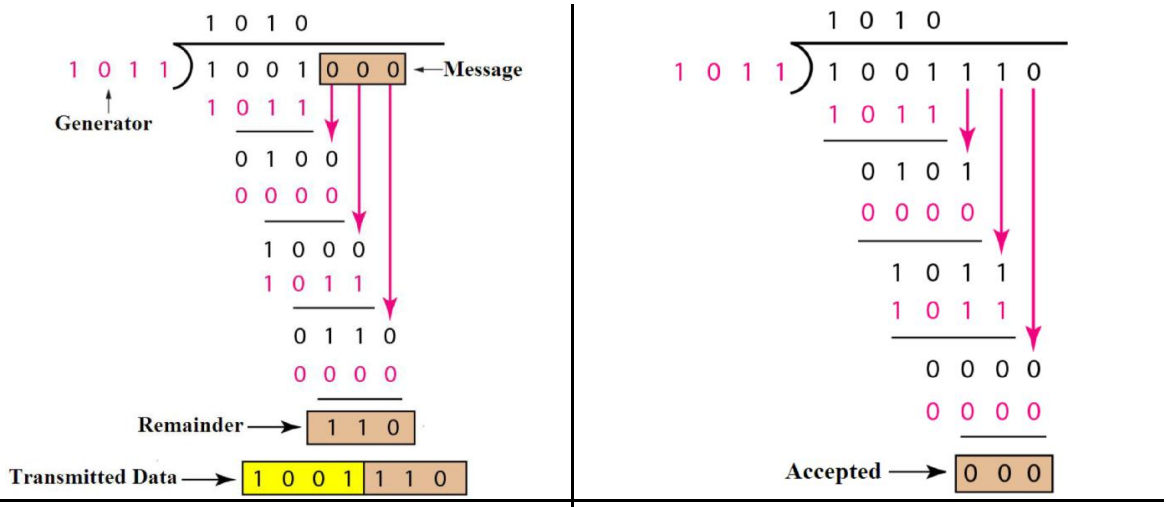
Message M(*x*) is represented as a polynomial by using the value of each bit as coefficient for each term. For example, 10011001 is represented as $x^7 + x^4 + x^3 + 1$

For calculating a CRC, sender and receiver agree on a divisor polynomial, *C*(*x*) of degree *k* such that *k* $n - 1$

Sender

Multiply M(*x*) by x^k i.e., append *k* zeroes. Let the modified poly be M'(*x*) Divide M'(*x*) by C(*x*) using XOR operation. The remainder has *k* bits

Subtract the remainder from M'(*x*) using XOR, say T(*x*) and transmit T(*x*) with *n* + *k* bits.



Sender | *Receiver*

Receiver

Divide the received polynomial T(*x*) by C(*x*) as done in sender If the remainder is non-zero then discard the frame

If zero, then no errors and redundant bits are removed to obtain data

Divisor Polynomial

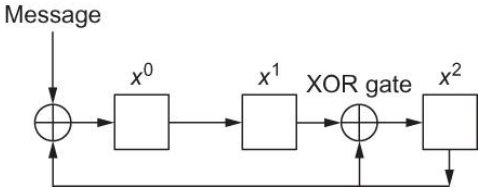
Divisor polynomial C(*x*) should has the following error-detecting properties:

- All single-bit errors, as long as the x^k and x^0 terms have nonzero coefficients.
- Any "burst" error for which the length of the burst is less than *k* bits.
- Any odd number of errors, as long as C(*x*) contains the factor (*x* + 1)

It is implemented in hardware using a *k*-bit shift register and XOR gates.

Widely used in networks such as LANs and WANs.

Different versions of CRC are CRC-8, CRC-10, CRC-12, CRC-16, and CRC-32.



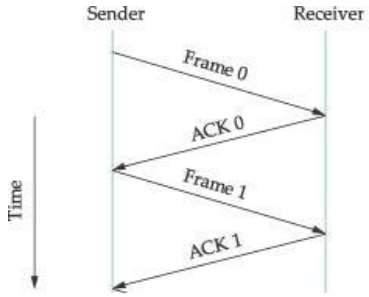
5. Explain various flow control mechanism or reliable transmission. (Nov/Dec 16, Nov/Dec 15)

Stop and Wait ARQ

The sender keeps a copy of the frame and then transmits it.

The sender waits for an acknowledgment before transmitting the next frame.

If acknowledgment does not arrive before timeout, the sender retransmits the frame.



Scenarios

a) ACK is received before the timer expires. The sender sends the next frame.
b) The frame gets lost in transmission. Sender eventually times out and retransmits frame.
c) ACK frame gets lost. The sender eventually times out and retransmits the frame.
d) The sender times out soon before ACK arrives and retransmits the frame.

Sequence number

In scenarios (c) and (d), since the receiver has acknowledged the received frame, it treats the arriving frame as the next one. This leads to duplicate frames.

To address duplicate frames, the header for a stop-and-wait protocol includes a 1-bit sequence number (0 or 1) based on modulo-2 arithmetic.

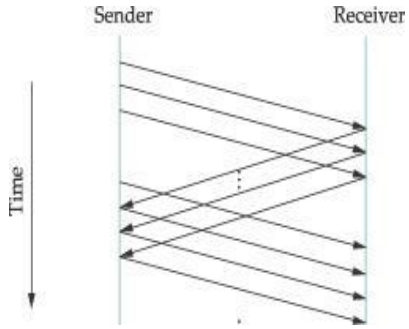
Drawbacks

It allows the sender to have only one outstanding frame on the link at a time

Inefficient if the channel has a large bandwidth and the round-trip delay is long.

Sliding window

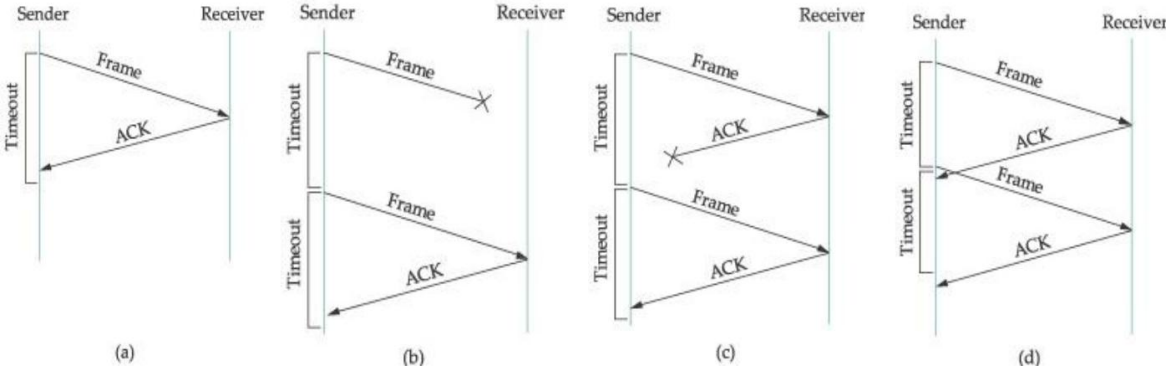
To improve efficiency, multiple frames must be in transition while waiting for an acknowledgment. Sliding window protocol makes this possible.



The *window* defines range of sequence numbers for both sender and receiver to deal with. The window position change (*slides*) due to transmission of frame and acknowledgement

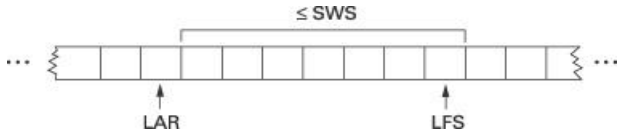
Sender

The sender assigns a sequence number SeqNum to each frame.



A *timer* with each frame it transmits, and retransmits the frame on timeout. It maintains three state variables:

- The send window size SWS gives the upper bound on the number of outstanding frames that the sender can transmit.
- LAR denotes the sequence number of the last acknowledgment received. o LFS denotes the sequence number of the last frame sent.
- The invariant LFS LAR SWS is always maintained



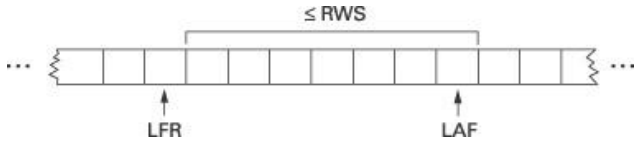
When an acknowledgment arrives, the sender moves LAR to the right, thereby allowing the sender to transmit the subsequent frames.

The sender buffers up to SWS frames (for retransmission), until they are acknowledged.

Receiver

Similarly the receiver maintains three state variables:

- The receive window size RWS gives the upper bound on number of out-of-order frames that the receiver is willing to accept.
- LAF denotes acceptable frame with the largest sequence number
- LFR denotes sequence number of the last frame received
- The invariant LAF LFR RWS is always maintained.



A frame numbered SeqNum is accepted if LFR < SeqNum LAF, otherwise discarded. Frames that arrive out of order is buffered but not acknowledged.

If all preceding frames up to SeqNumToAck has arrived, then receiver acknowledges frame SeqNumToAck. The acknowledgement is cumulative. Variables updated are:

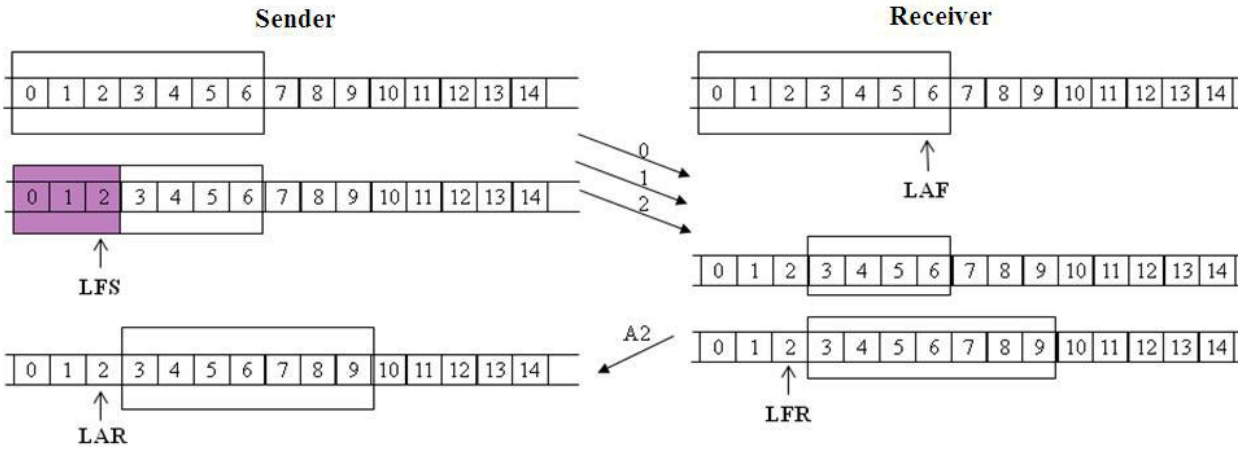
- LFR = SeqNumToAck
- LAF = LFR + RWS

Window size

SWS depends on how many frames are expected to be outstanding on the link. It is based on *delay* × *bandwidth* product.

RWS is same as SWS. Receiver buffers out-of-order frames, but does not acknowledge.

Example



Lost/Corrupt frames

When frames are lost or corrupt, there is less data in transit, since the sender cannot advance its window without an acknowledgement.

When an out-of-order frame arrives, receiver sends a negative acknowledgement (NAK) forcing the sender to retransmit the expected frame. This is known as *Selective Repeat*.

NAK speeds up retransmission of a frame before timer expires and improves performance.

Sequence Number

Sequence numbers are modulo 2^m where *m* is the size of sequence field and wrap around. To avoid the issue of identifying sequence numbers of different sets, SWS is defined as

SWS < (MaxSeqNum + 1) / 2

Advantages

It delivers frames *reliably* across an unreliable link using timeout and acknowledgement. It preserves the *order* in which frames are transmitted. The receiver ensures that it does

not pass a frame to the upper layer until all lower numbered frames are passed.

It supports *flow control*. The receiver through acknowledgement informs the sender about how many frames it can still receive.

6. Explain how framing is done using bit and byte oriented protocols.

Framing enables the message to reach the destination by adding physical address of sender and destination.

When a message is divided into smaller frames, error affects only that small frame. In *fixed*-size framing, there is no need for defining frame boundary.

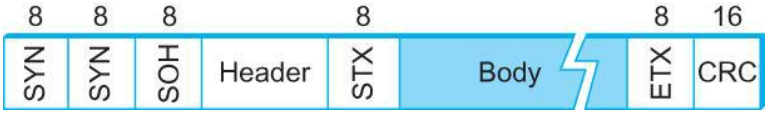
In *variable*-size framing, receiver should be able to determine where a frame starts/ends.

BYTE-ORIENTED PROTOCOLS

. The two different approaches are *sentinel* and the *byte-counting*.

Sentinel approach

Binary Synchronous Communication (BISYNC) protocol developed by IBM.



SYN special synchronization bits indicating beginning of the frame SOH special *sentinel* character that indicates start of header

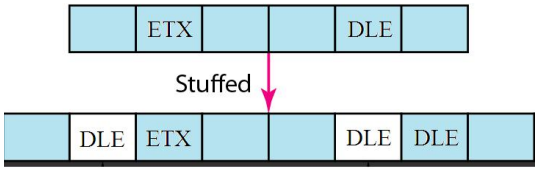
Header contains physical address of source, destination and other information STX special *sentinel* character that indicates start of text/body

ETX special *sentinel* character that indicates end of text/body CRC 16-bit CRC code used to detect transmission error

Character stuffing

The problem with *sentinel* approach, is that the ETX character might appear in the data. In such case, ETX is preceded with a DLE (data-link-escape) character. If the data portion contains escape character, then it is preceded by another DLE. The insertion of DLE character onto the data is known as *character stuffing*.

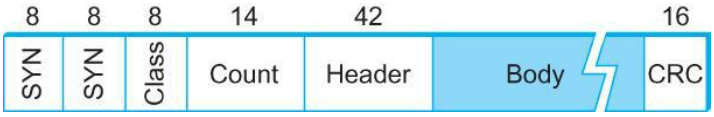
The receiver removes the additional escape characters and correctly interprets the frame. If ETX field is corrupted, then it is known as framing error. Such frames are discarded.



Byte-Counting Approach

An alternative to detect end-of-frame is to include number of bytes in the frame body as part of the frame header.

Digital Data Communication Message Protocol (DDCMP) uses the *count* approach.



The Count field specifies how many bytes are contained in the frame's body.

If Count field is corrupted, then it is known as framing error. The receiver comes to know of it when it comes across the SYN field of the next frame.

BIT-ORIENTED PROTOCOL

The bit-oriented protocols such as High-Level Data Link Control (HDLC) view the frame as a collection of bits. The frame format

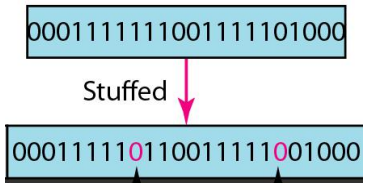
The beginning and end of a frame has a distinguished bit sequence 01111110 Sequence is also transmitted when link is idle for synchronization

Bit Stuffing

To prevent occurrence of bit pattern 01111110 as part of frame body, bit stuffing is used. In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added.

This extra stuffed bit is eventually removed from the data by the receiver.

The real flag 01111110 is not stuffed by the sender and is recognized by the receiver If a bit such as 01111111 arrives, then an error has occurred and the frame is discarded.

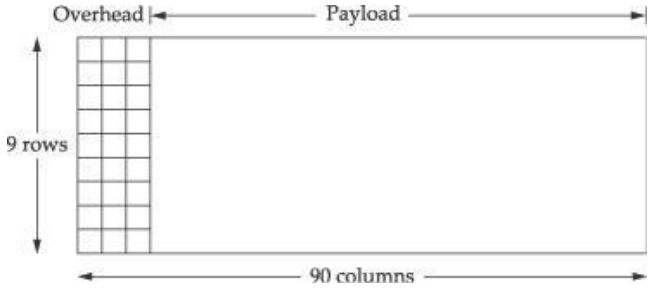


CLOCK-BASED FRAMING

Synchronous Optical Network (SONET) standard is clock-based framing of fixed size.

SONET runs on the carrier's optical network and offers rich set of services such as voice channel apart from data transfer.

Lowest speed SONET link STS-1 frame consist of 9 rows with 90 bytes each row.



First 2 bytes of the frame contain a special bit pattern indicating start of frame. First 3 bytes of each row are overhead and rest containing data.

Bit stuffing is not employed here

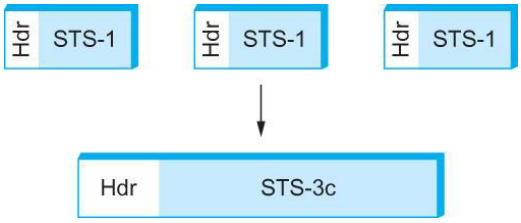
Receiver looks for the special bit pattern every 810 bytes. If not, the frame is discarded. Overhead bytes of a SONET frame are encoded using NRZ encoding. It allows the

receiver to recover sender's clock, the payload bytes are scrambled.

SONET supports the multiplexing of multiple low-speed links. The links range from 51.84 Mbps (STS-1) to 2488.32 Mbps (STS-48).

STS-1 frame is 810 bytes long with speed 51.84 Mbps, whereas STS-3 frame is 2430 bytes long at rate 155.52 Mbps.

STS-N signal can being used to multiplex N STS-1 frames. The payload from STS-1 frames are linked together to form a STS-N payload, denoted as STS-Nc.



7. Discuss the requirements for building a computer network. (April/May 17)

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*.

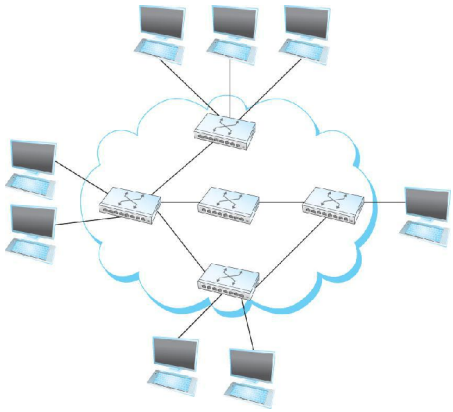
Switching could be either circuit or packet switching.

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*.

Each node on the network is assigned a unique address. A node can also send messages to a group of nodes (*multicasting*) or to all nodes on the network (*broadcasting*).



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.

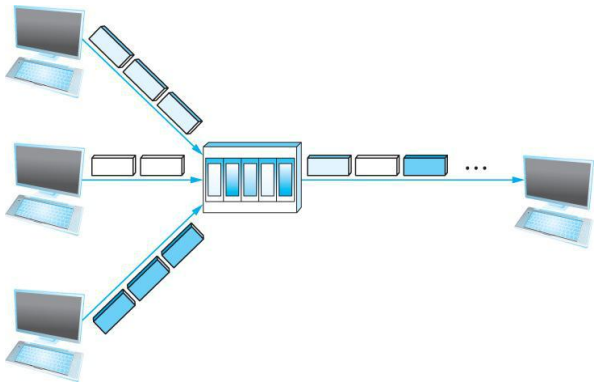
Synchronous time-division multiplexing (*STDM*) divides time into equal slots and flows use the slots in a round-robin manner.

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.

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 provided are *request/reply* and *message stream*.

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.

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.

Unit 2 – MEDIA ACCESS AND INTERNETWORKING

Part A

1. ***Define CSMA.(April/May 15)***
   - In Carrier Sense Multiple Access (CSMA), each station first checks state of the link using 1-persistent, non-persistent or p-persistent method.
   - Nodes using CSMA can distinguish between an idle and busy link.
   - If link is idle, then data is sent, else attempted after waiting random amount of time.
   - Two or more nodes may find the medium idle and transmit. Thus collision still exists.

2. ***What are the functions of Bridges (April/May 15)***
   - A bridge is a multi-input, multi-output node between two LANs that runs in *promiscuous* mode, accepts frames transmitted from either sides and forwards them to the other.
   - Bridge implements collision detection mechanism on all its interfaces.
   - LANs connected by one or more bridges is called *extended LAN.*

3. ***When ICMP indirect message used? (April/May 17)***

   The Internet Control Message Protocol (ICMP) identifies potentially weak and poorly protected networks. ICMP is a short messaging protocol that's used by systems administrators and end users for continuity testing of networks (e.g., using the *ping* or *traceroute* commands). Indirect ICMP query messages can be sent to the broadcast address of a given subnet (such as `192.168.0.255` in a `192.168.0.0/24` network). Operating systems respond in different ways to indirect queries issued to a broadcast address

4. ***What is exponential back-off? (Nov/Dec 16)***
   - When collision is detected, the node waits random amount of time and tries again.
   - Each time it fails to transmit, the adaptor doubles the amount of time it waits for each reattempt. This is known as exponential back-off.

5. ***What is Scatternet? (Nov/Dec 16)***

   A scatternet is a type of network that is formed between two or more Bluetooth-enabled devices, such as smartphones and newer home appliances. A scatternet is made up of at least two piconets.

6. ***What is hidden node problem? (May/Jun 16)***

   In wireless networking, the **hidden node problem** or **hidden terminal problem** occurs when a **node** is visible from a wireless access point (AP), but not from other **nodes** communicating with that AP. This leads to difficulties in media access control sublayer.

7. ***What is the need for ARP? (Nov/Dec 15)***

   Address Resolution Protocol is a function of the IP layer of the TCP/IP protocol stack. It is necessary to translate a hosts software address (IP address) to a hardware address (MAC address). Typically, a host uses ARP to determine the hardware address of another host. The system maintains a table that maps IP addresses to MAC addresses of different systems and routers on your network.

8. ***List the function of a repeater?***
   - A repeater is a device that connects LAN segments and extends length of the LAN.
   - It reconstructs a weak digital signal and forwards on all outgoing segments.

- Utmost four repeaters can be placed between a pair of hosts.
- It operates in the physical layer.

9. *What is a switch and its function?*

- A switch is a *multi-input*, *multi-output* device, receives packets on one of its links and transmits them on one or more other links. This is known as *switching* or *forwarding*
- Hosts are connected to the switch using point-to-point link (star topology).
- Large networks can be built by *interconnecting* a number of switches, i.e., scalable.
- Switching is either by datagram (connection-oriented) or virtual circuit (connection-less).

10. *What is the drawback of class-based addressing in IPv4?*

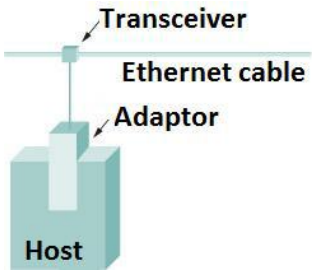
- In c1assful addressing, a large part of the available addresses were *wasted*, since Class A and B were too large for most organizations.
- Class C is suited only for *small* organization and therefore class B was opted.
- Reserved addresses were *sparingly* used.

Part B

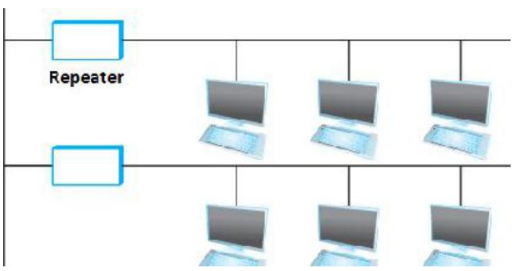
1. *Explain IEEE 802.3 standard or Ethernet in detail. (April/May 15, Nov/Dec 15, Nov/Dec 16)*

- Ethernet was developed by DEC, Intel and Xerox. It was standardized as IEEE 802.3
- Standard Ethernet is the most successful LAN technology with a data rate of 10 Mbps.
- It has evolved to Fast Ethernet (100 Mbps), Gigabit Ethernet (1 Gbps) and Ten-Gigabit Ethernet (10 Gbps).

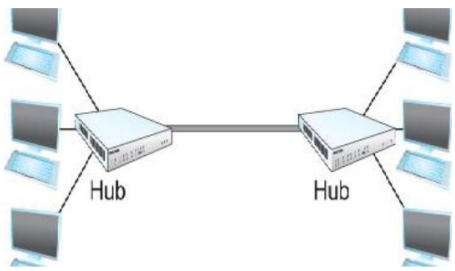
Physical Properties



- Hosts are tapped on to the Ethernet segment, each at least 2.5 m apart.
- Transceiver is responsible for transmitting/receiving frames and collision detection.
- Protocol logic is implemented in the adaptor.
- Ethernet can support a maximum of 1024 hosts.
- Maximum length of Ethernet is 2500 m.
- Manchester encoding scheme is used with digital signaling at 10 Mbps.
- Various forms of Standard Ethernet are 10Base5 (thick ethernet), 10Base2 (thin ethernet), 10Base-T (twisted-pair) and 10Base-F (fiber-optic).
- Ethernet segments can be connected using *repeater or a hub*.



Ethernet Repeater

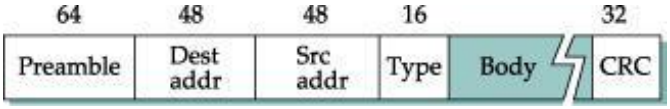


Ethernet Hub

Media Access Control (MAC)

Ethernet MAC protocol regulates access to the shared Ethernet link.

Frame Format



- *Preamble*—alternating 0s and 1s that alerts the receiving node.
- *Destination address*—physical address of the destination host.
- *Source address*—physical address of the sender.
- *Type*—contains either type of upper layer protocol or frame length.
- *Body*—data (46–1500 bytes).
- *CRC*—error detection information (CRC-32).

Addressing

- Each host on the Ethernet network has its own network interface card (NIC).
- NIC provides a globally unique 6-byte physical address (in *Hex* for readability).

06 : 01 : 02 : 01 : 2C : 4B

- If LSB of the first byte in a destination address is 0, then it is *unicast* else *multicast*.
- In *broadcast* address, all bits are 1s (FF:FF:FF:FF:FF:FF).

Transmitter

- Ethernet is a working example of CSMA/CD.
- Minimum frame length of 64 bytes is required for operation of CSMA/CD.
- Signals placed on the ethernet propagate in both directions and is broadcasted.
- Ethernet is a 1-persistent protocol. When there is a frame to be sent:
  - If link is idle, the frame is transmitted immediately.
  - If link is busy, it waits till it becomes idle and then transmits immediately.
- When two or more nodes transmit frame simultaneously, they collide. CSMA/CD works as follows:
  - Current transmission is aborted.
  - A 96-bit *runt* frame (64-bit preamble + 32-bit jamming sequence) is sent.
  - Other nodes refrain from transmission on receiving runt frame.
  - Retransmission is attempted after a back-off procedure ($k \times 51.2\mu$ s, k = 1,2,3,…). o After 16 attempts, retransmission is given up.

Receiver

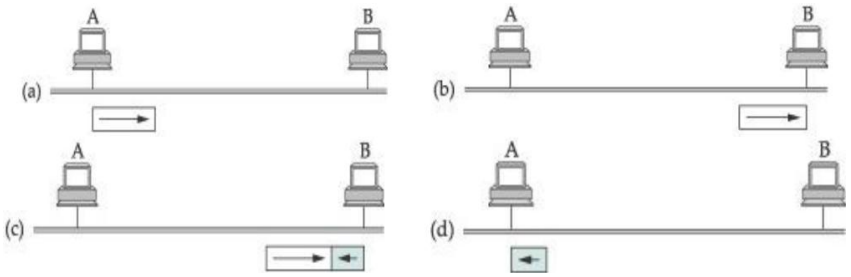
- Each frame transmitted on an Ethernet is received by every adaptor on that network.
- A frame is accepted if destination address:
  - matches *its* address,
  - contains *broadcast* address,
  - *multicast* address, if it's part of that multicast group.
- Frames are discarded, if it is not meant for that host.
- All frames are accepted, if configured in *promiscuous* mode.
- Ethernet does not acknowledge received frames.

Advantages and disadvantages of Ethernet.

- Easy to administer and maintain.
- Relataively inexpensive.
- Produces better output only when lightly loaded (< 200 hosts).
- It is an unreliable medium.

2. Why the minimum frame length in Ethernet should be at least 64 bytes (512 bits)?(May/Jun 14)

- Consider the following worst case scenario in which hosts *A* and *B* are at either ends.



- Host *A* begins transmitting a frame at time *t* (*fig a*).
- It takes link latency *d* for the frame to reach host *B*. Thus, the first bit of *A*'s frame arrives at *B* at time *t* + *d* (*fig b*)
- Suppose an instant before host *A*'s frame arrives, *B* senses it idle and begins to transmit.
- *B*'s frame collides with *A*'s frame, and this collision will be detected by host *B* (*fig c*)
- Host *B* aborts its transmission and sends a runt frame.
- Host *A* knows about collision only when *runt* frame reaches it, at time *t* + 2*d* (*fig d*)
- RTT for Ethernet with maximum distance (2500 m) is 51.2µ s. It corresponds to 512 bits (64 bytes) on 10 Mbps standard ethernet.
- Thus frame length of 512 bits is required for an host to detect collision before it transmits the last bit.

3. *Explain the functioning of wireless LAN or IEEE 802.11 in detail (April/May 15, Nov/dec15, April/May17)*

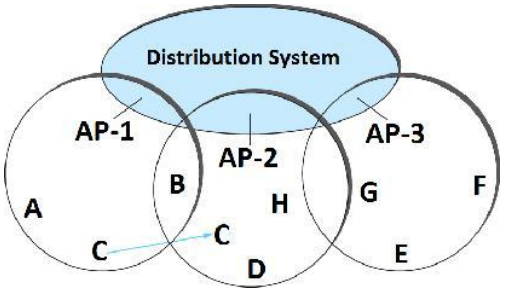
- Wireless LAN or WLAN or Wi-Fi is designed for use in a limited area (office, campus, building, etc). It is standardized as IEEE 802.11

Physical Properties

- WLAN runs over free space based on *FHSS* (frequency hopping over 79 1-MHz-wide frequency bandwidth) and DSSS (11-bit chipping sequence) with data rate of 2 Mbps.
- Variants of 802.11 are:
  - 802.11b operates in 2.4-GHz frequency band with data rate of 11 Mbps. 802.11a/g runs in 5-GHz band using orthogonal FDM (OFDM) at 54 Mbps
  - 802.11n uses multiple antennas (multiple input/output) and offers up to 100 Mbps
- Optimal bit rate for transmission is based on signal-to-noise ratio (SNR) in environment.

Distribution System

In wireless network, nodes are mobile and the set of reachable nodes change with time. Mobile nodes are connected to a wired network infrastructure called *access points* (AP) Access points are connected to each other by a *distribution system* (DS) such as Ethernet.



Nodes communicate directly with each other if they are reachable (eg, *A* and *C*) Communication between two nodes in different APs occurs via two APs (eg, *A* and *E*) Whenever a mobile node joins a network, it selects an AP. This is called *active scanning.* o Node sends a Probe frame.

- All APs within reach reply with a Probe Response frame.

o Node selects an AP and sends an Association Request frame. o Corresponding AP replies with an Association Response frame

Access points periodically send a Beacon frame advertising its features such as transmission rate. This is known as *passive scanning.*

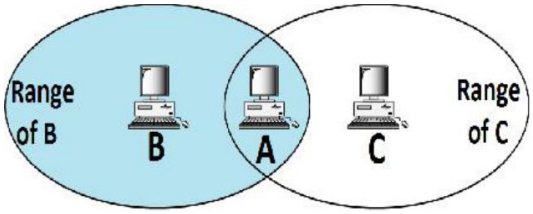
Hidden / Exposed Node Problem

All nodes are not within the reach of each other.

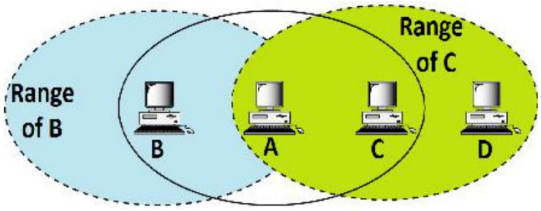
Carrier sensing may fail because of hidden node and exposed node problem.

Hidden Node

- Suppose node *B* is sending data to *A*. At the same time, node *C* also wishes to send to *A*.
- Since node *B* is not within the range of *C*, *C* finds the medium free and transmits to *A.*
- Frames from nodes *B* and *C* sent to *A* collide with each other.
- Thus nodes *B* and *C* are *hidden* from each other.



Hidden Node



Exposed Node

Exposed Node

p Suppose node *A* is transmitting to node *B* and node *C* has some data to be sent to node *D*.

q Node *C* finds the medium busy, since it hears the transmission from node *A* and refrains from sending to node *D*, even though its transmission to *D* would not interfere.

r Thus node *C* is *exposed* to transmission from node *A* to *B*

Multiple Access with Collision Avoidance (MACA)

Sender and receiver exchange *control frames* to reserve access, so that nearby nodes avoid transmission during duration of a data frame.

Control frames used to avoid collision are *Request to Send* (RTS) and *Clear to Send* (CTS).

Sender sends RTS frame to the receiver containing sender/receiver address and transmission duration.

Nodes that receive RTS frame are close to sender and wait for CTS to be transmitted back.

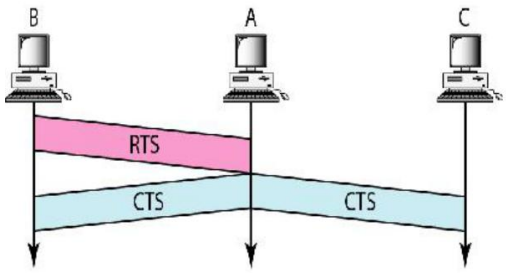
Receiver acknowledges and sends a CTS frame containing sender address and duration.

Nodes that receive CTS remain silent for the upcoming data transmission.

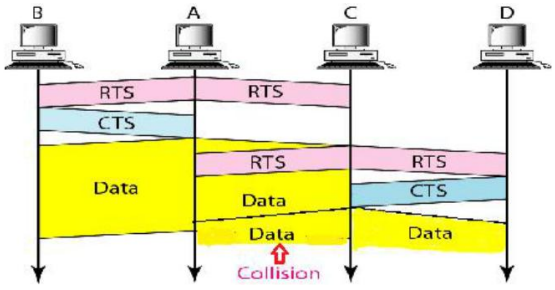
Nodes that receive RTS but not CTS, is away from the receiver and is free to transmit.

Receiver sends an ACK frame to the sender after successfully receiving data frames.

If RTS frames from two or more nodes collide, then they do not receive CTS. Each node waits for a random amount of time and then tries to send RTS again (back-off procedure).



Handshake for hidden node



Handshake for exposed node

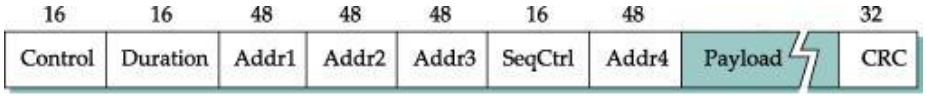
Handshake for Hidden node

Node *B* has frames for *A* and sends RTS to *A*. It reaches *A*, but not *C*.
Node *A* sends CTS frame to *B*, which is also received by node *C*.
Node *B* starts to transmit data frames to node *A*.
Node *C* knows of upcoming transmission from *B* to *A* and refrains from transmitting.

Handshake for Exposed node

- Assume that node *A* is transmitting to node *B* after exchanging control frames.
- Node *C* sends RTS to node *D* which is also sent to node *A*.
- Node *D* replies with CTS to *C*, whereas node *A* does not reply, since it is transmitting.
- Node *C* infers that there is no interference and transmits data frames to node *D*.

Frame Format



- *Control*—indicates frame type (RTS, CTS, ACK or data) and 1-bit ToDS / FromDS
- *Duration*—specifies duration of frame transmission.

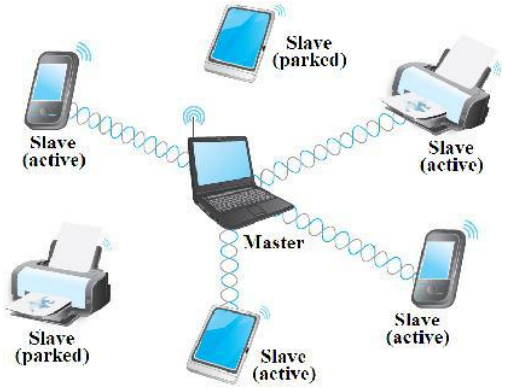
ToDS	FromDS	Addr1	Addr2	Addr3	Addr4	*Description*
0	0	Destination	Source			Sent directly
0	1	Destination	Sending AP	Source		Frame is coming from a distribution system
1	0	Receiving AP	Source	Destination		Frame is going to a distribution system
1	1	Receiving AP	Sending AP	Destination	Source	Frame is going from one AP to another AP

- *Addresses*—The *four* address fields depend on value of ToDS and FromDS subfields.
- *Sequence Control*—defines sequence number of the frame.
- *Payload*—contains a maximum of 0–2312 bytes.
- *CRC*—contains CRC-32 error detection sequence.

4. Write short notes on Bluetooth (May/Jun 16)

- Bluetooth technology, standardized as IEEE 802.15.1 is a personal area network (PAN).
- It is used for short-range wireless communication (maximum 10 m) between mobile phones, PDAs, notebook and other peripheral devices.
- Uses low power transmission, operates in 2.45 GHz band with data rate up to 3 Mbps.
- Bluetooth Special Interest Group has specified a set of protocols for a range of application, known as *profiles*. For instance, a profile synchronizes PDA and PC.
- Bluetooth network configuration is known as *piconet*. A piconet can have up to eight stations, one of which is called the master and the rest are called slaves.
  - o Slaves do not directly communicate with each other, but via the master.
  - o Bluetooth uses FHSS (79 channels, each 625 μs) for transmission.
  - o Master transmits in odd-numbered slots, whereas slave respond in even slots.
  - o Slaves in *parked* or inactive state cannot communicate, until it is activated by the master. Maximum of 255 devices can be in parked state.

- Bluetooth hardware and software is simpler and cheaper.



5. List and compare the features of any two wireless technologies. (May/Jun 16)

	Bluetooth	**WiFi**	**WiMax**	**3G**
IEEE standard	802.15.1	802.11	802.16	
Link length	10 m	100 m	10 km	Tens of km
Bandwidth	2.1 Mbps (shared)	54 Mbps (shared)	70 Mbps	384 Kbps
Usage	Link a peripheral to a computer	Link a computer to a wired base	Link a building to a wired tower	Link a cell phone to a wired tower

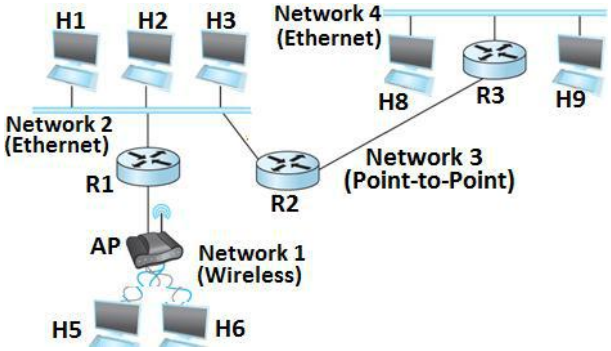
6. Discuss Internetworking Protocol in detail.(Nov/Dec 16)

♏◑ *Internet Protocol* (IP) is used to build scalable, heterogeneous internetworks.

♐◑ Ability of IP to run over *any* networking technology is its strength.

♑◑ IP *Service* model has two parts

- *Datagram* (connectionless) model of data delivery
- *Addressing* scheme to identify all hosts uniquely in the internetwork.

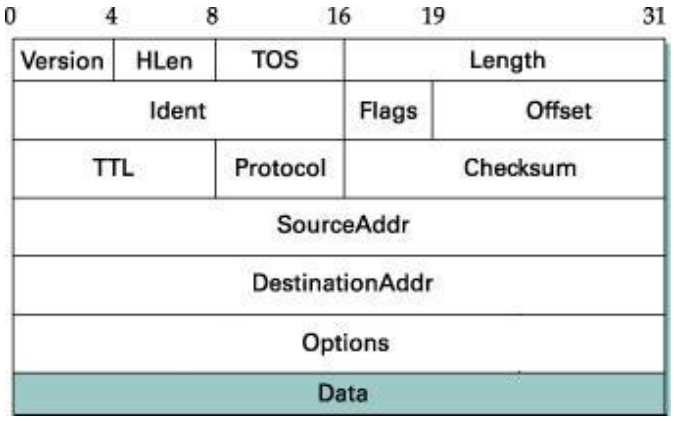


Datagram Delivery

- Best-effort, connectionless service is used by IP to deliver a datagram
- Packets can be lost or corrupted. It can also be delivered out of order.
- IP provides *neither* error control *nor* flow control. It is an *unreliable* service.

Packet Format

- IPv4 datagram is a variable-length packet consisting of two parts, *header* and *data*.
- Header is 20–60 bytes long and contains information essential to routing and delivery
- Minimum packet length is 20 bytes and maximum 65,535 bytes.



- Version—version of IPv4 protocol, i.e. 4.
- HLen—length of the header in 4-byte words. When there are no options, its value is 5.
- TOS—allows packets to be placed on separate queues based on QoS required.
- Length—total packet length (header + data), which is restricted to 65,535 bytes.
- Ident—a 16-bit identifier that uniquely identifies a datagram packet.
- Flags—3-bit field contains D (*do not fragment*) bit and M (*more fragment*) bit.
- Offset—shows relative position of the fragment in units of 8 bytes.
- TTL—defines lifetime of the datagram (default 64 hops).
- Protocol—specifies upper layer protocol (e.g., 6 for TCP, 17 for UDP).
- Checksum—16-bit internet checksum for the packet header.
- SourceAddr / DestinationAddr—32-bit IP address of source and destination host.

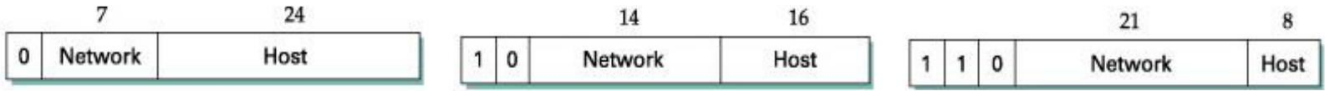
Global Addressing

- IPv4 uses 32-bit addresses, i.e., approximately 4 billion addresses (2^{32}).
- IP addresses are *hierarchical*, i.e., it corresponds to hierarchy in the internetwork.
- IP addresses consist of two parts, *network id* and *host id*.
  - Network id identifies *physical* network to which the host is attached.
  - Hosts attached to a network have the same network id in their IP address.
  - Host id is used to *uniquely* identify a host on that network.
- Router that connects networks has a unique IP address on each of its *interface*.
- It is written as four octets (0–255) in dotted decimal notation. For eg, 172.16.15.161
- IPv4 address space is divided into five classes: A, B, C, D and E.
- Class of an IP address is identified by MSBs (in binary) or first byte (in decimal).

Class	Binary	Decimal	Addressing	Application
A	0	0–127	Unicast	WAN
B	10	128–191	Unicast	Campus Network
C	110	192–223	Unicast	LAN
D	1110	224–239	Multicast	
E	1111	240–255	Reserved	

- Classes A, B and C are used for *unicast* addressing.
- Class D was designed for *multicasting* and class E is *reserved*.
- Classes A, B, C have certain bits for network part and rest for host part i.e., networks belonging to a class and number of hosts attached to it are *limited*
  - Class A—upto 126 networks and 16 million hosts per network
  - Class B—upto 2^{14} networks and 65,534 hosts per network

- Class C—upto 2^{21} networks and 254 hosts per network



Class A *Class B* *Class C*

Datagram Forwarding

- Destination address is used by routers to forward packets in a *connectionless* manner.
- *Forwarding* table at a router is a list of (NetworkNum, NextHop) pairs.

Algorithm

```
if (NetworkNum of destination = NetworkNum of any of its interface)
    then Deliver packet to destination over that interface
else
    if (NetworkNum of destination is in forwarding table) then
        Deliver packet to NextHop router
    else
        Deliver packet to default router
```

Example

Suppose *H5* sends a datagram to *H8*, then forwarding is as follows:
H1 sends datagram to its *default* router *R1*, since it cannot deliver directly.
R1 sends datagram over wireless network to its *default* router *R2*, since *H8* network id does not match any of its interfaces.
R2 forwards the datagram to *R3* based on its *forwarding* table.
R3 forwards the datagram *directly* to *H8*, since both are on the same network.

NetworkNum	NextHop
1	R1
	Interface 1
	Interface 0
4	R3

R2 Forwarding Table

7. Write short notes on CIDR or Superneting.

- *Subnetting* does not prevent an organization opting for Class B. Address efficiency for Class B can be as low as 0.39% (256 / 65535).
- If Class C addresses were given instead of Class B, then routing tables gets larger.
- Classless Interdomain Routing (*CIDR*) tries to balance between minimize the number of routing table entries and handling addresses space efficiently.
- CIDR *aggregates* routes, by which an entry in forwarding table is used to reach multiple networks. It collapses multiple addresses into a single supernet, i.e., *supernetting*.

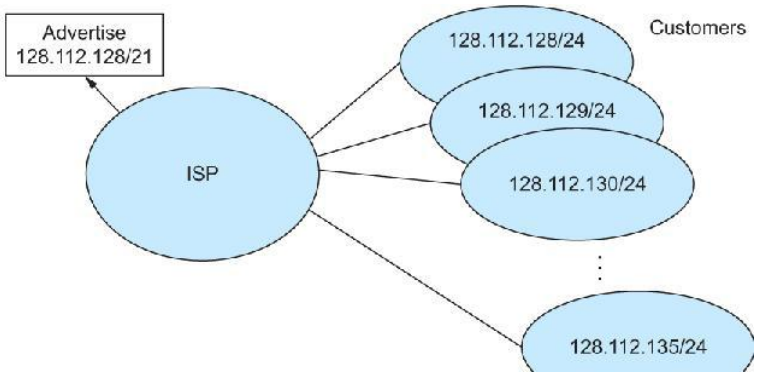
Example

- Consider an organization with 16 Class C networks.
- Instead of providing 16 addresses at random, a block of *contiguous* Class C address is given. For example, from 192.4.16 to 192.4.31
- Bitwise analysis show 20 MSBs (11000000 00000100 0001) are same. Thus a 20-bit *network number* is created, i.e., range *between* Class B and C network.
- Thus higher address *efficiency* is achieved by providing small chunks of address, smaller than Class B network. Thus a single network prefix is used in forwarding table.

Restrictions

- CIDR uses a new type of notation to represent network numbers or prefixes.
- It is represented as /X, where X is the prefix length in bits. For example, 192.4.16/20
- Addresses in a block must be *contiguous* and number of addresses must be *powers* of 2.

Example

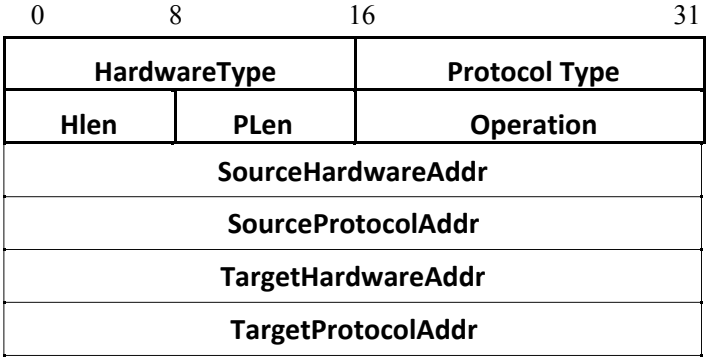


- When different customers are connected to a service provider, prefixes can be assigned such that they share a common, further aggregation can be achieved.
- Consider an ISP providing internet connectivity to 8 customers. All customer prefix starts with the same 21 bits.
- Since all customers are reachable through the same provider network, a single route is advertised by ISP with common 21-bit prefix that all customers share.

8. Detail the process of determining the physical address of a destination host (ARP).

- Physical interface on a host or router understands physical addressing scheme of that network only. Therefore IP address has to be translated to link-level address.
- To send datagram to a host or router, *both* logical and physical address must be known.
- *Address Resolution Protocol* (ARP) enables a source host to know the physical address of another node when the logical address is known.
- ARP relies on *broadcast* support from physical networks such as ethernet, token ring, etc.
- ARP enables each host to build a *mapping* table between IP address and physical address.

Packet Format



- HardwareType—type of the physical network (e.g., 1 for *ethernet*).
- ProtocolType—value of upper-layer protocol (e.g., 8 for *IPv4*).
- HLen—length of the physical address in bytes (e.g., 6 for *Ethernet* address).
- PLen—length of the logical address in bytes (e.g., 4 for *IPv4* address).

- Operation—type of ARP (1 for *request*, 2 for *reply*).
- SourceHardwareAddr— physical address of the sender.
- SourceProtocolAddr—logical address of the sender.
- TargetHardwareAddr—physical address of the target node.
- TargetProtocolAddr—logical address of the target node.

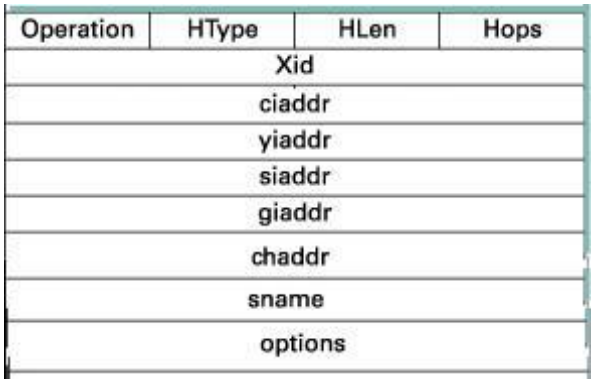
Address Translation

- Host *checks* its ARP table with destination IP address.
  - If an entry exists, then corresponding physical address is used to send datagram.
  - Otherwise, source host finds physical address using ARP.
- Source host creates a *ARP request* packet with:
  - Operation field set to 1.
  - Target Physical address field is unknown and *filled* with 0s (broadcast address).
- ARP request is encapsulated in IP packet and *broadcasted* over the physical network.
- Each host takes note of sender's logical and physical address. All nodes except the destination host *discard* the packet.
- Destination host constructs an *ARP reply* packet with Operation field set to 2.
- ARP reply is *unicast* and sent back to the sender.
- Sender *stores* target logical-physical address pair in its ARP table from reply packet.

9. Discuss the automatic configuration of IP address to hosts using DHCP.

- Operating systems allow system administrator to *manually* configure IP address, which is tedious and *error-prone*.
- Dynamic Host Configuration Protocol (DHCP) enables *auto* configuration of IP address to hosts using DHCP server.
- DHCP server sends and receives message using UDP over ports 67 and 68 respectively.
- DHCP provides *dynamic* (automatic) address allocation when host connects to a network.

Packet Format



- Operation—specifies type of DHCP packet.
- Xid—specifies the transaction id.
- ciaddr—specifies client IP address in case of DHCPREQUEST
- yiaddr— known as *your IP address*, filled by DHCP server.
- siaddr—contains IP address of the DHCP server.
- giaddr—contains IP address of the Gateway or relay agent.
- chaddr—contains hardware (physical) address of the client.
- options—contains information such as lease duration, default route, DNS server, etc.

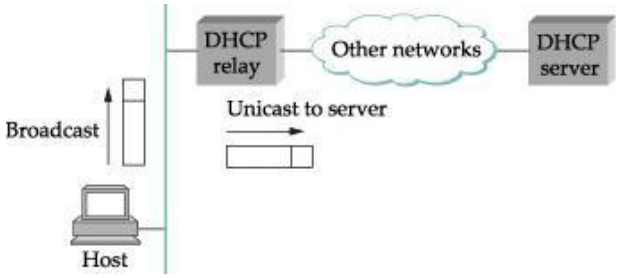
Dynamic Address Allocation

- DHCP server is configured with range of addresses to be assigned to hosts on demand.

- To contact DHCP server, client *broadcasts* a DHCPDISCOVER message with IP address 255.255.255.255 and it's physical address placed in chaddr field
- DHCP server selects an *unassigned* IP address for yiaddr field and adds an entry to *dynamic* database along with client's physical address.
- DHCP server sends DHCPOFFER message containing client's IP and physical address, server IP address and options.
- Client *sends* a DHCPREQUEST message, requesting the offered address.
- Based on transaction id, the DHCP server *acknowledges* with a DHCPACK message.
- When lease period expires, client attempts to *renew*. It's up to server to accept or reject it.

DHCP relay

- DHCP is an *application* layer protocol, i.e., server/client need not be on the same network



- In such case, *DHCP relay* agent receives broadcast message.
    - DHCP relay stores relay address in giaddr before sending it to DHCP server. ○ DHCP server response is sent to relay agent, which is sent back to the client.

10. Write short notes on error reporting using ICMP.

- Internet Control Message Protocol (ICMP) is used to report *error messages* to source host and *diagnose* network problems. ICMP message is *encapsulated* within an IP packet.

Error reporting

- *Destination Unreachable*—When a router *cannot route* a datagram, the datagram is discarded and sends a destination unreachable message to source host.
- *Source Quench*—When a router or host discards a datagram due to *congestion*, it sends a source-quench message to the source host. This message acts as flow control.
- *Time Exceeded*—Router discards a datagram when TTL field becomes 0 and a time-exceeded message is sent to the source host.
- *Parameter Problem*—If a router discovers ambiguous or *missing* value in any field of the datagram, it discards the datagram and sends parameter problem message to source.
- *Redirection*—Redirect messages are sent by the default router to inform the source host to *update* its forwarding table when the packet is routed on a wrong path.

Query Messages

- *Echo Request & Reply*—The combination of echo-request and echo-reply messages determines whether two systems can *communicate* at the IP level.
- *Timestamp Request & Reply*—Two machines can use the timestamp request and timestamp reply messages to determine the *round-trip time* (RTT).
- *Address Mask Request & Reply*—A host to obtain its subnet mask, sends an address mask request message to the router, which responds with an address mask reply message.
- *Router Advertisement*—A host broadcasts a *router solicitation* message to know about the router. Router broadcasts its routing information with *router advertisement* message.

UNIT 3- ROUTING

Part A

1. State the difference between router and bridge. (April/May 15)

router	bridge
It uses software configured network address(ip address) to determine the address.	It determines the destination address with the help of MAC address(i.e. ethernet address) of the device.
The router devices use the routing tables to route the data to the destination.	The bridge devices do not use any other device to transfer the data to the destination.
Router devices communicate with other routers to decide(select) the best path to transfer the data.	With the help of MAC addresses of the devices connected in the network, the bridges listen to the network traffic and then decide the best path to send the data.
Router devices are used to connect the LAN and WAN links.	Bridge devices are used to connect the LAN links.
Routing Protocol examples: TCP/IP, IPX/SPX, Apple Talk	Types of bridges include transparent bridge and translational bridge.

2. What are the metrics used in Routing protocols? (April/May 15)

- Parameters used to calculate link costs is known as metric or routing metric.
- One way is to assign uniform cost to all links (say 1 hops). The pros and cons are:
  - Easy to calculate least-cost route.
  - *Latency* on a link is not considered. For example, links with different latency say *250 ms* and *1 ms* are not distinguished.
  - *Bandwidth* of the link is not considered. For example, links with different capacity such as *10 Kbps* and *45 Mbps* are treated in a similar manner.
  - *Current load* is not considered, i.e., routing around overloaded links is impossible

3. Write about unicast, multicast and broadcast routing. (April/May 17)

Unicast: traffic, many streams of IP packets that move across networks flow from a single point, such as a website server, to a single endpoint such as a client PC. This is the most common form of information transference on networks.

Broadcast: Here, traffic streams from a single point to all possible endpoints within reach on the network, which is generally a LAN. This is the easiest technique to ensure traffic reaches to its destinations.

This mode is mainly utilized by television networks for video and audio distribution.

Multicast: In this method traffic recline between the boundaries of unicast (one point to one destination) and broadcast (one point to all destinations). And multicast is a "one source to many destinations" way of traffic distribution, means that only the destinations that openly point to their requisite to accept the data from a specific source to receive the traffic stream.

4. Why we need to change from IPv4 to IPv6?(April/May 17)

The IPv6 has following advantages over IPv4

- *Address space*—IPv6 uses 128-bit address whereas IPv4 uses 32-bit address. Hence IPv6 has huge address space whereas IPv4 faces address shortage problem.
- *Header format*—Unlike IPv4, optional headers are separated from base header in IPv6. Each router thus need not process unwanted addition information.
- *Extensible*—Unassigned IPv6 addresses can accommodate needs of future technologies.

5. **What is VCI? (Nov/Dec 16)**
   - *Virtual Circuit Identifier* (VCI) uniquely identifies a connection. It has *link local scope*.
   - Incoming and outgoing VCI is always distinct.
   - VCI and interface on which it was received, uniquely identifies a virtual connection.
   - Connection state set by the administrator is known as Permanent virtual circuit (PVC).
   - Hosts can set virtual circuit through signalling (SVC). It consist of two phases: Setup Request and Acknowledgement

6. **What is Fragmentation and Reassembly? (Nov/Dec 16)**
   Fragmentation is the process of chopping larger chunks of data into smaller chunks. Fragmentation is usually performed at the hardware level, and when data is chopped into fragments, it is referred to as a *frame*. Fragmentation occurrs so that data can be transmitted across a connection without overwhelming the memory buffers on either side of the connection. Fragmentation allows for the coordination of data transmission amongst devices connected to a common transmission medium.
   Reassembly is the reverse of segmentation. Protocol Data Units are put back together in the correct order to reassemble a stream of data in its original form.

7. **Mention some connecting devices in internetworking. (May/Jun 16)**
   An internetworking device is a widely-used term for any hardware within networks that connect different network resources. Key devices that comprise a network are routers, bridges, repeaters and gateways.

8. **Identify the class of IP addresses.(Nov/Dec 15)**

Historical classful network architecture

Class	Leading bits	Size of *network number* bit field	Size of *rest* bit field	Number of networks	Addresses per network	Start address	End address
A	0	8	24	128 (2^7)	16,777,216 (2^{24})	0.0.0.0	127.255.255.255
B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	128.0.0.0	191.255.255.255
C	110	24	8	2,097,152 (2^{21})	256 (2^8)	192.0.0.0	223.255.255.255

9. **Define routing. (Nov/Dec 15)**
   In internetworking, the process of moving a packet of data from source to destination. Routing is usually performed by a dedicated device called a router. Routing is a key feature of the Internet because it enables messages to pass from one computer to another and eventually reach the target machine. Each intermediary computer performs routing by passing along the message to the next computer. Part of this process involves analyzing a *routing table*to determine the best path.

10. **How does a router differ from a switch?**
    - Control processor executes routing protocols and acts as central point of control.
    - Routers are designed to handle variable-length packets, whereas switch is cell-based, i.e., fixed length.
    - Routers throughput is hard to characterize than a switch and is given by its linerate.

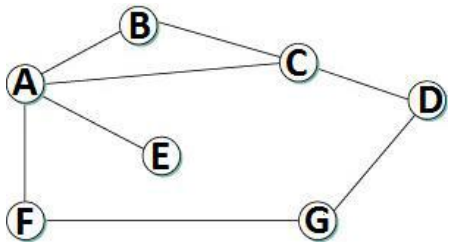
      packetsize × pps = linerate
    - IP forwarding algorithm is complex than a simple lookup done in a switch.

- Routers can be classified as centralized or distributed forwarding model. In centralized, one processor handles all ports, whereas multiple processors exist in distributed model
- Routers include network processor, optimized for networking tasks such as performing lookups, calculating checksums, etc.

Part B

1. ***Explain distance vector routing (or) routing information protocol (or) bellman-ford algorithm (Nov/Dec 15, May/Jun 16, April/May 15)***
   - Distance vector routing is *distributed*, i.e., algorithm is run on all nodes.
   - Each node *knows* the distance (cost) to each of its directly connected neighbors.
   - Nodes construct a *vector* (Destination, Cost, NextHop) and distributes to its neighbors.
   - Nodes compute routing table of *minimum* distance to every other node via NextHop using information obtained from its neighbors.

Initial State



- In given network, *cost* of each link is 1 hop.
- Each node sets a distance of 1 (hop) to its *immediate* neighbor and cost to itself as 0.
- Distance for non-neighbors is marked as *unreachable* with value ∞ (infinity).
- For node *A*, nodes *B*, *C*, *E* and *F* are *reachable*, whereas nodes *D* and *G* are *unreachable*.

Destination	Cost	NextHop
A	0	A
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

Node A's initial table

Destination	Cost	NextHop
A	1	A
B	1	B
C	0	C
D	1	D
E	∞	—
F	∞	—
G	∞	—

Node C's initial table

Destination	Cost	NextHop
A	1	A
B	∞	—
C	∞	—
D	∞	—
E	∞	—
F	0	F
G	1	G

Node F's initial table

Sharing & Updation

- Each node *sends* its initial table (distance vector) to neighbors and receives their estimate.
- Node *A* sends its table to nodes *B*, *C*, *E* & *F* and receives tables from nodes *B*, *C*, *E* & *F*.

- Each node *updates* its routing table by comparing with each of its neighbor's table
- For each destination, Total Cost is computed as:

  Total Cost = Cost (*Node* to *Neighbor*) + Cost (*Neighbor* to *Destination*)
- If Total Cost < Cost then

  Cost = Total Cost and NextHop = *Neighbor*
- Node *A learns* from *C*'s table to reach node *D* and from *F*'s table to reach node *G*. ○ Total Cost to reach node *D* via *C* = Cost (*A* to *C*) + Cost(*C* to *D*) = 1 + 1 = 2.
  - Since 2 < ∞, entry for destination *D* in *A*'s table is changed to (*D*, 2, *C*) ○ Total Cost to reach node *G* via *F* = Cost(*A* to *F*) + Cost(*F* to *G*) = 1 + 1 = 2
  - Since 2 < ∞, entry for destination *G* in *A*'s table is changed to (*G*, 2, *F*)
- Each node builds *complete* routing table after few exchanges amongst its neighbors.

Destination	Cost	NextHop
A	0	A
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

Node A's final routing table

- System stabilizes when all nodes have complete routing information, i.e., *convergence*.
- Routing tables are exchanged *periodically* (every 30 sec.) and in case of *triggered* update.

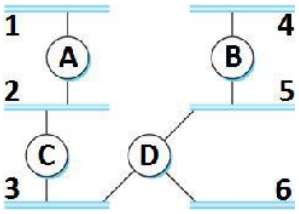
Triggered Update

- Link failure is *assumed*, if a node does not receive periodic updates from a neighbor.
- *Infinite* cost is assigned to that neighbor and immediately *shares* with other neighbors.
- Neighbors *update* their neighbors and so on. This is known as *triggered* update.
- Assume that node *F* detects that its link to *G* has *failed*.
  - Node *F sets* distance to *G* as ∞ and shares its table with *A*.
  - Node *A updates* its distance to *G* as ∞.
  - Meanwhile, node *A* receives *periodic* update from *C* with distance to *G* as 2 hops. o Node *A updates* its distance to *G* as 3 hops via *C* and shares it with *F*.
  - Eventually node *F* is updated to reach *G* via *A* in 4 hops.

  Network stabilizes after few updates, when an alternate path is found.

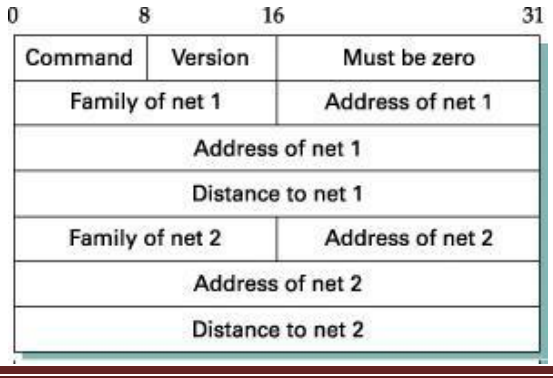
Routing Information Protocol (RIP)

- RIP is an intra-domain routing protocol based on distance-vector algorithm.



Example Network

- Routers *advertise* the cost of reaching networks. Cost of reaching each link is 1 hop. For example, router *C* advertises to *A* that it can reach network 2, *3* at cost 0 (directly connected), networks *5*, *6* at cost 1 and network *4* at cost 2.
- Each router *updates* cost and next hop for each network number.
- Infinity is defined as 16, i.e., any route cannot have more than 15 hops. Therefore RIP can be implemented on small-sized networks only.
- Advertisements are sent every 30 seconds or in case of triggered update.
- RIP packet format (version 2) contains (*network address*, *distance*) pairs.



What is count-to-infinity or loop instability problem? List the solutions.

- Suppose link from node *A* to *E* goes *down*.
  - Node *A* advertises a distance of ∞ to *E* to its neighbors.
    Node B receives periodic update from C before A's update reaches B,
  - Node *B* updated by *C*, concludes that *E* can be reached in 3 hops via *C*.
  - Node *B* advertises to *A* as 3 hops to reach *E*
  - Node *A* in turn updates *C* with a distance of 4 hops to *E* and so on.
- Thus nodes update each other until cost to *E* reaches *infinity*, i.e., *no convergence*.
- Routing table does not stabilize. This problem is called *loop instability* or *count to infinity*

Solutions

- *Infinity* is redefined to a small number, say 16. Distance between any two nodes can be 15 hops maximum. Thus distance vector routing *cannot be used* in large networks.
- When a node updates its neighbors, it does not send those routes it learned from each neighbor back to that neighbor. This is known as *split horizon*.
- *Split horizon with poison reverse* allows nodes to advertise routes it learnt from a node back to that node, but with a warning message.

2. Explain link state routing (or) OSPF protocol (or) shortest path algorithm with an example.(April/may 15, April/May 17, Nov/Dec 16)

- Each node knows *state* of link to its neighbors and *cost*.
- Nodes create an update packet called *link-state packet* (LSP) that contains:
  - ID of the node
  - List of neighbors for that node and associated cost
  - 64-bit Sequence number
  - Time to live
- Link-state routing protocols rely on two mechanisms:
  - Reliable *dissemination* of link-state information to all other nodes
  - Route *calculation* from the accumulated link-state knowledge

Reliable Flooding

- Each node *sends* its LSP out on each of its directly connected links.
- When a node receives LSP of another node, checks if it has an LSP already for that node.
  - If not, it stores and forwards the LSP on all other links except the incoming one.
  - Else if the received LSP has a *bigger* sequence number, then it is stored and forwarded. Older LSP for that node is *discarded*.
  - Otherwise discard the received LSP, since it is not latest for that node.
- Thus recent LSP of a node eventually *reaches* all nodes, i.e., reliable *flooding*.

(a) (b) (c) (d)

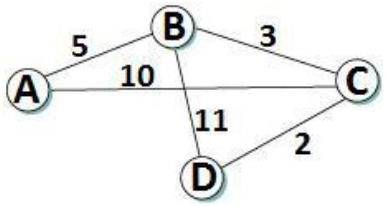
- Flooding of LSP in a small network is as follows:
  When node *X* receives *Y*'s LSP (*fig a*), it floods onto its neighbors *A* and *C* (*fig b*)
  - Nodes *A* and *C* forward it to *B*, but does not sends it back to *X* (*fig c*).
  - Node *B* receives two copies of LSP with same sequence number.
  Accepts one LSP and forwards it to *D* (*fig d*). Flooding is complete.
- LSP is generated either *periodically* or when there is a *change* in the topology.

Route Calculation

- Each node knows the entire topology, once it has LSP from every other node.
- Forward search algorithm is used to compute routing table from the received LSPs.
- Each node maintains two lists, namely Tentative and Confirmed with entries of the form (Destination, Cost, NextHop).

Forward Search algorithm (Djkstra's Shortest Path)

- Initialize the Confirmed list with an entry for the Node (Cost = 0).
- Node just added to Confirmed list is called Next. Its LSP is examined.
- For each neighbor of Next, calculate cost to reach each neighbor as Cost (Node to Next) + Cost (Next to Neighbor).
    - If Neighbor is neither in Confirmed nor in Tentative list, then add (Neighbor, Cost, NextHop) to Tentative list.
    - If Neighbor is in Tentative list, and Cost is less than existing cost, then replace the entry with (Neighbor, Cost, NextHop).
- If Tentative list is empty then *Stop*, otherwise move *least* cost entry from Tentative list to Confirmed list. Go to *Step 2*.



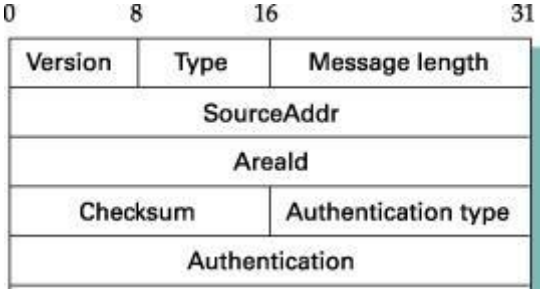
- For the given network, the process of building routing table for node *D* is tabulated

Step	Confirmed	Tentative	Comment
1	(D, 0, –)		*D* is moved to Confirmed list initially

Step	Confirmed	Tentative	Comment
2	(D, 0, –)	(B, 11, B) (C, 2, C)	Based on *D*'s LSP, its immediate neighbors *B* and *C* are added to Tentative list
3	(D, 0, –) (C, 2, C)	(B, 11, B)	Lowest cost entry *C* in Tentative list is moved to Confirmed list. *C*'s LSP is to be examined next.
4	(D, 0, –) (C, 2, C)	(B, 5, C) (A, 12, C)	Cost to reach *B* through *C* is 5, so the entry (*B*, 11, *B*) is replaced. *C*'s neighbor *A* is also added to Tentative list
5	(D, 0, –) (C, 2, C) (B, 5, C)	(A, 12, C)	Lowest cost entry *B* is moved to Confirmed list. *B*'s LSP is examined next.
6	(D, 0, –) (C, 2, C) (B, 5, C)	(A, 10, C)	Since *A* could be reached through *B* at a lower cost than the existing one, the Tentative list entry (*A*, 12, *C*) is replaced to (*A*, 10, *C*)
7	(D, 0, –) (C, 2, C) (B, 5, C) (A, 10, C)		Only member *A* is moved to Confirmed list. Process completed.

Open Shortest Path First Protocol (OSPF)

- OSPF is a non-proprietary widely used link-state routing protocol. Features added are:
- *Authentication*—Malicious host can collapse a network by advertising to reach every host with cost 0. Such disasters are averted by authenticating routing updates.
- *Additional hierarchy*—Domain is partitioned into areas, i.e., OSPF is more scalable.
- *Load balancing*—Multiple routes to the same place are assigned same cost. Thus traffic is distributed evenly.



- Version—represents the current version, i.e., 2.
- Type—represents the type (1–5) of OSPF message.
- SourceAddr—identifies the sender
- AreaId—32-bit identifier of the area in which the node is located
- Checksum—16-bit internet checksum
- Authentication type—1 (simple password), 2 (cryptographic authentication).
- Authentication—contains password or cryptographic checksum

3. Explain distance vector multicast routing protocol.(Nov/Dec 15, Nov/Dec 16)

- Multicast forwarding table is a tree structure, known as *multicast distribution trees*.
- Internet multicast is implemented on physical networks that support broadcasting by *extending* forwarding functions.
- Major multicast routing protocols are:
  - Distance-Vector Multicast Routing Protocol (DVMRP)
  - Protocol Independent Multicast (PIM)

Distance vector routing for unicast is extended to support multicast routing.
Each router maintains (Destination, Cost, NextHop) for all destination through exchange of distance vectors.

Multicasting is added to distance-vector routing in two stages.

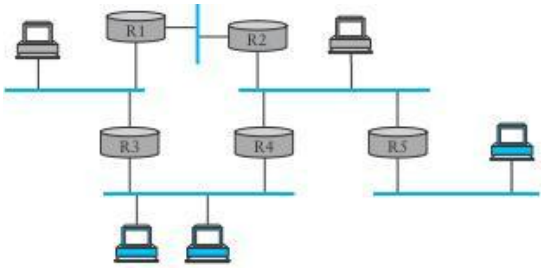
- Reverse Path Broadcast *floods* packets to all networks
- Reverse Path Multicasting *prunes* end networks that do not have hosts belonging to a multicast group.
- DVMRP is also known as *flood-and-prune* protocol.

Reverse-Path Broadcasting

- Router on receiving a multicast packet from source *S* to a Destination from NextHop, *forwards* the packet on all out-going links, since it comes from shortest path.
- Packet is flooded but not looped back to *S*. The drawbacks are:
  - h) It floods a network, even if it has *no members* for that group.
    Packets are forwarded by each router connected to a LAN, i.e., *duplicate flooding*
- Duplicate flooding is avoided by
  Router that has the *shortest* path to source *S*, is selected as parent router.
  Only parent router *forwards* multicast packets from source *S* to that LAN.
- Thus shortest path to source (*reverse*) is considered for forwarding decisions..

Reverse-Path Multicasting

- Multicasting is achieved by pruning networks that do not have members for a group *G*.
- Step 1: Identify a *leaf* network which has only one router (parent).
  - Leaf network is *monitored* to determine if it has any members for group *G,* by having hosts periodically announce to which group it belongs to.
    Router thus decides whether or not to forward group *G* packets over that LAN.
- Step 2: Propagate "*no members of G here*" up the shortest path tree.
  Routers augments (Destination, Cost) pairs with set of groups for which the leaf network is interested in receiving multicast packets.
  Information is propagated amongst routers so that a router knows for what groups it should forward on each of its links.
- Including all this information in a routing update is expensive.



Hosts of group G in color

4. *Explain protocol independent multicast (PIM) using an example.(April/May 17)*

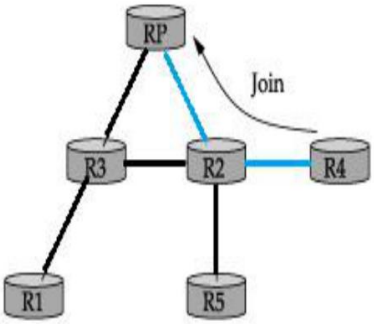
- PIM divides multicast routing problem into *sparse* and *dense* mode.
- PIM sparse mode (PIM-SM) is widely used. PIM does not rely on any type of unicast routing protocol, hence protocol independent.
- Routers explicitly join and leave multicast group using Join and Prune messages.
- One of the router is designated as *rendezvous point* (RP) for each group in a domain to receive PIM messages.
- Multicast forwarding *tree* is built as a result of routers sending Join messages to RP.
- Initially the tree is *shared* by multiple senders and depending on traffic it may be *source-specific* to a sender.

Shared Tree

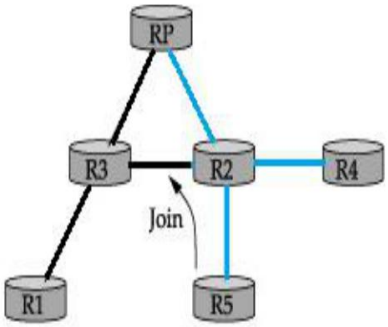
- When a router sends Join message for group *G* to RP, it goes through a set of routers.
  - Join message is *wildcarded* (*), i.e., it is applicable to all senders.
  - Routers create an *entry* (*, *G*) in its forwarding table for the shared tree.
  - *Interface* on which the Join arrived is marked to forward packets for that group. ○ *Forwards* Join towards rendezvous router RP.
- Eventually, the message arrives at RP. Thus a shared tree with RP as *root* is formed.

Example

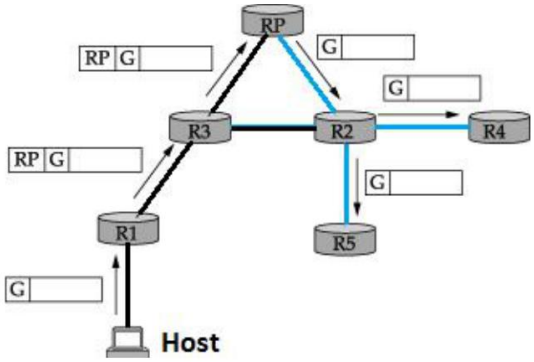
- Router *R4* sends Join message for group *G* to rendezvous router RP.
- Join message is received by router *R2*. It makes an entry (*, *G*) in its table and forwards the message to *RP*.
- When *R5* sends Join message for group *G*, *R2* does not forwards the Join. It *adds* an outgoing interface to the forwarding table created for that group.



Join from R4



Join from R5



Multicast message to group G

- As routers send Join message for a group, branches are *added* to the tree, i.e., shared.
- Multicast packets sent from hosts are forwarded to *designated* router RP.
- Suppose router *R1*, receives a message to group *G*.

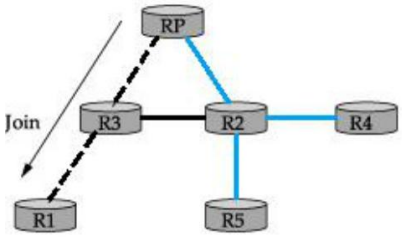
  R1 has no state for group *G*.
  - Encapsulates the multicast packet in a Register message.
  - Multicast packet is tunneled along the way to RP.

  RP decapsulates the packet and sends multicast packet onto the shared tree, towards *R2*.
  R2 forwards the multicast packet to routers *R4* and *R5* that have members for group *G*.

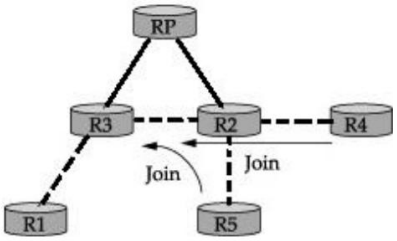
Source-specific tree.

RP can force routers to know about group *G*, by sending Join message to the sending host, so that tunneling can be avoided.
Intermediary routers create *sender-specific* entry (*S*, *G*) in their tables. Thus a source-specific route from *R1* to RP is formed.
If there is high rate of packets sent from a sender to a group *G*, then shared-tree is *replaced* by source-specific tree with sender as root.

Example



Source-specific Join from RP



Routers switch to Source tree

Rendezvous router RP sends a Join message to the host router *R1*.
Router *R3* learns about group *G* through the message sent by RP.
Router *R4* send a source-specific Join due to high rate of packets from sender.
Router *R2* learns about group *G* through the message sent by *R4*.
Eventually a source-specific tree is formed with *R1* as root.

Analysis

- Protocol independent because, tree is based on Join messages via *shortest* path.
- Shared trees are more *scalable* than source-specific trees.
- Source-specific trees enable *efficient* routing than shared trees.

Unit 4 TRANSPORT LAYER

Part A

1. What are the QOS parameters in transport layer? (April/May 2015)

A QoS value applies to the whole of a network connection. It must be identical at both ends of the connection, even if it is supported by several interconnected subnetworks each offering different services.

QoS is described by parameters. Defining a QoS parameter indicates how to measure or determine its value, mentioning if necessary the events specified by the network service primitives.

2.How does the transport layer perform duplication control. (April/May 2015)

The transport layer is one of the 7 layers of the OSI model. It's purpose is to provide robust end-to-end service to to the upper layers and is responsible for end-to-end delivery of the message. Therefore, the transport layer must consider addressing, reliability, flow control and multiplexing in order to accomplish it's main goals.

Reliable delivery is provided by using error control, sequence control, loss control and duplication control.

3. What are the advantages of connectionless over connection oriented services (April/May 17)

Connection-oriented communication includes the steps of setting up a call from one computer to another, transmitting/receiving data, and then releasing the call, just like a voice phone call. Connection-oriented services must first establish a connection between the two end-points (sending/receiving) before passing any data traffic between them. Connection-oriented service involves three phases:

1. Connection Establishment
2. Data Transfer
3. ConnectionTtermination

Connectionless services can send data without requiring an established connection. A packet transmitted in a connectionless mode is frequently called a datagram.

Connection-oriented services provide some level of delivery guarantee, whereas connectionless services do not.

4. How do fast retransmit mechanism of TCP works? (April/May 17)

Fast retransmit is a modification to the congestion avoidance algorithm. As in Jacobson's ***fast retransmit*** algorithm, when the sender receives 3rd duplicate ACK, it assumes that the packet is lost and retransmit that packet without waiting for a retransmission timer to expire. After retransmission, the sender continues normal data transmission. That means TCP does not wait for the other end to acknowledge the retransmission.

5. Distinguish between connection-less and connection-oriented protocol in transport layer.(Nov/Dec 16)

UDP (Connection-less)	TCP (Connection-oriented)
Datagram model (connection-less)	Byte-stream service (connection-oriented)
Unreliable delivery	Reliable delivery using acknowledgement
No flow control	Supports flow control
No congestion control	Built-in congestion control mechanism
Light overhead	Heavy overhead
Data is collected in order of receipt	Segments are ordered using sequence number

6.

7. What is slow start in TCP congestion? (May/Jun 16)

TCP slow start is an algorithm which balances the speed of a network connection. Slow start gradually increases the amount of data transmitted until it finds the network's maximum carrying capacity.

TCP slow start is one of the first steps in the congestion control process. It balances the amount of data a sender can transmit (known as the congestion window) with the amount of data the receiver can accept (known as the receiver window). The lower of the two values becomes the maximum amount of data that the sender is allowed to transmit before receiving an acknowledgment from the receiver.

8. What are the different phases in TCP connection? (May/Jun 16)

- TCP is connection-oriented.
  - Client performs an *active* connection to establish connection with a *passive* open server, prior to data communication
  - Eventually connection is terminated after data transmission.

9. *Distinguish between flow control and congestion control.(Nov/Dec 15)*

- Flow control prevents a fast sender from overrunning the capacity of slow receiver.
- Congestion control prevents too much data from being injected into the network, thereby causing switches or links overloaded beyond its capacity.
- Flow control is an end-to-end issue, whereas congestion control is interaction between hosts and network.

10. *Define QoS?(April/May 15)*

- Best-effort service offered by the network is insufficient for applications. They require assurances from network. For example:
  - Multimedia applications require minimum bandwidth.
  - Real-time applications require timeliness rather than correctness.
- Network that supports different level of service based on application requirements offer Quality of Service (QoS).
- QoS is defined as a set of attributes pertaining to the performance of a connection. Attributes may be either user or network oriented.

Part B

1. Write short notes on simple demultiplexer (or) UDP. (May/Jun 16)

- User Datagram Protocol (UDP) is a *connectionless*, *unreliable* transport protocol.
- Adds *process-to-process* communication to best-effort service provided by IP.
- Simple *demultiplexer* allows multiple processes on each host to communicate.
- *Does not provide* flow control / reliable / ordered delivery.
- UDP is *suitable* for a process that requires simple request-response communication with little concern for flow control/error control.

UDP Header

- UDP packets are known as user *datagrams* It has a 8-byte header.

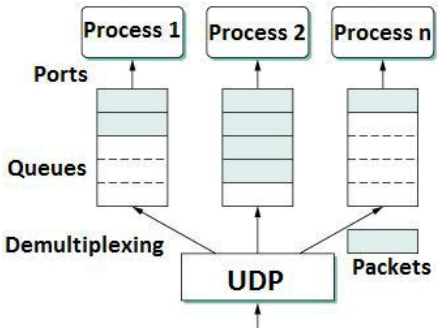
- SrcPort and DstPort—Source and destination port number.
- Length—total length of the user datagram, i.e., header plus data.
- Checksum—computed over UDP header, data and *pseudo header*. Pseudo header consist of IP fields (Protocol, SourceAddr, DestinationAddr) and UDP Length field. UDP delivers message to the correct recipient process using *checksum*.

Ports

- Processes (server/client) are identified by an abstract locator known as port.

- Server accepts message at *well known port*. Some well-known UDP ports are 7–Echo, 53–DNS, 111–RPC, 161–SNMP, etc.
- < *port*, *host* > pair is used as key for demultiplexing.
- Ports are implemented as a *message queue*.
  - When a message arrives, UDP *appends* it to end of the queue. ○ When queue is *full*, the message is discarded.

- When a message is *read*, it is removed from the queue.

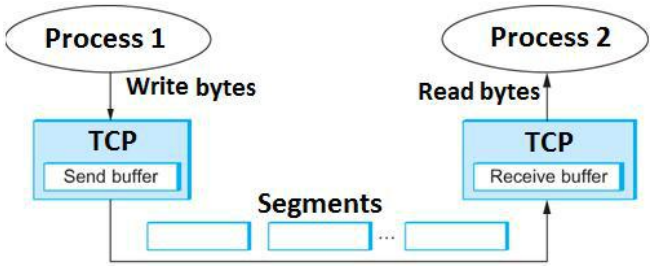


Applications

- Used for management processes such as SNMP.
- Used for route updating protocols such as RIP.
- It is a suitable transport protocol for multicasting.
- UDP is suitable for a process with internal flow and error control mechanisms such as Trivial File Transfer Protocol (TFTP).

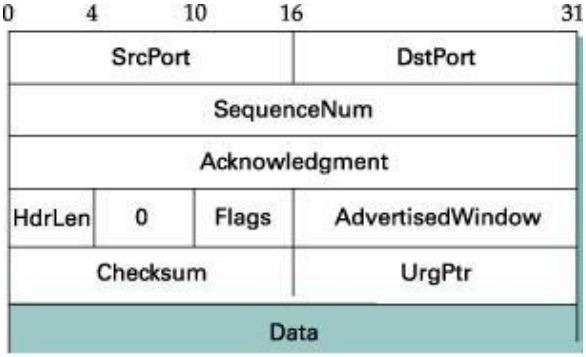
2. List the features of TCP. Draw TCP segment format and explain its fields.(April/May 15, Nov/Dec 16)

- Transmission Control Protocol (TCP) offers *connection-oriented*, *byte-stream* service.
- Guarantees *reliable*, *in-order* delivery of message.
- TCP is a *full-duplex* protocol.
- Like UDP, TCP provides *process-to-process* communication.
- Has built-in *congestion-control* mechanism.
- Ensures *flow control*, as sliding window forms heart of TCP operation.
- Some *well-known* TCP ports are 21–FTP, 23– TELNET, 25–SMTP, 80–HTTP, etc.
- Sending TCP buffers bytes in *send* buffer and transmits data unit as segments. Segments are stored in *receive* buffer at the other end for application to read.
- TCP's *demux* key is < SrcPort, SrcIPAddr, DstPort, DstIPAddr >



Segment Format

- Data unit exchanged between TCP peers are called *segments*.



- SrcPort and DstPort—*port number* of source and destination process.
- SequenceNum—contains sequence number, i.e. *first* byte of data segment.
- Acknowledgment— byte number of segment, the receiver expects *next*.
- HdrLen— length of TCP header as 4-byte *words*.
- Flags—contains *six* control bits known as flags.
  - *URG*—segment contains *urgent* data.

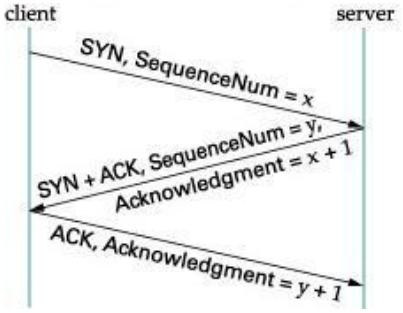
  ACK—value of *acknowledgment* field is valid.
  PUSH—sender has invoked the *push* operation.
  RESET—receiver wants to *abort* the connection.
  - *SYN*—synchronize sequence numbers during connection *establishment*.
- *FIN*—terminates the TCP *connection.*
- AdvertisedWindow—defines receiver's window size and acts as *flow control*.
- Checksum—It is computed over TCP *header*, *Data*, and *pseudo header* containing IP fields (Length, SourceAddr & DestinationAddr).
- UrgPtr—specifies first byte of *normal* data contained in the segment, if URG bit is set.

3. Explain TCP connection management (or) TCP architecture (or) state transition diagram.(Nov/Dec 15)

- TCP is connection-oriented.
  - Client performs an *active* connection to establish connection with a *passive* open server, prior to data communication
  - Eventually connection is terminated after data transmission.

Connection Establishment

- Connection establishment in TCP is a *three-way handshaking*.
  - Client sends a SYN segment to the server containing its initial sequence number (Flags = SYN, SequenceNum = *x*)
  - Server responds with a segment that acknowledges client's segment and specifies its initial sequence number (Flags = SYN+ ACK, Ack = *x* + 1 SequenceNum = *y*).
  - Finally, client responds with a segment that acknowledges server's sequence number (Flags = ACK, Ack = *y* + 1).



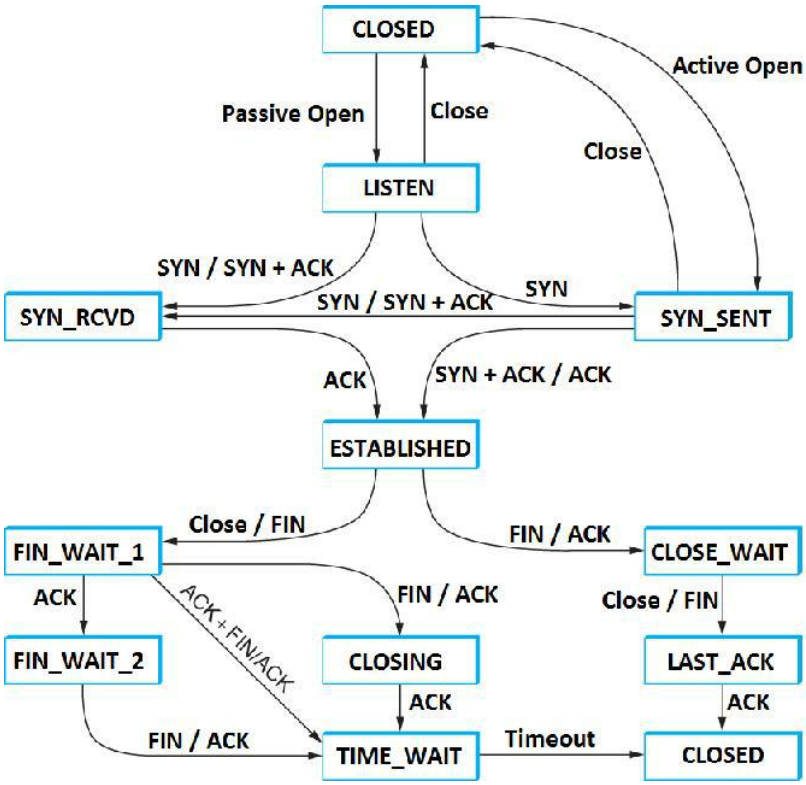
Connection Termination

- Connection termination or teardown can be done in two ways
- *Three-way close*—Both client and server close *simultaneously.*
  - Client sends a FIN segment. The FIN segment can include last chunk of data.
  - Server responds with FIN + ACK segment to inform its closing.
  - Finally, client sends an ACK segment.
- *Half-Close*—Client stops sending but receives data. This is known as *half-close*.
  - Client half-closes the connection by sending a FIN segment.
  - Server sends an ACK segment. Data transfer from client to the server *stops*.

- After sending all data, server sends FIN segment to client, which is acknowledged by the client.

State Transition Diagram

- States involved in opening and closing a connection is shown above and below ESTABLISHED state respectively.
- Events that trigger a state transition is:
  - Segments that *arrive* from its peer.
    Application process invokes an *operation* on TCP
- Operation of sliding window is hidden in the ESTABLISHED state



Opening

- s Server invokes a *passive* open on TCP, which causes TCP to move to LISTEN state
- t Client does an *active* open, which causes its TCP to send a SYN segment to the server and move to SYN_SENT state.
- u When SYN segment arrives at the server, it moves to SYN_RCVD state and *responds* with a SYN + ACK segment.
- v Arrival of SYN + ACK segment causes the client to move to ESTABLISHED state and sends an ACK to the server.
- w When ACK arrives, the server finally moves to ESTABLISHED state.

Closing

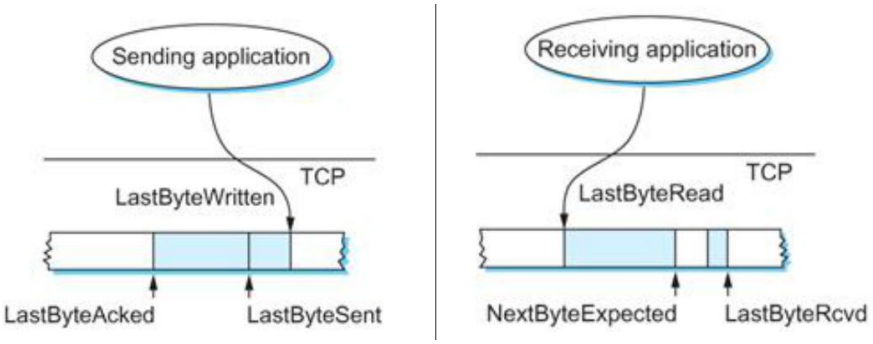
- Client / Server can independently close its half of the connection or simultaneously. Transitions from ESTABLISHED to CLOSED state are:
- *One side closes*: ESTABLISHED→FIN_WAIT_1→FIN_WAIT_2→TIME_WAIT→CLOSED
- *Other side closes*: ESTABLISHED → CLOSE_WAIT → LAST_ACK → CLOSED
- *Simultaneous close*: ESTABLISHED→FIN_WAIT_1→CLOSING→TIME_WAIT→ CLOSED

4. Explain TCP flow control (or) adaptive flow control (or) TCP Sliding window in detail.(April/May 17)

- TCP uses a variant of sliding window known as adaptive flow control that: o guarantees *reliable* delivery of data
  - ensures *ordered* delivery of data

- enforces *flow control* at the sender

- Receiver advertises its window size to the sender using AdvertisedWindow field.
- Sender thus cannot have *unacknowledged* data greater than AdvertisedWindow.



Send Buffer | *Receive Buffer*

Send Buffer

- Sending TCP maintains *send buffer* which contains 3 segments, acknowledged data, unacknowledged data and data to be transmitted.
- Send buffer maintains three *pointers* LastByteAcked, LastByteSent, and LastByteWritten such that:

  LastByteAcked ≤ LastByteSent ≤ LastByteWritten
- A byte can be sent only *after* being written and only a sent byte *can be* acknowledged.
- Bytes to the *left* of LastByteAcked are not kept as it had been acknowledged.

Receive Buffer

- Receiving TCP maintains *receive* buffer to hold data even if it arrives out-of-order.
- Receive buffer maintains three *pointers* namely LastByteRead, NextByteExpected, and LastByteRcvd such that:

  LastByteRead < NextByteExpected ≤ LastByteRcvd + 1
- A byte *cannot* be read until that byte and all preceding bytes have been received.
- If data is received *in order*, then NextByteExpected = LastByteRcvd + 1
- Bytes to the *left* of LastByteRead are not buffered, since it is read by the application.

Flow Control

- Size of *send* and *receive* buffer is MaxSendBuffer and MaxRcvBuffer respectively.
- Sending TCP prevents *overflowing* of send buffer by maintaining

  LastByteWritten − LastByteAcked ≤ MaxSendBuffer
- Receiving TCP avoids *overflowing* its receive buffer by maintaining

  LastByteRcvd − LastByteRead ≤ MaxRcvBuffer
- Receiver *throttles* the sender by having AdvertisedWindow based on *free* space available for buffering.

  AdvertisedWindow = MaxRcvBuffer − ((NextByteExpected − 1) − LastByteRead)
- Sending TCP *adheres* to AdvertisedWindow by computing EffectiveWindow that *limits* how much data it should send.

  EffectiveWindow = AdvertisedWindow − (LastByteSent − LastByteAcked)
- When data arrives, LastByteRcvd moves to its right and AdvertisedWindow shrinks.
- Receiver acknowledges only, if preceding bytes have arrived.
- AdvertisedWindow *expands* when data is *read* by the application.
  - If data is read as *fast* as it arrives then AdvertisedWindow = MaxRcvBuffer ○ If data is read *slowly*, it eventually leads to a AdvertisedWindow of size 0.

- AdvertisedWindow field is designed to allow sender to keep the pipe *full*.

Fast Sender vs Slow Receiver

- If sender transmits at a *higher* rate, receiver's buffer gets *filled* up. Hence, AdvertisedWindow shrinks, eventually to 0.
- Receiver advertises a window of size 0, thus sender cannot transmit as it gets *blocked*.
- When receiving process reads some data, those bytes are acknowledged and AdvertisedWindow expands.
- When an acknowledgement arrives for *x* bytes, LastByteAcked is incremented by *x* and send buffer space is freed accordingly to send further data.

5. Explain adaptive retransmission algorithms. (or) How is timeout estimated in TCP?

- TCP guarantees reliability through *retransmission* when ACK arrives after timeout.
- Timeout is based on RTT, but it is highly *variable* for any two hosts on the internet.
- Appropriate timeout is chosen using *adaptive* retransmission.

Original Algorithm

- SampleRTT is the *duration* between sending a segment and arrival of its ACK.
- EstimatedRTT is *weighted average* of previous estimate and current sample.

  EstimatedRTT = α × EstimatedRTT + (1 − α) × SampleRTT

  (where α is known as smoothening factor with value in the range 0.8–0.9)

  Timeout is determined as twice the value of EstimatedRTT .

  TimeOut = 2 × EstimatedRTT

- In original TCP, timeout is thus computed as function of *running average* of RTT.

Karn/Partridge Algorithm

Flaw discovered in TCP original algorithm was that an ACK segment, acknowledges *receipt* of data, not a transmission.

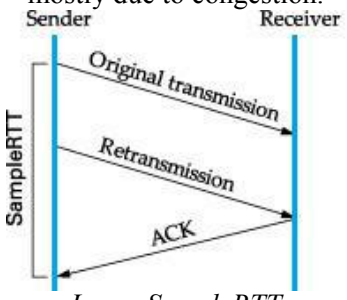
When an ACK arrives after retransmission, it is impossible to decide, whether to pair it with original or retransmitted segment for SampleRTT estimation.

- If ACK is associated with original one, then SampleRTT becomes too large
- If ACK is associated with retransmission, then SampleRTT becomes too small

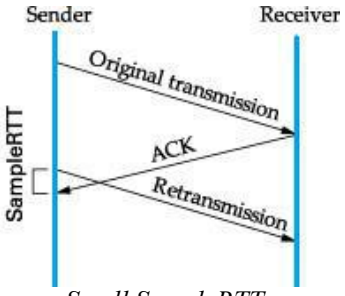
Karn and Partridge proposed that

SampleRTT should be taken for segments that are sent *only once*, i.e, for segments that are not retransmitted.

Each time TCP retransmits, timeout is *doubled*, since loss of segments is mostly due to congestion.



Large SampleRTT



Small SampleRTT

Jacobson/Karels Algorithm

- Jacobson and Karel discovered that problem with original algorithm was *variance* in SampleRTT was not considered.
- Mean RTT and variation in mean is calculated as:

  Difference = SampleRTT − EstimatedRT

  EstimatedRTT = EstimatedRTT + (δ × Difference)

  Deviation = Deviation + δ (|Difference|

  (*where δ is a fraction between 0 and 1*)

- TimeOut is computed as a function of both EstimatedRTT and Deviation as:

  TimeOut = μ × EstimatedRTT + ϕ × Deviation

  (*where μ = 1 and φ = 4*)

When *variance* is small, TimeOut is close to EstimatedRTT. If variation among samples is *small,* then EstimatedRTT can be trusted.

Define Congestion.

Congestion occurs if load (number of packets sent) is greater than capacity of the network (number of packets a network can handle).
When load is less than network capacity, throughput increases proportionally.
When load exceeds capacity, queues become full and the routers discard some packets and throughput declines sharply.
TCP uses mechanisms to control or avoid congestion.

6. Explain TCP congestion control mechanisms in detail.(Nov/Dec 15, Nov/Dec 16, April/May 15, May/Jun 16)

- Each source determines *capacity* of the network, so as to send packets without loss.
- TCP uses ACKs for further transmission of packets, i.e., *self-clocking*.
- TCP maintains a state variable CongestionWindow for each *connection*.
- A source is *not allowed* to send faster than network or destination host.

  MaxWindow = MIN(CongestionWindow, AdvertisedWindow)
- Congestion control mechanisms are:

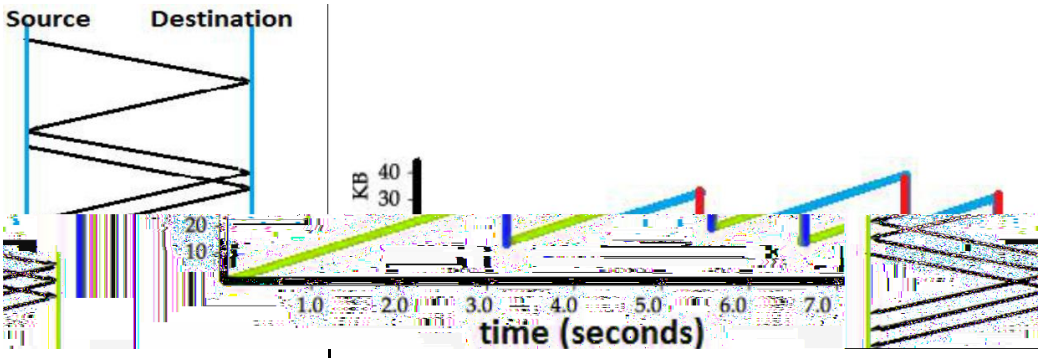
  Additive Increase / Multiplicative Decrease (AIMD)
  Slow Start
  Fast Retransmit and Fast Recovery

Additive Increase/Multiplicative Decrease (AIMD)

- TCP source *initializes* CongestionWindow based on congestion level in the network.
- Source *increases* CongestionWindow when level of congestion goes down and *decreases* the same when level of congestion goes up.
- TCP interprets *timeouts* as a sign of congestion and reduces the rate of transmission.
- On timeout, source reduces its CongestionWindow by half, i.e., *multiplicative decrease*. For example, if CongestionWindow = *16* packets, after timeout it is *8*.
- Value of CongestionWindow is never less than maximum segment size (MSS).
- When ACK arrives CongestionWindow is *incremented* marginally, i.e., *additive increase.*

  Increment = MSS × (MSS / CongestionWindow)

  CongestionWindow += Increment
- For *example*, when ACK arrives for 1 packet, 2 packets are sent. When ACK for both packets arrive, 3 packets are sent and so on.
- CongestionWindow increases and decreases throughout *lifetime* of the connection.
- When CongestionWindow is plotted as a function of time, a *saw-tooth* pattern results.



Additive Increase | *CongestionWindow Trace*

Analysis

- AIMD decreases its CongestionWindow *aggressively* but increases *conservatively*.
- Small CongestionWindow results in *less probability* of packets being dropped..
- AIMD is appropriate when source is operating close to capacity of the network.

Slow Start

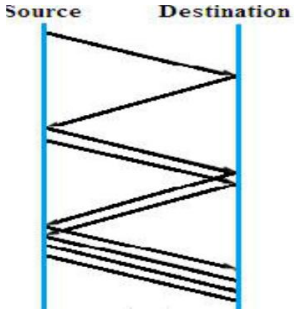
- Slow start is used to increase CongestionWindow *exponentially* from a cold start.
- Source TCP *initializes* CongestionWindow to one packet.
- TCP *doubles* the number of packets sent every RTT on successful transmission.
  - When ACK arrives for first packet TCP adds 1 packet to CongestionWindow and sends two packets.
  - When two ACKs arrive, TCP increments CongestionWindow by 2 packets and sends four packets and so on.
- Instead of sending entire permissible packets at once (bursty traffic), packets are sent in a phased manner, i.e., *slow start*.
- Initially TCP has no idea about congestion, henceforth it increases CongestionWindow rapidly until there is a timeout. On timeout:

  CongestionThreshold = CongestionWindow /

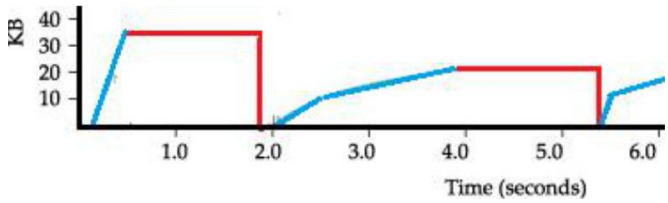
  2 CongestionWindow = 1
- Slow start is repeated until CongestionWindow reaches CongestionThreshold and thereafter 1 packet per RTT.

Example

- Initial slow start causes increase in CongestionWindow up to 34KB,
- Congestion occurs at 0.4 seconds and packets are lost.
- ACK does not arrive and therefore trace of CongestionWindow becomes flat.
- Timeout occurs at 2 sec. CongestionThreshold=17KB, CongestionWindow=1PKT
- Slow start is done till 17KB and additive increase thereafter till congestion occurs.



Exponential Increase



CongestionWindow Trace

Analysis

- Slow start provides exponential growth and is designed to avoid *bursty* nature of TCP.
- TCP loses more packets initially, because it attempts to learn the available *bandwidth* quickly through exponential increase.
- If connection goes *dead* while waiting for timer to expire, slow start phase is used only up to current value of CongestionWindow.

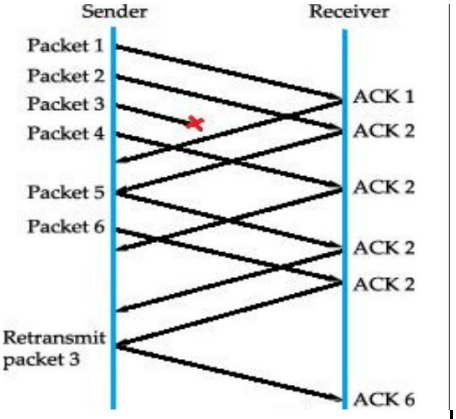
Fast Retransmit and Fast Recovery

- TCP timeouts led to long periods of time during which the connection went dead while waiting for a timer to expire.
- Fast retransmit is a heuristic approach that *triggers* retransmission of a dropped packet sooner than the regular timeout mechanism. It *does not* replace regular timeouts.
- When a packet arrives out of order, receiving TCP resends the same acknowledgment (*duplicate ACK*) it sent last time.
- When *three duplicate* ACK arrives at the sender, it infers that corresponding packet may be lost due to congestion and retransmits that packet. This is called *fast retransmit* before regular timeout.
- When packet loss is detected using fast retransmit, the slow start phase is replaced by additive increase, multiplicative decrease method. This is known as *fast recovery*.
- Instead of setting CongestionWindow to one packet, this method uses the ACKs that are still in pipe to clock the sending of packets.

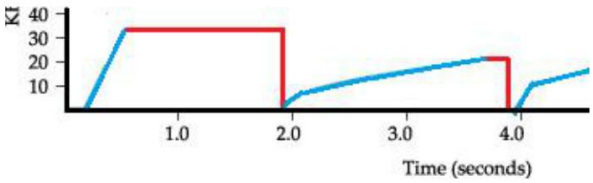
- Slow start is only used at the beginning of a connection and after *regular* timeout. At other times, it follows a pure AIMD pattern.

Example

- In *example*, packets 1 and 2 are received whereas packet 3 gets lost.
  - Receiver sends a duplicate ACK for packet 2 when packet 4 arrives.
  - Sender receives 3 duplicate ACKs after sending packet 6 retransmits packet 3.
  - When packet 3 is received, receiver sends cumulative ACK up to packet 6.
- In *example* trace, slow start is used at beginning and during timeout at 2 secs.
  - Fast recovery avoids slow start from 3.8 to 4 sec.
  - CongestionWindow is reduced by half from 22 KB to 11 KB.
  - Additive increase is resumed thereafter.



Duplicate ACK



CongestionWindow Trace

Analysis

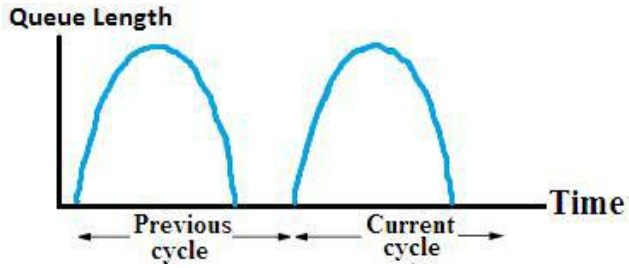
- Long periods with flat congestion window and no packets sent are *eliminated*.
- TCP's fast retransmit can detect up to three dropped packets per window.
- Fast retransmit/recovery increases throughput by 20%.

7. Explain in detail about TCP congestion avoidance algorithms.(April/May 17)

- Congestion avoidance mechanisms *prevent* congestion before it actually occurs.
- TCP *creates* loss of packets in order to determine bandwidth of the connection.
- Routers *help* the end nodes by intimating when congestion is likely to occur.
- Congestion-avoidance mechanisms are:
  - DECbit
  - Random Early Detection (RED)
  - Source-based congestion avoidance

DECbit

- Each router monitors its load and *explicitly* notifies the end node when congestion is likely to occur. Source *reduces* its transmission rate and congestion is avoided.
- A binary congestion bit called *DECbit* is *added* to the packet header.
- Router *sets* this bit in packets that flow through, if its average queue length is ≥ 1.
  - Average queue length is measured over a time interval that includes the *last busy* + *last idle* cycle + *current busy* cycle.
  - Calculates average queue length by *dividing* the curve area with time interval.



- Destination host *copies* the DECbit onto ACK and sends it back to the source.
- Source checks *how many* ACK has DECbit set for previous window packets.

- If less than 50% of ACK have DECbit set, then source *increases* its congestion window by 1 packet, otherwise *decreases* the congestion window by 87.5%.
- *Increase by 1, decrease by 0.875* rule was based on AIMD for stabilization.

Random Early Detection (RED)

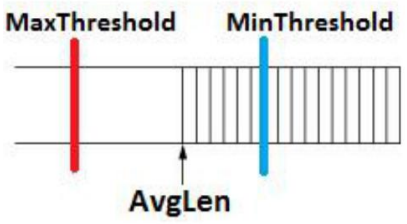
- Router notifies the source that congestion is likely to occur by *dropping* packets before its buffer space exhausts (*early drop*), rather than later due to congestion.
- Source is *implicitly* notified by timeout or duplicate ACK.
- Each incoming packet is dropped with a probability known as *drop probability* when the queue length exceeds *drop level*. This is called early random drop.
- Average queue length is computed as a weighted running average:

  AvgLen = (1 − Weight) × AvgLen + Weight × SampleLen
- Queue length *thresholds* defined by RED are MinThreshold and MaxThreshold.
- When a packet arrives, gateway *compares* current AvgLen with these thresholds and decides whether to queue or drop the packet as follows:

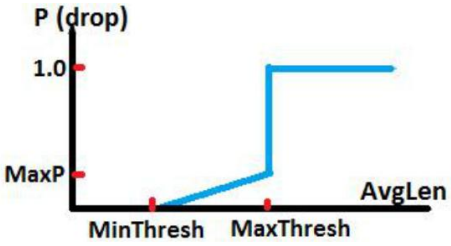
```
if AvgLen ≤ MinThreshold
        Queue the packet
if MinThreshold < AvgLen < MaxThreshold
        Calculate probability P
        Drop the arriving packet with probability P
if AvgLen ≥ MaxThreshold Drop
        the arriving packet
```

- When AvgLen exceeds MinThreshold, a small percentage of packets are dropped. It forces TCP to reduce CongestionWindow, which in turn reduces the rate at which packets arrive at the router. Thus, AvgLen decreases and congestion is *avoided*.
- Drop probability P is computed as a function of AvgLen.

  P = MaxP × (AvgLen − MinThreshold) / (MaxThreshold − MinThreshold)
- Drop probability increases slowly when AvgLen is between two thresholds. On reaching MaxP at the upper threshold, it jumps to unity.
- MaxThreshold value is twice of MinThreshold due to bursty Internet traffic.
- RED drops packets *randomly*. The probability that a flow's packet being dropped is proportional to its share of the bandwidth.



RED thresholds



Drop probability function

Source-Based Congestion Avoidance

- Source looks for signs of congestion in the network. For instance, increase in RTT indicates queuing at a router.

Some mechanisms

1. TCP checks to see if current RTT is greater than mean RTT. If so, congestion window is decreased by one-eighth, else normal increase.
2. TCP increases window size by one packet and compares the throughput achieved when the window was one packet smaller.

TCP Vegas

- *Throughput* increases as congestion window increases. Increase in window size beyond available bandwidth, results in packets queuing at the bottleneck router.
- TCP Vegas goal is to measure and control the right amount of *extra data* in transit.
- Extra data refers to amount of data that source would have refrained from sending so as to not *exceed* the available bandwidth.

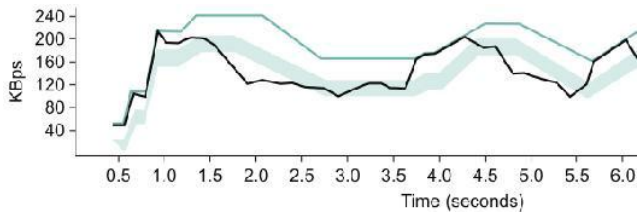
- A flow's BaseRTT is set to RTT of a packet when the flow is not congested.

  BaseRTT = MIN(RTTs)
- Expected throughput without overflowing is:

  ExpectedRate = CongestionWindow / BaseRTT
- ActualRate, i.e., current sending rate for a packet is calculated by recording bytes transmitted during a RTT.

  ActualRate = ByteTransmitted / SampleRTT
- ExpectedRate and ActualRate are compared.

  Diff = ExpectedRate – ActualRate
- Thresholds α and β are defined and corresponds to less data and too much extra data in the network, such that α < β.
- TCP uses difference in rates and adjusts CongestionWindow accordingly.
  - If Diff < α, CongestionWindow is linearly increased during the next RTT
  - If Diff > β, CongestionWindow is linearly decreased during the next RTT
  - If α < Diff < β, CongestionWindow is unchanged
- When actual and expected rates *vary* significantly, it indicates congestion in the network. The β threshold triggers *decrease* in sending rate.
- When actual and expected rate is almost the *same*, there is available bandwidth that goes wasted. The α threshold triggers *increase* in sending rate.
- Overall goal of TCP Vegas is to keep *between* α and β extra bytes in the network.



Black line (actual throughput), color line (expected throughput) α and β threshold (shaded region)

Difference between DECbit and RED.

- In DECbit, explicit notification about congestion is sent to source, whereas RED implicitly notifies the source by dropping a few packets.
- DECbit may lead to tail drop policy, whereas RED drops packet based on drop probability in a random manner.

Explicit congestion notification

- RED drops packets early to notify congestion is not acceptable to applications that are intolerant to delay or loss of packets.
- A bit in the TOS field can be set by routers along the path when congestion is encountered. The bit is echoed back to source. This is known as explicit congestion notification (ECN).

Unit 5 – APPLICATION LAYER

Part A

1. Define SMTP. (April/May 15)

"Simple Mail Transfer Protocol" is the protocol used for sending e-mail over the Internet. Your e-mail client (such as Outlook, Eudora, or Mac OS X Mail) uses SMTP to send a message to the mail server, and the mail server uses SMTP to relay that message to the correct receiving mail server. Basically, SMTP is a set of commands that authenticate and direct the transfer of electronic mail.

2. What are the groups of HTTP header? (April/May 15)

HTTP headers allow the client and the server to pass additional information with the request or the response. A request header consists of its case-insensitive name followed by a colon ':', then by its value.

Headers can be grouped according to their contexts:

- General header: Headers applying to both requests and responses but with no relation to the data eventually transmitted in the body.
- Request header: Headers containing more information about the resource to be fetched or about the client itself.
- Response header: Headers with additional information about the response, like its location or about the server itself (name and version etc.).
- Entity header: Headers containing more information about the body of the entity, like its content length or its MIME-type.

Headers can also be grouped according to how proxies handle them:

- End-to-end headers
- Hop-by-hop headers

3. State the use of conditional get in HTTP. (April/May 17)

- The HTTP Protocol defines a caching mechanism, in which the proxy web-servers can cache pages, files, images etc. Since caching is in place, There is a method which the servers are asked to return the document, either the "cached" or "live" document.
- This request of asking the server for a document considering a specific parameter is called a **Conditional GET Request**. In this request, a specific request header is sent **If-Modified-Since**. This header sends a RFC 2822 formatted date as the value.

4. Present the information contained in DNS resource record. (April/May 17)

DNS resource records are contents of the DNS zone file. The zone file contains mappings between domain names and IP addresses in the form of text records. There are many types of the resource records.

The common types of DNS Resource Records are given below. There are many other resources also.

1) SOA
2) NS
3) A
4) PTR
5) CNAME
6) MX
7) SRV

Start of Authority

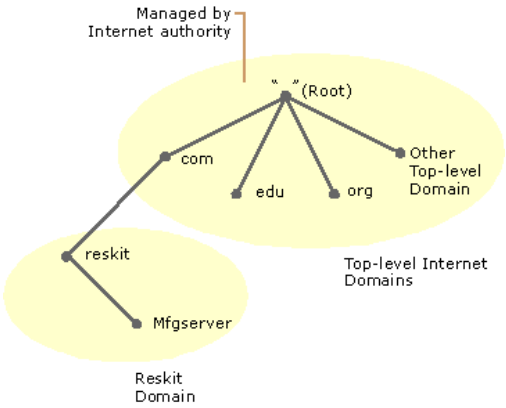
At the top level of a domain, the name database must contain a Start of Authority (SOA) record. This SOA record identifies what is the best source of information for data within the domain. SOA contains the current version of the DNS database, and various other parameters that define the behavior of a particular DNS server.

5. Define URL.(May/Jun 16)

A URL (Uniform Resource Locator), as the name suggests, provides a way to locate a resource on the web, the hypertext system that operates over the internet. The URL contains the name of the protocol to be used to access the resource and a resource name. The first part of a URL identifies what protocol to use. The second part identifies the IP address or domain name where the resource is located.

6. **Mention the different levels in domain space. .(May/Jun 16)**

The naming system on which DNS is based is a hierarchical and logical tree structure called the *domain namespace* . Organizations can also create private networks that are not visible on the Internet, using their own domain namespaces. Figure 5.1 shows part of the Internet domain namespace, from the root domain and top-level Internet DNS domains, to the fictional DNS domain named reskit.com that contains a host (computer) named Mfgserver.



7. **Expand POP3 and IMAP4. (Nov/Dec 16)**

POP (Post Office Protocol) version 3 and IMAP (Internet Message access protocol) ver 4
POP simply downloads email to your computer, and usually (but not always) deletes the email from the remote server.
IMAP allows users to store their email on remote servers. This two-way protocol also allows the user to synchronize their email among multiple devices, which is extremely important today, when most people have at least two devices - their laptop and smartphone.

8. **What is persistent HTTP? (Nov/Dec 16)**

HTTP persistent connections, also called HTTP keep-alive, or HTTP connection reuse, is the idea of using the same TCP connection to send and receive multiple HTTP requests/responses, as opposed to opening a new one for every single request/response pair. Using persistent connections is very important for improving HTTP performance.
There are several advantages of using persistent connections, including:

- Network friendly. Less network traffic due to fewer setting up and tearing down of TCP connections.
- Reduced latency on subsequent request. Due to avoidance of initial TCP handshake
- Long lasting connections allowing TCP sufficient time to determine the congestion state of the network, thus to react appropriately.

9. List the advantages of IMAP over POP.

IMAP is more powerful and more complex than POP.

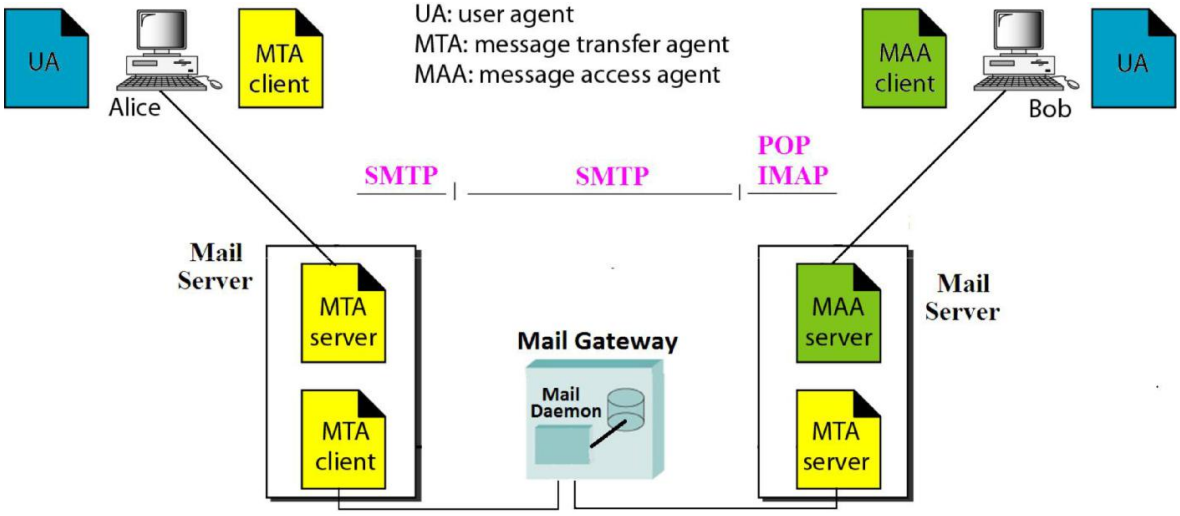
- User can *check* the e-mail header prior to downloading.
- User can *search* e-mail for a specific string of characters prior to downloading.
- User can download *partially*, very useful in case of limited bandwidth.
- User can create, delete, or rename *mailboxes* on the mail server.

10. What is hypertext?

Hypertext is a text that contains *embedded* URL known as links.
When hypertext is clicked, browser opens a new connection, retrieves file from the server and displays the file.

Part B

1. Explain email message transfer using Simple Mail Transfer Protocol. (May/Jun 16, April/May 15)



Alice sends a mail to Bob

An email system involves:

- User interface or agent
- Companion protocols that defines message format (RFC 822 and MIME) ○ Message transfer protocol (SMTP)
- ○ Mail Readers (IMAP and POP)

User Agent

User agent (UA) is software (eg. Microsoft Outlook, Netscape) that facilitates: ○ *Compose* create message by providing template with built-in editor.

- ○ *Read* read mail and provide sender, subject, flag (read, new) information.
- ○ *Reply* allows user to reply (send message) back to sender
- ○ *Forward* facilitates forwarding message to a third party.
- ○ *Mailboxes* two mailboxes for each user namely *inbox* and *outbox*.

Message Format

RFC 822 defines email message with two parts namely *header* and *body*.

Each header line contains *type* and *value* separated by a colon (:). Some are: ○ From identifier sender of the message.

- ○ To mail address of the recipient(s).
- ○ Subject says about purpose of the message.
- ○ Date mestamp of when the message was transmitted.

E-mail address is *userid@domain* where *domain* is hostname of the *mail server*. Header is separated from the body by a *blank* line.

Body contains the *actual* message.

Multipurpose Internet Mail Extension (MIME)

Email system was designed to send messages only in *ASCII* format.

- ○ Languages such as French, Chinese, etc., are not supported.
- ○ Image, audio and video files cannot be sent.

MIME is a protocol that *converts* non-ASCII data to ASCII and vice-versa. *Headers* defined in MIME are:

- MIME-Version current version, i.e., 1.1

- Content-Type message type (text/html, image/jpeg, application/pdf, etc) ○ Content-Transfer-Encoding message encoding scheme (eg base64).
  - ○ Content-Id unique identifier for the message.
  - ○ Content-Description describes type of the message body.

Example

```
MIME-Version: 1.1
Content-Type: multipart/mixed; boundary="-------417CA6-------"
From: Alice Smith <alice@cisco.com>
To: bob@cs.princeton.edu
Subject: photo
Date: Mon, 07 Sep 1998 19:45
---------417CA6-------
Content-Type: text/plain
Content-Transfer-Encoding: 7bit
PFA my photo
---------417CA6-------
Content-Type: image/jpeg
Content-Transfer-Encoding: base64
```

Message Transfer Agent (MTA)

MTA is a mail daemon (sendmail) active on hosts having mailbox, used to send an email. Mail passes through a sequence of *gateways* before it reaches the recipient mail server. Each gateway stores and forwards the mail using Simple mail transfer protocol (SMTP). SMTP defines communication between MTAs over TCP on port 25.

In an SMTP session, sending MTA is *client* and receiver is *server*. In each exchange:

Client posts a command (HELO, MAIL, RCPT, DATA, QUIT, VRFY, etc.)

Server responds with a code (250, 550, 354, 221, 251 etc) and an explanation.

Client is identified using HELO command and verified by the server

Client forwards message to server, if server is willing to accept.

Message is terminated by a line with only single period (.) in it.

Eventually client terminates the connection.

Example

Exchange between sending host *cs.princeton.edu* and receiving host *cisco.com* is:

```
HELO cs.princeton.edu
250 Hello
daemon@mail.cs.princeton.edu MAIL
FROM: <bob@cs.princeton.edu> 250 OK
```

```
RCPT TO: <alice@cisco.com>
 OK
DATA
 Start mail input
See u at conference
.
s OK
QUIT
 Closing connection
```

2. ***Explain the final delivery of email to the end user using POP3 / IMAP4. (April/May 15, Nov/Dec 16, April/May 17)***

Message Access Agent (MAA) or mail reader allows user to *retrieve* messages in the mailbox from a remote host, so that user can perform actions such as reply, forward, etc.

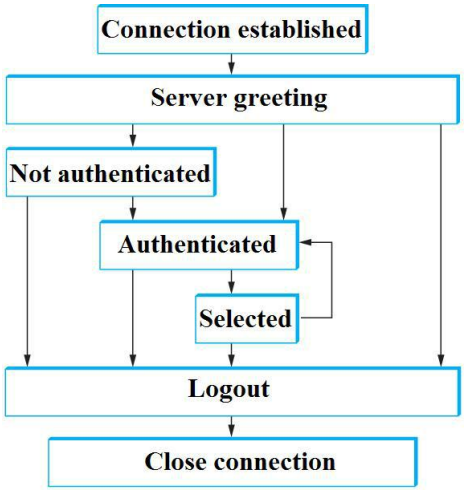
MAA protocols used are Post office protocol and Internet message access protocol

SMTP is a *push* type protocol whereas POP and IMAP are *pop* type protocol.

Internet Message Access Protocol (IMAP4)

IMAP is a client/server protocol running over TCP on port 143. Current version is 4.

Client authenticates itself in order to access the mailbox.



LOGIN, AUTHENTICATE, SELECT, EXAMINE, CLOSE, LOGOUT, FETCH, STORE, DELETE, etc., are some commands issued by the client.

Server responses are OK, NO (no permission), BAD (incorrect command), etc.

When user wishes to FETCH a message, server responds in MIME format.

Message *attributes* such as size are also exchanged.

Flags (Seen, Answered, Deleted, Recent) are used by client to report user actions.

Post Office Protocol (POP3)

POP is *simple* and limited in functionality. Current version is POP3.

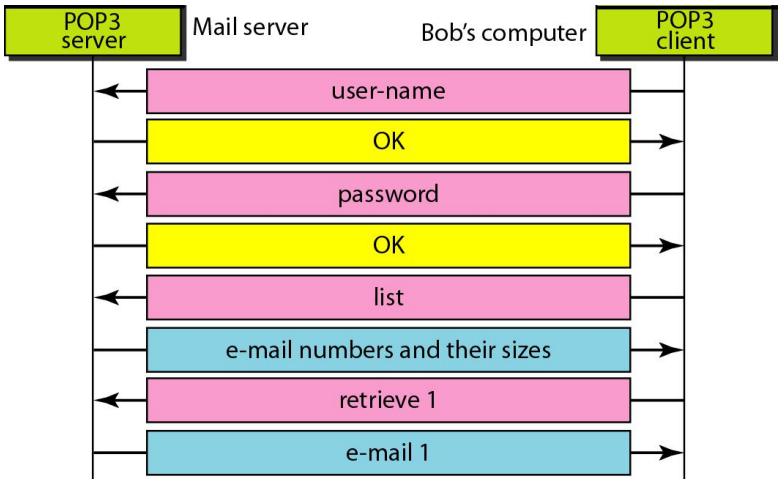
POP client is *installed* on the recipient computer and POP server on the mail server.

Client *opens* a connection to the server using TCP on port 110.

Client sends username and password to *access* mailbox and to retrieve messages.

\POP works in two modes namely, *delete* and *keep* mode.

- In delete mode, mail is *deleted* from the mailbox after retrieval
- In keep mode, mail after reading is *kept* in mailbox for later retrieval.



3. *Explain WWW or HTTP protocol or URL in detail.(Nov/Dec 15)*

WWW is a *distributed* client/server service, in which a client (Browsers such as IE, Firefox, etc.) can access services at a server (Web server such as IIS, Apache).

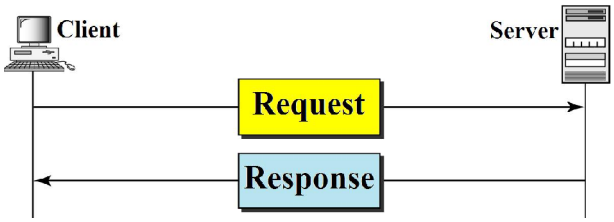
HyperText Transfer Protocol (HTTP) is a *stateless* request/response protocol that governs client/server communication using TCP on port 80.

Uniform Resource Locator (URL) provides information about its location on the Web http://www.*domain_name/filename*

When user enters URL, browser forms a *request* message and sends it to the server.

Web server retrieves the requested URL and sends back a *response* message.

Web browser renders the response in HTML or appropriate format.



Request Message

Request Line
Request Header : *Value*
Body (optional)

Request Line

Request line contains fields:

Request type URL HTTP version

HTTP version specifies current version of the protocol i.e., 1.1

Request type specifies methods that operate on the URL. Some are:

Method	Description
GET	Retrieve the URL filename
HEAD	Retrieve meta-information about the URL filename
PUT	Store document under specified URL
POST	Give information to server
TRACE	Loopback request message (echoing).
DELETE	Delete specified URL
CONNECT	Used by proxies

Request Header

Headers defined for request message include:

Request Header	Description
Authorization	specifies what permissions the client has
From	e-mail address of the user
Host	host name of the server
If-modified-since	server sends the URL if it is newer than specified date
User-agent	name of the browser

For *example*, request message to retrieve file *result.html* on host *annauniv.edu* is:

```
GET result.html HTTP/1.1
Host : www.annauniv.edu
```

Response Message

Status Line
Response Header : *Value*
Body

Status Line

Status line contains three fields:

HTTP version *Status code* *Status phrase*.

3-digit status code classifies HTTP result based on *leading digit* (1*xx*–Informational, 2*xx*– Success, 3*xx*–Redirection, 4*xx*–Client error and 5*xx*–Server error).

Status phrase gives brief description about *status code*. Some are:

Code	Phrase	Description
100	Continue	Initial request received, client to continue process
200	OK	Request is successful
301	Moved permanently	Requested URL is no longer in use
404	Not found	Document not found
500	Internal server error	An error such as a crash, at the server site

Response Header

Provides additional information to the client. Some are:

Response Header	Description
Content-type	specifies the MIME type
Expires	date and time up to which the document is valid
Last-modified	date and time when the document was last updated
Location	specifies location of the created or moved document

For *example*, response for a *moved* page is:

```
HTTP/1.1 301 Moved Permanently
Location : http://www.princeton.edu/cs/index.html.
```

TCP Connection

HTTP 1.1 uses *persistent connection*, i.e., client and server exchange multiple messages over the same TCP connection. The advantages are:

- Eliminates connection setup *overhead* and additional load on the server.
- Congestion window is very *efficient* by avoiding slow start phase for each page.
- Server closes the connection on timeout.

Caching

Caching enables the client to retrieve document *faster* and reduces load on the server. Caching is implemented at Proxy server, ISP router and Browser.

Server sets *expiration* date (Expires header) for each page, beyond which it is not cached.

Cache document is returned to client only if it is an *updated* copy by checking against If-Modified-Since header.

If cache document is *out-of-date*, then request is forwarded to the server and response is cached along the way.

A web page will not be cached if *no-cache* directive is specified.

Define Uniform Resource Identifiers (URI).

URI is a string that identifies resources such as document, image, service, etc. It is of the form *scheme* : *scheme-specific*

Scheme *identifies* a resource type, such as mailto for mail address, file for file name, etc. and scheme-specific is a *resource* identifier. Example is mailto:skvijaianand@gmail.com

URI identifies a resource, whereas URL is used to locate a resource.

What is non-persistent TCP connection? List the disadvantage.

In non-persistent, a *separate* connection for each data item retrieved from server.

For example, a page with text and dozen graphics requires *13* separate TCP connections. Two RTTs are incurred to setup each connection and at least another couple of RTT in sending request and retrieving responses.

Define proxy server.

Proxy server copies responses sent by the server to recent requests.

Client's request is intercepted by proxy and responds if it has a cache of the document, or else forwards request to the server.

4. Write a note on web services.(April/May 15,May/Jun 16)

Web services are architectures that offer remotely accessible services for client applications to form network applications, such as business-to-business (B2B) and enterprise application integration (EAI).

An example is, an application from Amazon.com tracking shipping of a book order by interacting with application on Fedex.com

Two web services architectures are WSDL/SOAP (*custom*) and REST (*generic*):

WSDL and SOAP are frameworks based on XML for specifying and implementing application protocols and transport protocols, customized to a network application

REST treats individual web services as WWW resources, identified by URI and accessed via HTTP.

Web Service Description Language (WSDL)

WSDL is an *operation* model, where a web interface is a set of named operations that represents interaction between client and web service.

Each operation specifies a *message exchange pattern* (MEP) that provides the sequence in which the messages are to be transmitted.

Commonly used MEPs are In-Only (a message from client to service) and In-Out (request from a client and corresponding reply from the service).

Message formats are defined as an *abstract* data model using XML Schema.

Concrete part specifies how MEPs are mapped onto it (*binding*). Predefined bindings exist for HTTP and SOAP-based protocols.

Specification of a web service may contain multiple WSDL documents, and these documents could be used in other web service.

Each WSDL document specifies URI of the *target* XML namespace. A

WSDL document can *incorporate* components of another by

including the second document if both share the same target

namespace ○ *importing* it if the target namespaces differ.

SOAP

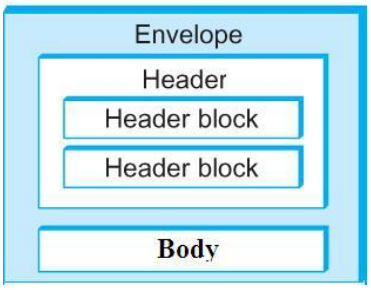
SOAP is used to define transport protocols with features required to support a particular application protocol.

A SOAP feature specification includes:

- URI that identi es the feature
- State information and processing required at each SOAP node for implementation ○ Information to be relayed to the next node
- ○ If MEP then, life cycle and temporal relationships of the messages exchanged.

SOAP is binded to an underlying protocol, to derive its features. This is known as layering. For *example* a request-response protocol is obtained by binding SOAP to HTTP.

A SOAP message consists of an *envelope*, which contains a *header* that contains header blocks, and a *body* that contains data.



SOAP message structure

A SOAP header block contains information that corresponds to a particular feature.

A SOAP *module* is a specification of syntax and semantics of one or more header blocks.

A given message may be processed not only by sender/receiver, but also by SOAP-aware nodes based on SOAP *role*.

For example, role next implies all nodes can process, whereas role ultimateReceiver specifies only receiver can process.

Each header block specifies a role. A node *processes* header blocks that specify a role assumed by the node and forwards the message.

A SOAP *fault* is generated if a node does not understand the blocks it should process.

REpresentational State Transfer (REST)

REST web services architecture is based on re-applying the model underlying the WWW architecture.

In REST model, *complexity* is shifted from protocol to the payload.

Payload is a representation of the abstract state of a resource. For example, a GET returns a representation of current state of the resource.

Message size is *reduced* by transmitting parts of a state by reference or URI.

XML and JSON are widely used as *presentation* language to define document structure.

REST uses infrastructure deployed to support the Web. For example, *proxies* can enforce security mechanism.

Web supports *intermediary* nodes as in SOAP. For example, since GET is read-only, nodes can cache the response.

Compare SOAP/WSDL and REST protocol.

WSDL/SOAP integrates application via protocols *customized* to each application protocol whereas REST adopts *generic* approach by using WWW architecture.

WSDL has user-defined operations, whereas REST uses HTTP methods GET and POST.

Interoperability in SOAP depends on the agreement with the underlying protocol, whereas in REST there is no interoperability problem.

Interface of legacy applications easily *map* onto WSDL operations than REST states.

What is a profile?

A *profile* is a set of guidelines that narrow choices in standards like WSDL, SOAP, etc.

Widely used profile is *WS-I Basic Profile*. It requires WSDL be bound exclusively to SOAP, and SOAP be bound exclusively to HTTP and use HTTP POST method.

WS-I Basic Security Profile adds security constraints to basic profile by specifying how SSL/TLS layer is to be used.

5. *Explain the role of DNS on a computer network (or) domain name resolution process.* (Nov/Dec 16, April/May 17)

DNS *maps* user-friendly domain names to router-friendly IP address, i.e., *middle-ware*.

Domain Naming System (DNS) includes:

- *namespace* to define domain names without any collision
- *binds* domain names to IP address
- *name server* to lookup IP address for a given name.

Domain Hierarchy

DNS implements *hierarchical* name space for domains in the Internet.

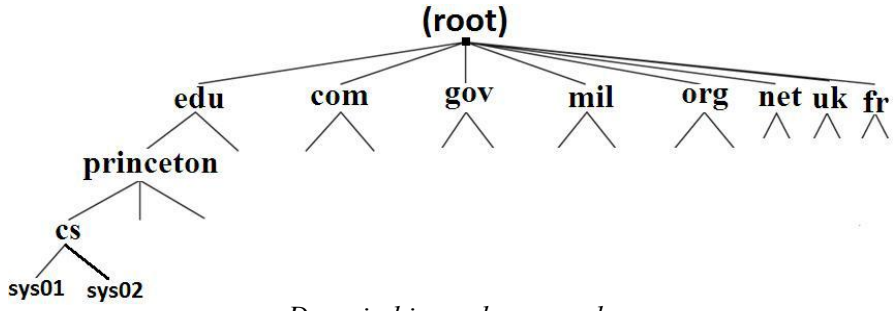
Domain names are processed from right to left and use *periods* (.) as separator.

DNS hierarchy is represented as a tree, where each *node* is a domain and *leaves* are hosts. Six top level domains (TLD) are .edu .com .gov .mil .org and .net

TLD also exists for each country, e.g., .fr (france) .in (india), etc.

Domain hierarchy is partitioned into *zones*. Each zone acts as *central* authority for that part of the sub-tree. For example, in .edu domain, *princeton* is a zone.

Zones can be further *sub-divided* such as *CS department* under *Princeton university*.



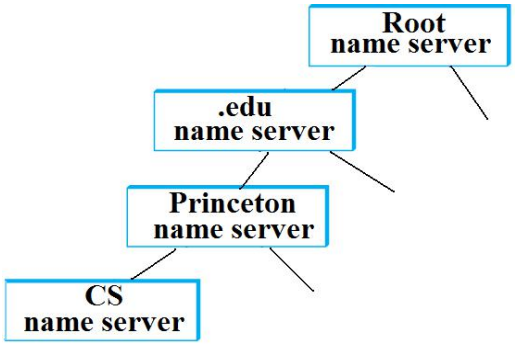
Domain hierarchy example

Name Servers (NS)

Root NS is maintained by NIC and TLD name servers are managed by ICANN.

Name servers contain zonal information as *resource records* that binds name-to-value.

Name servers receive *query* and return IP address of domain name or another NS to client



Hierarchy of CS Nameserver

Resource record is a *5-tuple* with fields <Name, Value, Type, Class, TTL>

- Name—specifies the domain/zone name. It is used as primary search key.
- Type—indicates what kind of record it is. Commonly used types are:
  - NS Value field contains address of a name server
  - MX Value field contains a mail server.
  - A Value field contains an IP address
  - CNAME Canonical name or alias name for that host
- Class field is always IN for internet domain names.
- TTL field gives an indication of how long the resource record is valid.

Resource Records

Root name server contain a NS record for each TLD name server and an A record that translates TLD into corresponding IP address.

```
edu, a3.nstld.com, NS, IN >
a3.nstld.com, 192.5.6.32, A, IN >
…
```

Each TLD name server has a NS record for each zone-level name server and an A record that translates zone name into corresponding IP address.

Resource records for TLD edu name server looks like:

```
princeton.edu, dns.princeton.edu, NS, IN >
dns.princeton.edu, 128.112.129.15, A, IN >
…
```

Zone name server *princeton.edu* resolves some queries directly and redirects others to a server at another layer in the hierarchy (*cs.princeton.edu*).

```
cs.princeton.edu, dns1.cs.princeton.edu, NS, IN >
dns1.cs.princeton.edu, 128.112.136.10, A, IN >
…
```

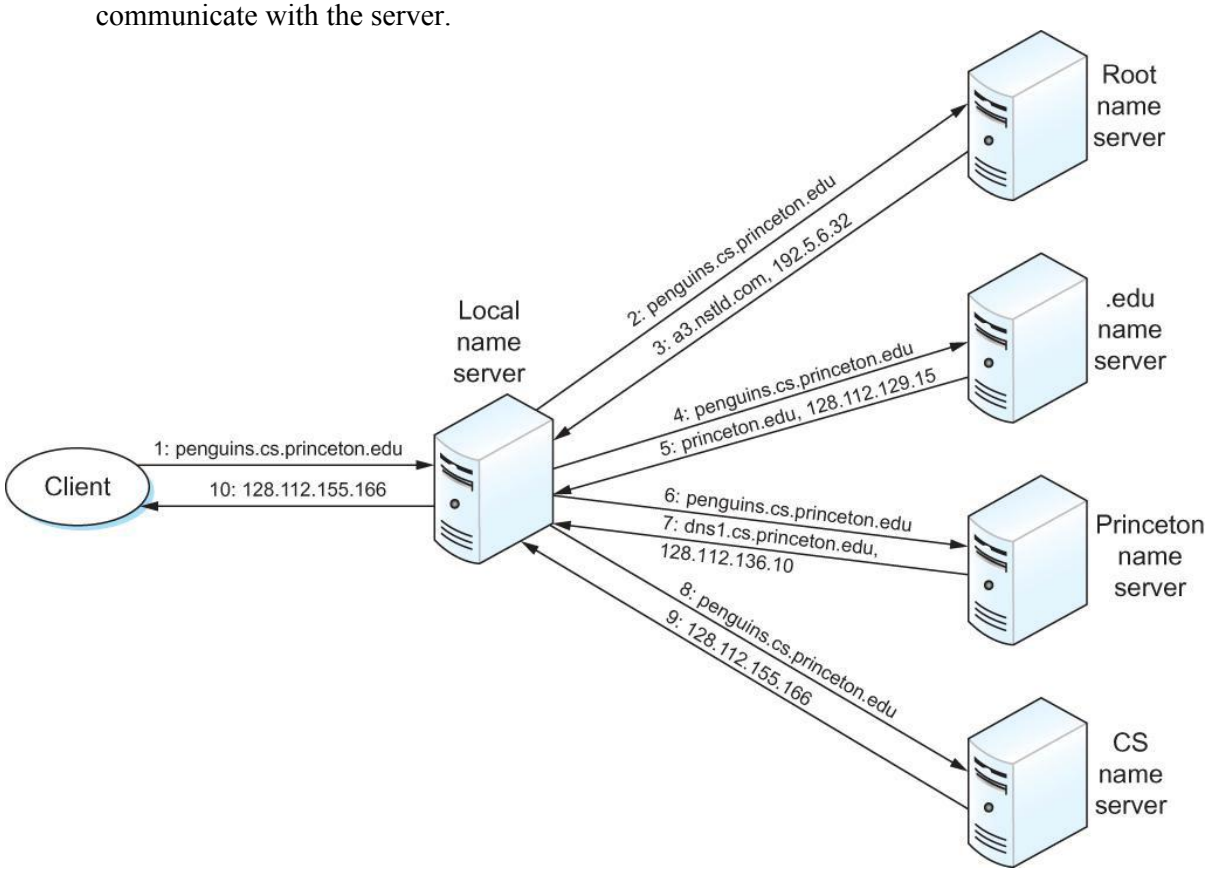
Third-level name server *cs.princeton.edu* contains A records for all hosts on that network.

```
 penguins.cs.princeton.edu, 128.112.155.166, A, IN >
…
```

Name Resolution

- p Client does not know address of root name server, therefore it sends query about *penguins.cs.princeton.edu* to the local name server in an UDP packet.
- q Local NS forwards the query to the *root* server.
- r Root server finds *no exact match*. Best match is NS record for edu that points to server *a3.nstld.com*. Therefore A record for *a3.nstld.com* is returned to local NS.
- s Local NS resends the query to *192.5.6.32* since it has not got IP address for the query.
- t edu server returns A record (*128.112.129.15*) for the best zone match *princeton.edu*
- u Local NS resends the query to zonal NS *princeton.edu* and receives A record (*128.112.136.10*) for *cs.princeton.edu*

- Finally local NS resends the query to *cs.princeton.edu* and gets the A record (*128.112.155.166*) for *penguins.cs.princeton.edu*
- Local NS *caches* the response and sends it to the client. Client uses the IP address to communicate with the server.



How was domain names resolved during early days of internet?

NIC maintained a table of name-to-address bindings called hosts.txt
Any host that joins the internet, *mails* its name and IP address to NIC.
NIC updates hosts.txt and mails it to all hosts. Thus a host comes to know about IP address of other hosts.

Internet grew in the 80's, after which hosts.txt approach *failed* and DNS evolved.

What is the need for DNS?

Host on a network is *uniquely* identified by its IP address. It is numeric with fixed length and *suitable* for processing by routers.
Host names are of variable-length and mnemonic. It is *easier* to remember than an IP address, but does not help in locating a host on the network.
DNS is required to find IP address for corresponding domain name, so that request message can be sent from the client.

6. *Explain how SNMP is used to manage nodes on the network (or) network management.(April/May 15, Nov/Dec 16, April/May 17)*

Simple Network Management Protocol (SNMP) is an application layer protocol that monitors and manages routers, distributed over a network.

SNMP uses the concept of *manager* and *agent*.

Manager is a host that runs SNMP *client* program (GUI)
Agent is a router that runs SNMP *server* program.

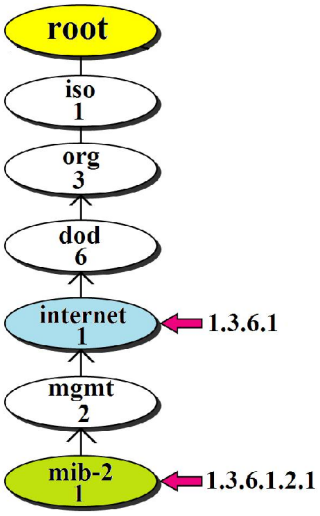
SNMP uses services of UDP on two well-known ports: 161 (agent) and 162 (manager).
SNMP is supported by two protocols:

Structure of Management Information (SMI)
- Management Information Base (MIB).

SMI Object Identifier

SMI defines rules for naming objects using *Abstract syntax notation* (ASN.1).
Basic Encoding Rules (BER) encoding is used to transmit data over the network.

Object identifiers are *hierarchical* that begins with root and uses *lexicographic* ordering.



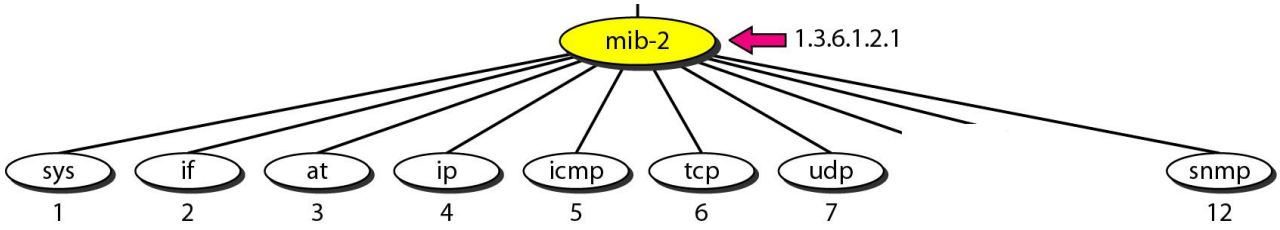
MIB Groups

Each agent has its own MIB, which is a *collection* of objects to be managed.

SNMP objects are located under mib-2 object, identifiers beginning with 1.3.6.1.2.1 MIB-II (version 2) classifies objects under *ten* groups. Some are:

- sys (*system* information about the node such as name, location, lifetime, etc.
- if (*interface* information about interfaces attached to the node such as physical address, packets sent and received on each interface, etc.
- at (*address translation* information about ARP table

- ip information about IP such as routing table, datagrams forwarded/dropped, etc
- tcp information related to TCP such as connection table, time-out value, number of TCP packets sent / received, etc.
- udp information on UDP traffic such as number of UDP packets sent/received.

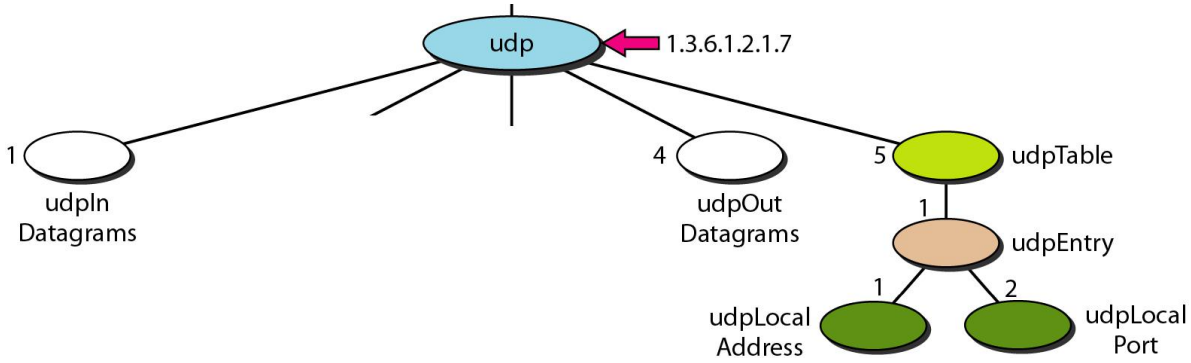


MIB variables

MIB variables are of two types namely *simple* and *table*.

Simple variables are accessed using *group-id* (1.3.6.1.2.1.7) followed by *variable-id* and 0 (*instance suffix*). For example, udpInDatagrams is accessed as 1.3.6.1.2.1.7.1.0

Tables are ordered as *column-row* rules, i.e., column by column from top to bottom. Only *leaf* elements are accessible in a table type.



UDP variables

Protocol Data Unit (PDU)

SNMP is request/reply protocol that supports various operations using

PDUs:
- GET used by manager to retrieve value of agent variable.
- GET-NEXT used by manager to retrieve next entries in an agent's table.
- SET used by manager to set value of an agent's variable.
- RESPONSE sent from an agent to manager in response to GET/GET-NEXT that contains *value* of variables.
- TRAP sent from agent to the manager to report an *event* such as reboot.

When administrator selects a piece of information, manager puts identifier for the MIB variable and sends *request* message to the agent.

Agent maps the identifier, *retrieves* value of the variable, and sends encoded value back to the manager.