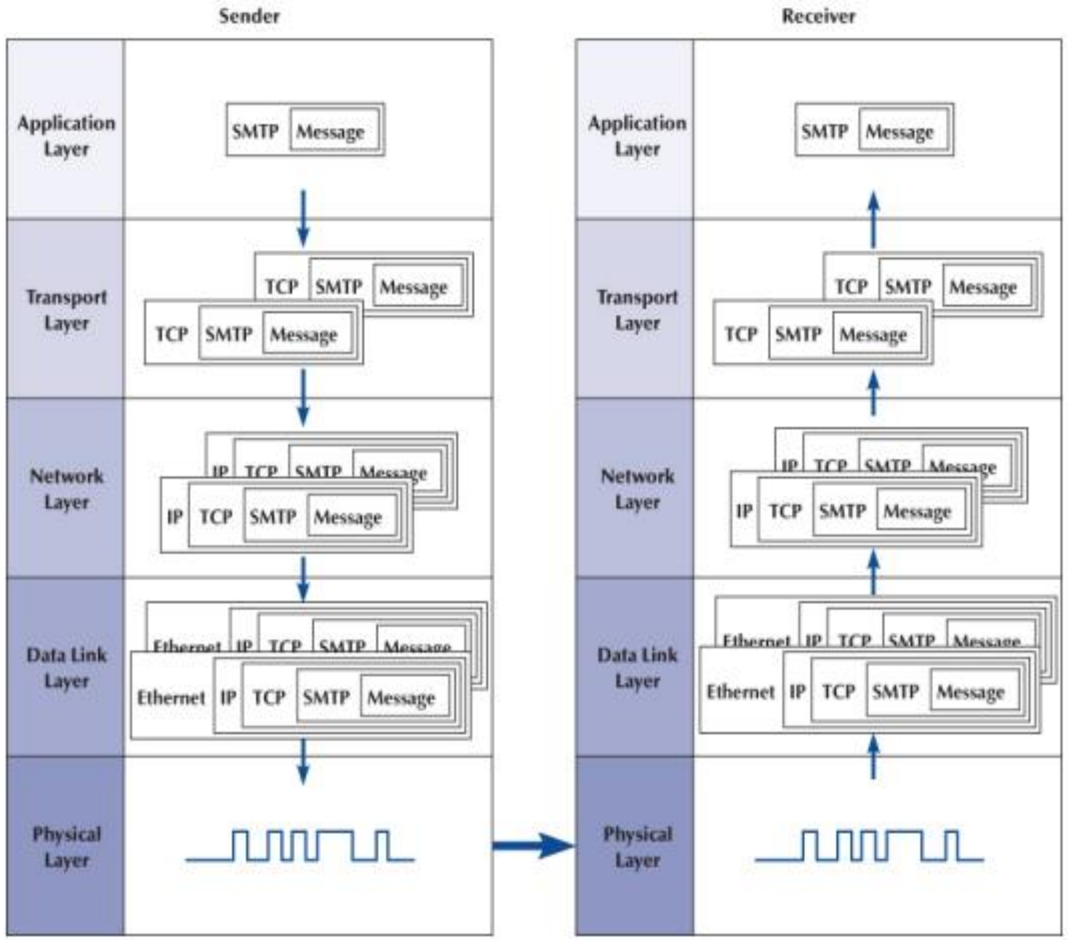
UNIT 5
NETWORK AND TRANSPORT LAYERS

CONTENTS

- Introduction; Transport and Network Layer Protocols (Transmission Control Protocol, Internet Protocol); Transport Layer Functions (Linking to the Application Layer, Segmenting, Session Management); Addressing (Assigning Addresses, Address Resolution); Routing (Types of Routing, Routing Protocols, Multicasting, The Anatomy of a Router); TCP/IP Example (Known Addresses, Unknown Addresses, TCP Connections, TCP/IP and Network Layers).

Introduction

- The transport layer links the application software in the application layer with the network and is responsible for the end-to-end delivery of the message. The transport layer accepts outgoing messages from the application layer (e.g., Web, email) and segments them for transmission. Figure shows the application layer software producing an Simple Mail Transfer Protocol (SMTP) packet that is split into two smaller TCP segments by the transport layer.
- The Protocol Data Unit (PDU) at the transport layer is called a segment. The network layer takes the messages from the transport layer and routes them through the network by selecting the best path from computer to computer through the network (and adds an IP packet). The data link layer adds an Ethernet frame and instructs the physical layer hardware when to transmit.



I . Message transmission using layers. SMTP = Simple Mail Transfer Protocol; HTTP = Hypertext Transfer Protocol; IP = Internet Protocol; TCP = Transmission Control Protocol

- The network and transport layers also accept incoming messages from the data link layer and organize them into coherent messages that are passed to the application layer. For example, as in Figure, a large email message might require several data link layer frames to transmit.
- The transport layer at the sender would break the message into several smaller segments and give them to the network layer to route, which in turn gives them to the data link layer to transmit. The network layer at the receiver would receive the individual packets from the data link layer, process them, and pass them to the transport layer, which would reassemble them into the one email message before giving it to the application layer.

- Network Layer is responsible for delivery of datagrams between two hosts. This is called **host-to-host delivery.**
- Transport Layer is responsible for delivery of entire message from one process running on source to another process running on destination. This is called **process-to process delivery.**

Introduction to Network Layer

- The network layer is the third layer of the Open Systems Interconnection Model (OSI Model) and the layer that provides data routing paths for network communication.
- Data is transferred in the form of packets via logical network paths in an ordered format controlled by the network layer.
- Logical connection setup, data forwarding, routing and delivery error reporting are the network layer’s primary responsibilities.
- The network layer involves each and every host and router in the network. The role of the network layer in a sending host is to begin the packet on its journey to the receiving host.
- Three important network-layer functions are:

- **Path determination.:** The network layer must determine the route or path taken by packets as they flow from a sender to a receiver. The algorithms that calculate these paths are referred to as routing algorithms.
- **Switching:** When a packet arrives at the input to a router, the router must move it to the appropriate output link.
- **Call setup:** With TCP, a three-way handshake is required before data actually flow from sender to receiver. This allowed the sender and receiver to set up the needed state information (for example, sequence number and initial flow control window size). Some network architectures require router call setup along path before data flows.

Introduction to Transport Layer

- The transport layer is the fourth layer from the bottom in the OSI reference model.
- It is responsible for message delivery from process running in source computer to the process running in the destination computer.
- Transport layer does not perform any function in the intermediate nodes.
- It is active only in the end systems.
- The transport layer has the critical role of providing communication services directly to the application processes running on different hosts in the network.
- The functions of Transport layer are as follows:

1. Service Point Addressing: Transport Layer header includes service point address which is port address. This layer gets the message to the correct process on the computer unlike Network Layer, which gets each packet to the correct computer.

2. Segmentation and Reassembling: A message is divided into segments; each segment contains sequence number, which enables this layer in reassembling and replaces packets which were lost in transmission.

3. Connection Control: It includes 2 types:

Connection Oriented- Before delivering packets, connection is made with transport layer at the destination machine.

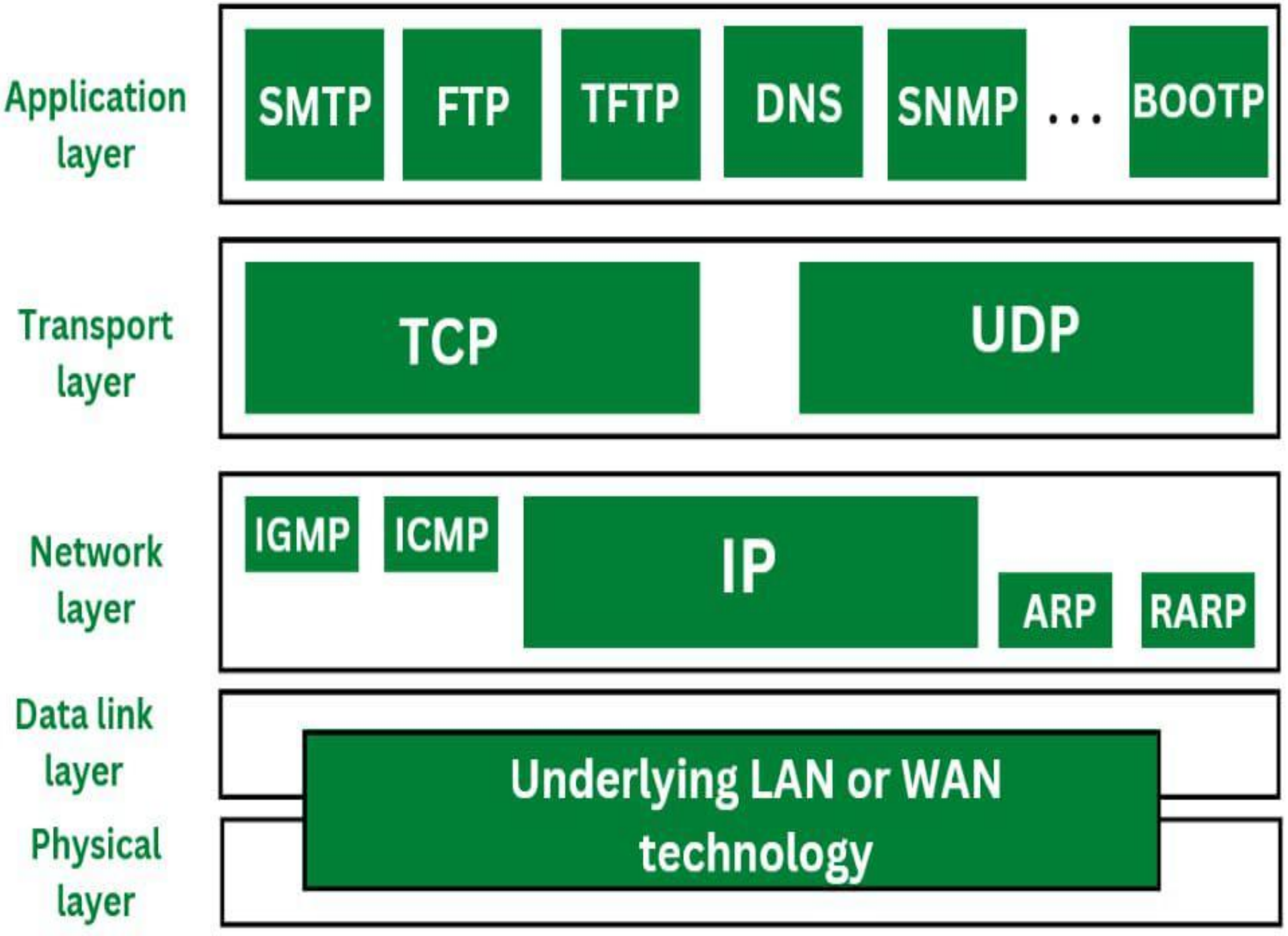
Connectionless: Each segment is considered as an independent packet and delivered to the transport layer at the destination machine.

4. Flow control: In this layer, flow control is performed end to end.

5. Error control: Error control is performed end to end in this layer to ensure that the complete message arrives at the receiving transport layer without any error. Error correction is done through retransmission.

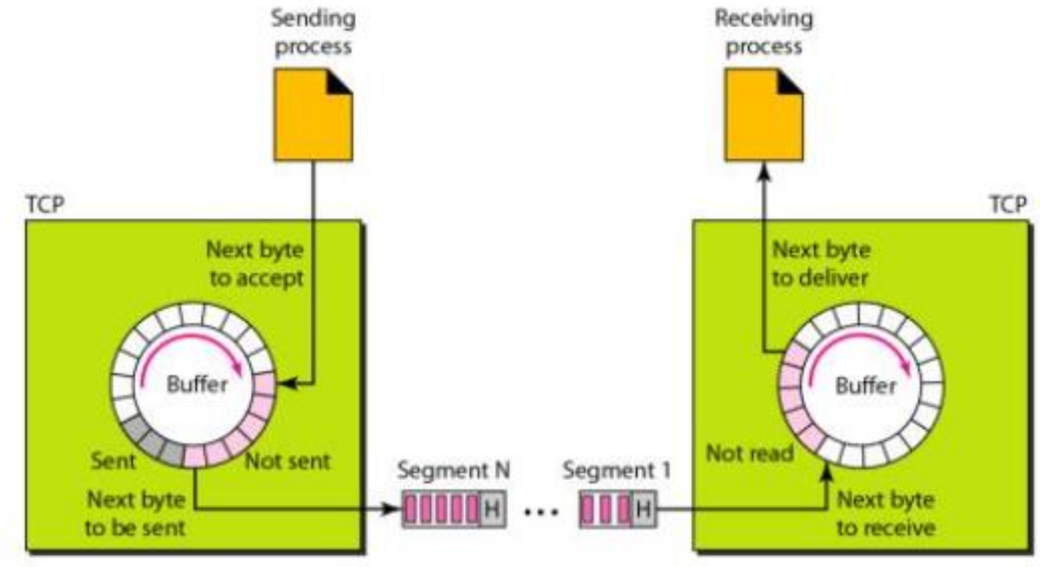
Transport and Network Layer Protocols

There are different transport/network layer protocols, but one family of protocols, the Internet Protocol Suite, dominates. Each transport and network layer protocol performs essentially the same functions, but each is incompatible with the others unless there is a special device to translate between them.



Transmission Control Protocol (TCP)

- TCP (Transmission Control Protocol) is a standard that defines how to establish and maintain a network conversation via which application programs can exchange data. TCP works with the Internet Protocol (IP), which defines how computers send packets of data to each other. Together. TCP and IP are the basic rules defining the Internet
- TCP is a connection-oriented protocol, which means a connection is established and maintained until the application programs at each end have finished exchanging messages. It determines how to break application data into packets that networks can deliver, sends packets to and accepts packets from the network layer, manages flow control, and because it is meant to provide error-free data transmission handles retransmission of dropped or garbled packets as well as acknowledgement of all packets that arrive.



- For example, when a Web server sends an HTML file to a client, it uses the HTTP protocol to do so. The HTTP program layer asks the TCP layer to set up the connection and send the file. The TCP stack divides the file into packets, numbers them and then forwards them individually to the IP layer for delivery. Although each packet in the transmission will have the same source and destination IP addresses, packets may be sent along multiple routes. The TCP program layer in the client computer waits until all of the packets have arrived, then acknowledges those it receives and asks for the retransmission on any it does not (based on missing packet numbers), then assembles them into a file and delivers the file to the receiving application.
- Operation of TCP protocol can be divided into 3 distinct sections:

  - Establishment of connection

  - Transmission of data

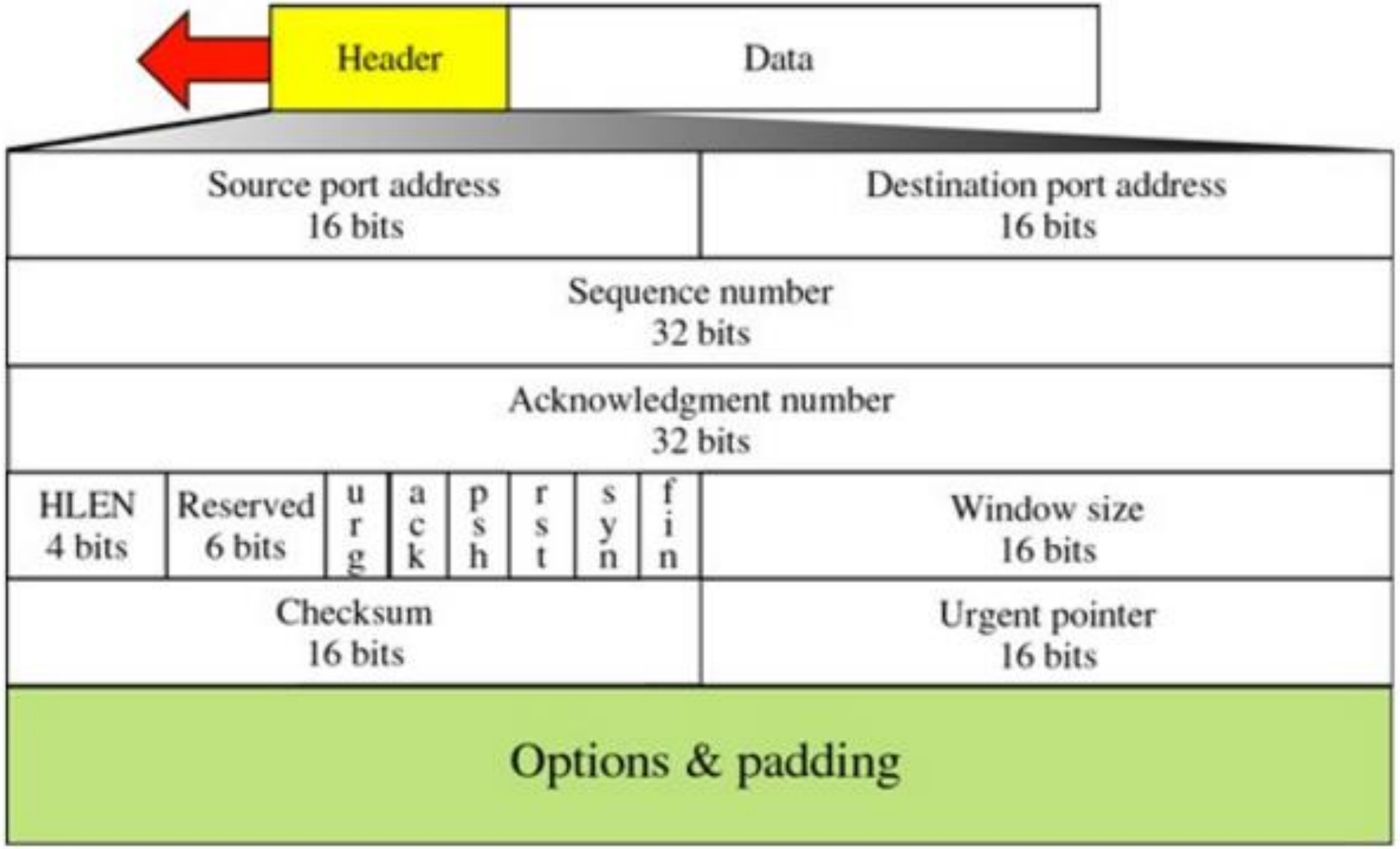
  - Termination of connection.

Thus, TCP is a full duplex, connection oriented, reliable and accurate protocol. TCP is very complicated and costly in terms of network overhead.

- TCP is used in non-time critical application where speed isn’t an essential factor. It is used in those process where reliable and sequential flow of data is required. Response and reliability are important factor for choosing TCP.

- TCP has several features that are briefly summarized as follows:
- **Stream Data transfer:** Applications working at the application layer transfers a continuous stream of bytes to the bottom layers. It is the duty of TCP to pack this byte stream to packets, known as TCP segments, which are passed to the IP layer for transmission to the destination device. The application does not have to bother to chop the byte stream data packets.
- **Reliability:** The most important feature of TCP is reliable data delivery. In order to provide reliability, TCP must recover from data that is damaged, lost, duplicated, or delivered out of order by the network layer. TCP assigns a sequence number to each byte transmitted, and expects a positive acknowledgment (ACK) from the receiving TCP layer. If the ACK is not received within a timeout interval, the data is retransmitted. The receiving TCP uses the sequence numbers to rearrange the TCP segments when they arrive out of order, and to eliminate duplicate TCP segments.

- **Flow control:** Network devices operate at different data rates because of various factors like CPU and available bandwidth. It may happen a sending device to send data at a much faster rate than the receiver can handle. TCP uses a sliding window mechanism for implementing flow control. The number assigned to a segment is called the sequence number and this numbering is actually done at the byte level. The TCP at the receiving device, when sending an ACK back to the sender, also indicates to the TCP at the sending device, the number of bytes it can receive without causing serious problems in its internal buffers.
- **Multiplexing:** Multitasking achieved through the use of port numbers.
- **Connections:** Before application processes can send data by using TCP, the devices must establish a connection. The connections are made between the port numbers of the sender and the receiver devices. A TCP connection identifies the end points involved in the connection. A socket number is a combination of IP address and port number, which can uniquely identify a connection.
- **Full duplex:** TCP provides for concurrent data streams in both directions.



Header

Data

Source port address
16 bits

Destination port address
16 bits

Sequence number
32 bits

Acknowledgment number
32 bits

HLEN
4 bits

Reserved
6 bits

urg

ack

psh

rst

syn

fin

Window size
16 bits

Checksum
16 bits

Urgent pointer
16 bits

Options & padding

1.Source port address:

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

2.Destination port address:

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

3. Sequence number (32-bit):

The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1. It puts the data back in the correct order or retransmits missing or damages data, a process called sequencing.

4.Acknowledgement number (32 bit):

Defines which TCP octet expected next. If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.

5. Header Length (4 bit):

Stands for header length, which defines the number of 32 bit words in the header. This indicates where the data begins. The TCP header (even one including options) is an integral number of 32 bits long.

6. Reserved (6 bits):

Reserved for future use, it always set to 0.

7. Control Bits (6 bits from left to right):

URG: Urgent Pointer field significant

ACK: Acknowledgment field significant

PSH: Push Function

RST: Reset the connection

SYN: Synchronize sequence numbers

FIN: No more data from sender

8. Window Size (16 bits):

The number of data octets beginning with the one indicated in the acknowledgement field which the sender of this segment is willing to accept.

9.Checksum (16 bits):

This field contains a checksum of the header. It actually uses a modified form of the header which includes some of the information from the IP header to detect some unusual types of errors.

10. Urgent Pointer:

It is only valid if the urgent flag is set, i.e. it is used when the segment contains urgent data. There is provision in TCP for some urgent messages to be sent bypassing the normal sequence.

11.Options and Padding:

There can be upto 40 bytes of optional information in the TCP header according to the need. It cab be padded with 0’s if the length is less

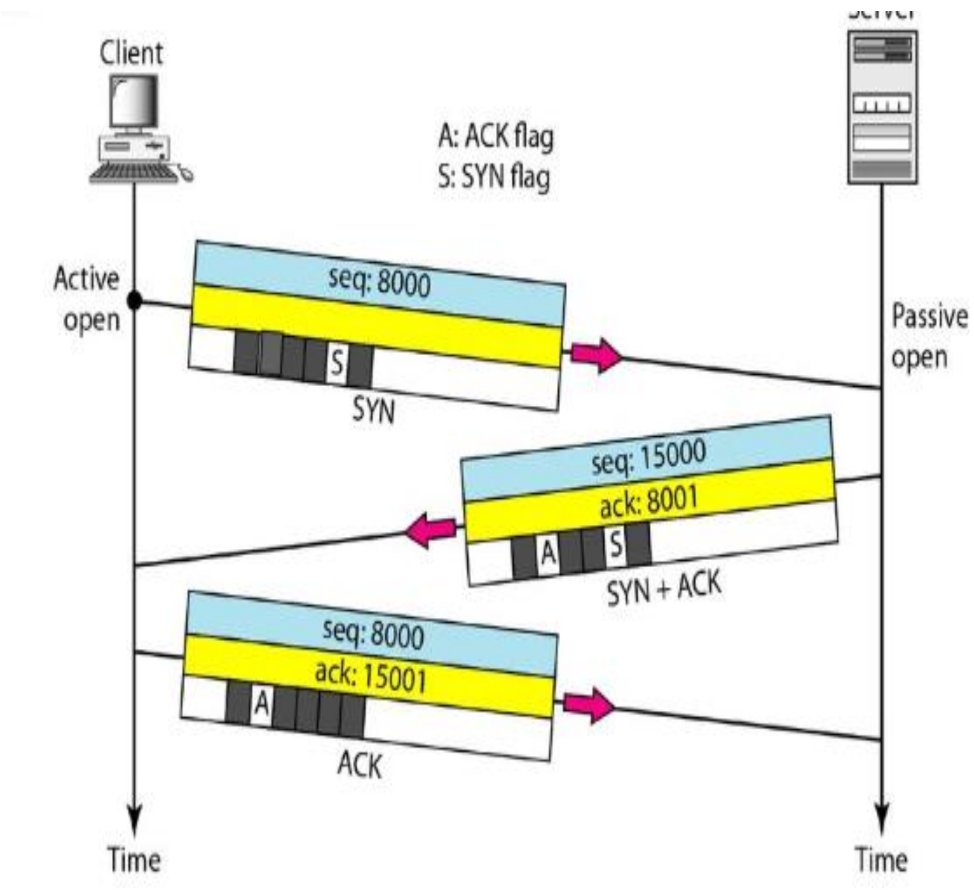
- **TCP Connection Establishment using Three-way Handshaking**
- TCP provides reliable communication with something called Positive Acknowledgement with Re- transmission (PAR). The Protocol Data Unit (PDU) of the transport layer is called segment. Now a device using PAR resend the data unit until it receives an acknowledgement. If the data unit received at the receiver's end is damaged (It checks the data with checksum functionality of the transport layer that is used for Error Detection), then receiver discards the segment. So, the sender has to resend the data unit for which positive acknowledgement is not received. You can realize from above mechanism that three segments are exchanged between sender (client) and receiver (server) for a reliable TCP connection to get established. Let's see how this mechanism works:

Step 1 (SYN): In the first step, client wants to establish a connection with server, so it sends a segment with SYN (Synchronize Sequence Number) which informs server that client is likely to start communication and with what sequence number it starts segments with

Step 2 (SYN + ACK): Server responds to the client request with SYN-ACK signal bits set. Acknowledgement (ACK) signifies the response of segment it received and SYN signifies with what sequence number it is likely to start the segments with

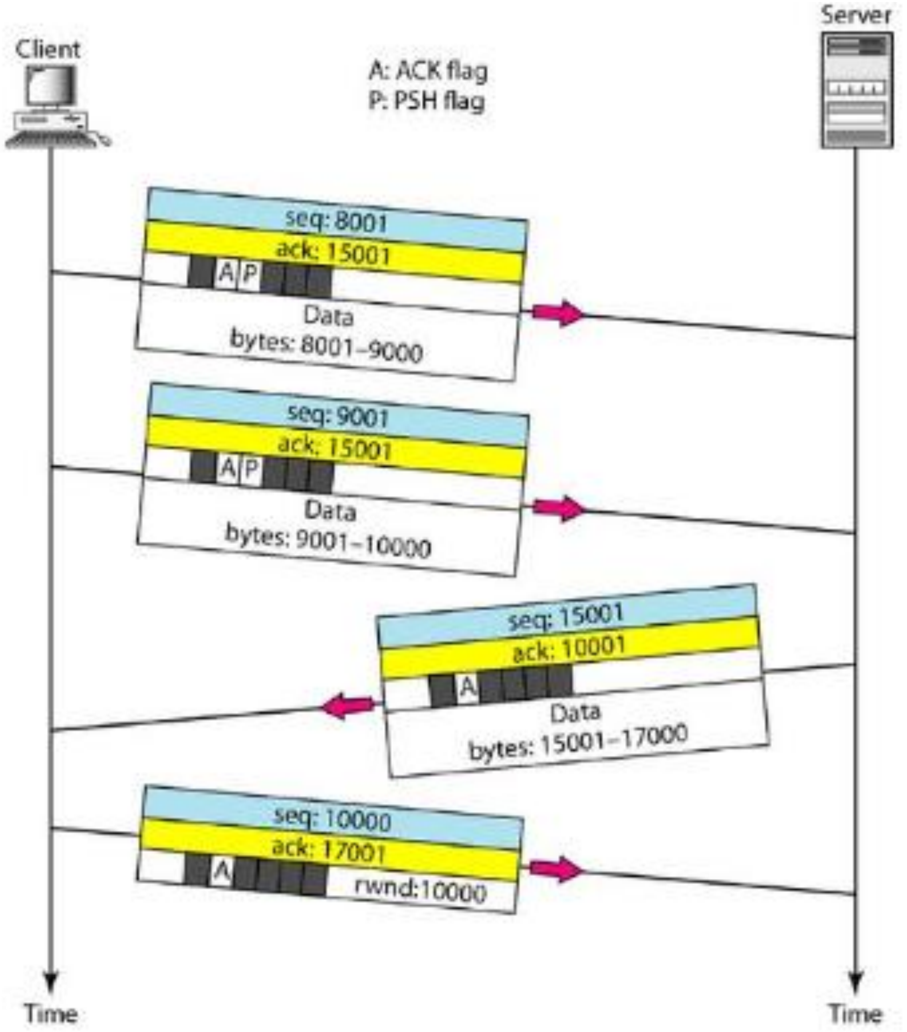
Step 3 (ACK): In the final part client acknowledges the response of server and they both establish a reliable connection with which they will start the actual data transfer.

The steps 1, 2 establish the connection parameter (sequence number) for one direction and it is acknowledged. The steps 2, 3 establish the connection parameter (sequence number) for the other direction and it is acknowledged. With these, a full-duplex communication is established.

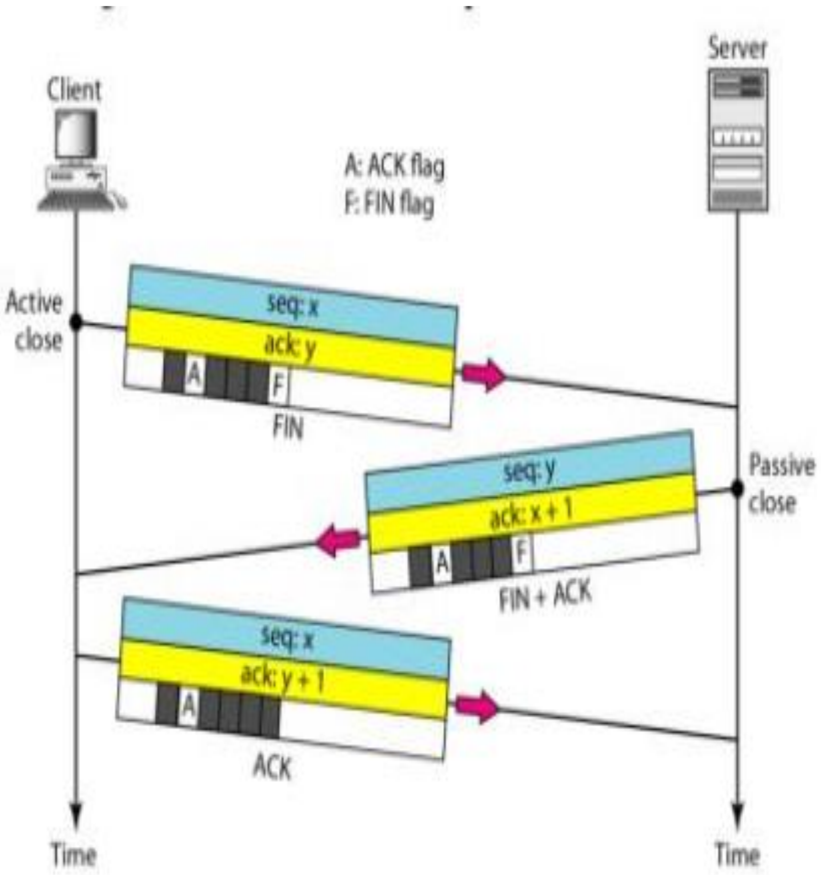


- **Data Transfer**
- Data are buffered by the transport entity on both transmission and reception. TCP normally exercises its own discretion as to when to construct a segment for transmission and when to release received data to the user. The PUSH flag is used to force the data so far accumulated to be sent by the transmitter and passed on by the receiver. This serves an end-of-block function.
- The user may specify a block of data as urgent. TCP will designate the end of that block with an urgent pointer and send it out in the ordinary data stream. The receiving user is alerted that urgent data are being received. If during data exchange, a segment arrives that is apparently not meant for the current connection, th RST flag is set on an outgoing segment. Examples of this situation are delayed duplicate SYNs and a acknowledgment of data not yet sent.

In above example, after a connection is established, the client sends 2,000 bytes of data in two segments. The server then sends 2,000 bytes in one segment. The client sends one more segment. The first three segments carry both data and acknowledgment, but the last segment carries only an acknowledgment because there is no more data to be sent. Note the values of the sequence and acknowledgment numbers. The data segments sent by the client have the PSH (push) flag set so that the server TCP tries to deliver data to the server process as soon as they are received. We discuss the use of this flag in more detail later. The segment from the server, on the other hand, does not set the push flag. Most TCP implementations have the option to set or not set this flag.



- **Connection Termination**
- Any of the two parties involved in exchanging data (client or server) can close the connection, although it is usually initiated by the client. Most implementations today allow two options for connection termination: three-way handshaking and four-way handshaking with a half-close option. Most implementations today allow three-way handshaking for connection termination as shown in Figure



1. In a common situation, the client TCP, after receiving a close command from the client process, sends the first segment, a FIN segment in which the FIN flag is set. Note that a FIN segment can include the last chunk of data sent by the client or it can be just a control segment as shown in the figure. If it is only a control segment, it consumes only one sequence number.

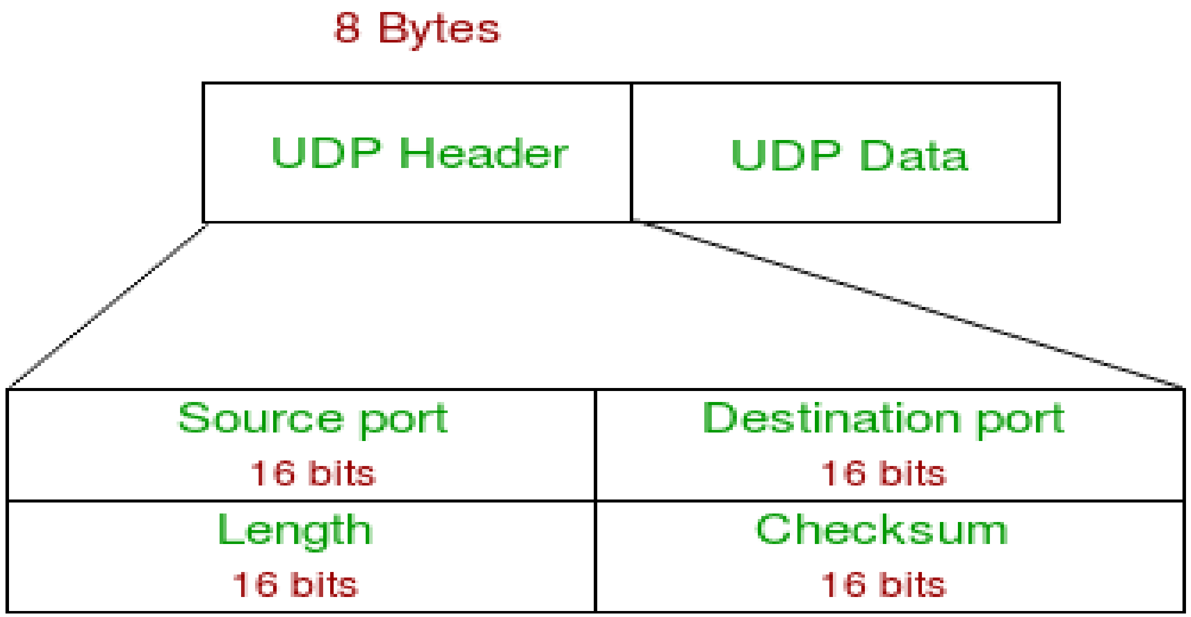
The server TCP, after receiving the FIN segment, informs its process of the situation and sends the second segment, a FIN+ACK segment, to confirm the receipt of the FIN segment from the client and at the same time to announce the closing of the connection in the other direction. This segment can also contain the last chunk of data from the server. If it doesnot carry data, it consumes only one sequence number.

The client TCP sends the last segment, an ACK segment, to confirm the receipt of the FIN segment from the TCP server. This segment contains the acknowledgment number, which is one plus the sequence number received in the FIN segment from the server. This segment cannot carry data and consumes no sequence numbers.

User Datagram Protocol (UDP)

- User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of Internet Protocol suite, referred as UDP/IP suite. Unlike TCP, it is unreliable and connectionless protocol. So, there is no need to establish connection prior to data transfer.
- For the real-time services like computer gaming, voice or video communication, live conferences; we need UDP. Since high performance is needed, UDP permits packets to be dropped instead of processing delayed packets. There is no error checking in UDP, so it also save bandwidth.
- User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

- **UDP Header:**
- UDP header is 8-bytes fixed and simple header. First 8 Bytes contains all necessary header information and remaining part consist of data. UDP port number fields are each 16 bits long, therefore range for port numbers defined from 0 to 65535; port number 0 is reserved. Port numbers help to distinguish different user requests or process.



8 Bytes

UDP Header	UDP Data

Source port 16 bits	Destination port 16 bits
Length 16 bits	Checksum 16 bits

1. Source Port : Source Port is 2 byte long field used to identify port number of source.
2. Destination Port : It is 2 byte long field, used to identify the port number of destination.
3. Length : It is 2 bytes long field which is the length of UDP including header and the data.
4. Checksum : Checksum is 2 bytes long field for error control.

- Notes – Unlike TCP, Checksum calculation is not mandatory in UDP. No Error control or flow control is provided by UDP. Hence UDP depends on IP and ICMP for error reporting.

UDP Operations:

The basic steps for transmission using UDP are:

- Higher-Layer Data Transfer: An application sends a message to the UDP software.
- UDP Message Encapsulation: The higher-layer message is encapsulated into the Data field of a UDP message. The headers of the UDP message are filled in, including the Source Port of the application that sent the data to UDP, and the Destination Port of the intended recipient. The checksum value may also be calculated.
- Transfer Message To IP: The UDP message is passed to IP for transmission.

On reception at the destination device this short procedure is reversed.

- **Applications of UDP:**
- Used for simple request response communication when size of data is less and hence there is lesser concern about flow and error control.
- It is suitable protocol for multicasting as UDP supports packet switching.
- UDP is used for some routing update protocols like RIP(Routing Information Protocol).
- Normally used for real time applications which can not tolerate uneven delays between sections of a received message.
- UDP takes datagram from Network Layer, attach its header and send it to the user. So, it works fast.
- Following implementations uses UDP as a transport layer protocol:

–DNS (Domain Name Service)

–DHCP

Internet Protocol (IP)

- The Internet Protocol (IP) provides a connectionless data transfer service over heterogeneous networks by passing and routing IP datagrams. IP datagram is essentially another name for a data packet. To be passed and routed on the Internet, all IP datagrams or packets that are passed down from the transport layer to the network layer (the connectionless packet delivery layer) are encapsulated with an IP header that contains the information necessary to transmit the packet from one network to another.
- There are currently two versions of IP: **IP version 4 (IPv4) and IP version 6 (IPv6).** IPv4 has been in existence since the early 1980s and is more than likely still the more common version. In order to support growing demands,
- IPv6 was introduced in 1998. IPv4 is being replaced by IPv6, which has a 320-bit header (40 bytes). The primary reason for the increase in the packet size is an increase in the address size from 32 bits to 128 bits. IPv6's simpler packet structure makes it easier to perform routing and supports a variety of new approaches to addressing and routing.
- Although modifications to the IP header represent profound changes in the protocol, there are even more significant differences between IPv4 and IPv6.

- Namely, IPv6 has:
- Better support for options using the extension headers
- Better security, with two extension headers devoted entirely to security
- More choices in type of service

IPv4

•An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet

•The IPv4 addresses are unique and universal.

•The address space of IPv4 is 2^32or 4,294,967,296

•IP address: 32-bit identifier for host, router interface

•interface: connection between host, router and physical link

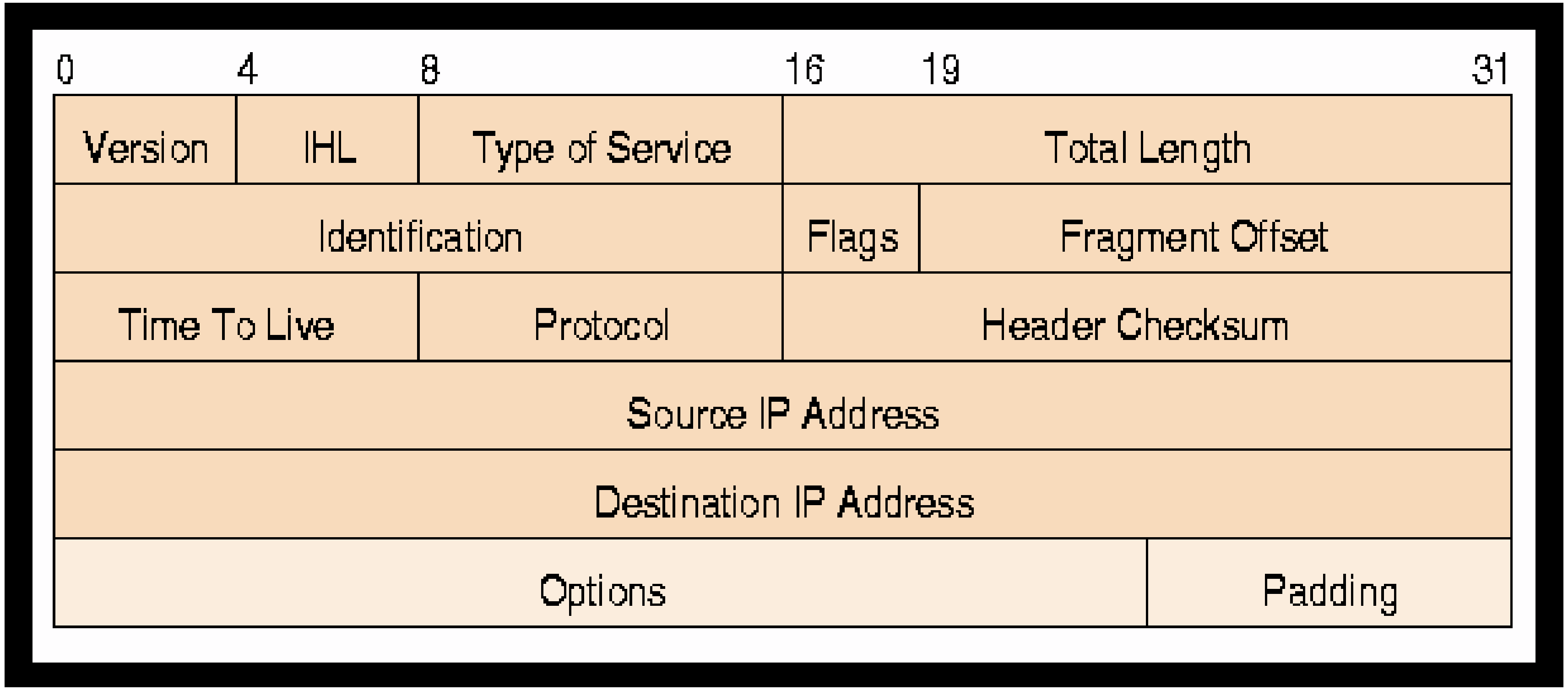
•router's typically have multiple interfaces

•host may have multiple interfaces

•IP addresses associated with interface, not host, or router.

•The IP address space is managed globally by the Internet Assigned Numbers Authority (IANA), and by five regional Internet registries (RIRs) responsible in their designated territories for assignment to end users and local Internet registries, such as Internet service providers

IPV4 FORMAT



0	4	8	16	19	31
Version	IHL	Type of Service	Total Length		
Identification			Flags	Fragment Offset	
Time To Live		Protocol	Header Checksum		
Source IP Address					
Destination IP Address					
Options					Padding

The header format is 22-60 bytes in length. It contains the necessary information for routing and delivery of data. Data field carries data for next layer.

1. Version (ver): It is 4 bits & keeps track of which version of protocol the datagram belongs. The current version is IPv4 and latest is IPv6.

2. Header length (HLEN): It is a 4 bit field that represents the size of a header. It rangers from 5 – 15 in value.

3. Service (8 bits): It determines how datagram should be handled. It is of two types which are **DS (Differentiated Service)** which is of 6 bits and consists of 3 bit precedence (priority) and other fields include delay (D), throughput (T), reliability (R) and **ECN (Explicit Congestion Notification)** which is of two bits and provides explicit signals for congestion.

4. Total length (16 bit): It defines the total length of an IP datagram including IP header and IP payload. It is usually written in octets.

5. Identification: It allows the destination host to determine which datagram a newly arrived fragment belongs to. It is primarily used for uniquely identifying fragments of an original IP datagram. If IP packet is fragmented during the transmission, all the fragments contain same identification number to identify original IP packet they belong to.

6. Flag (3 bits):

0	DF	MF

DF (Don’t fragment)
MF (More fragments)

As required by the network resources if IP packet is too large to handle these flag bits tell if they can be fragmented or not. It is used in fragmentation and reassembly.

7. Fragmentation offsets (13 bits): It is used to define where in the current datagram this fragment belongs. This offset tells the exact position of the fragment in the original IP Packet.

8. Time to live (TTL) (8 bits): It specifies how long in seconds a data gram is allowed to remain in the internet. The maximum life is 255 seconds.

9. Protocols (8 bits): It defines the next higher level protocols that is to receive the data frame at the destination. Example: TCP, UDP.

10. Header checksum (16 bits): It consists an error detecting code for header only because some header fields may alter during transmission.

11. Source and destination address (32 bits each): It denotes the network address of source and destination.

12. Options: It is used for future. The current defined options are security, source routing, route recording, string identification, time stamp.

13. Padding: It is used to ensure the datagram header is a multiple of 32 bits in length.

- **Notation of IP Address:**
- There are two common notations to show IP address

➢ Binary notation
➢ Dotted decimal notation

- **Binary Notation:**
- In binary notation, The IP address is displayed as 32-bits and this 32-bits is represented in 4-Octet (8bits) address or 4-bytes address. Example: 01110101. 10010101. 00011101. 11101010
- **Dotted-Decimal Notation:**
- To make the IP address more compact and easier to read, IP addresses are usually written in decimal form with separating each byte by a dot (.). Each number in dotted-decimal notation is between 0 and 255.
- For Example: 192.168.0.100

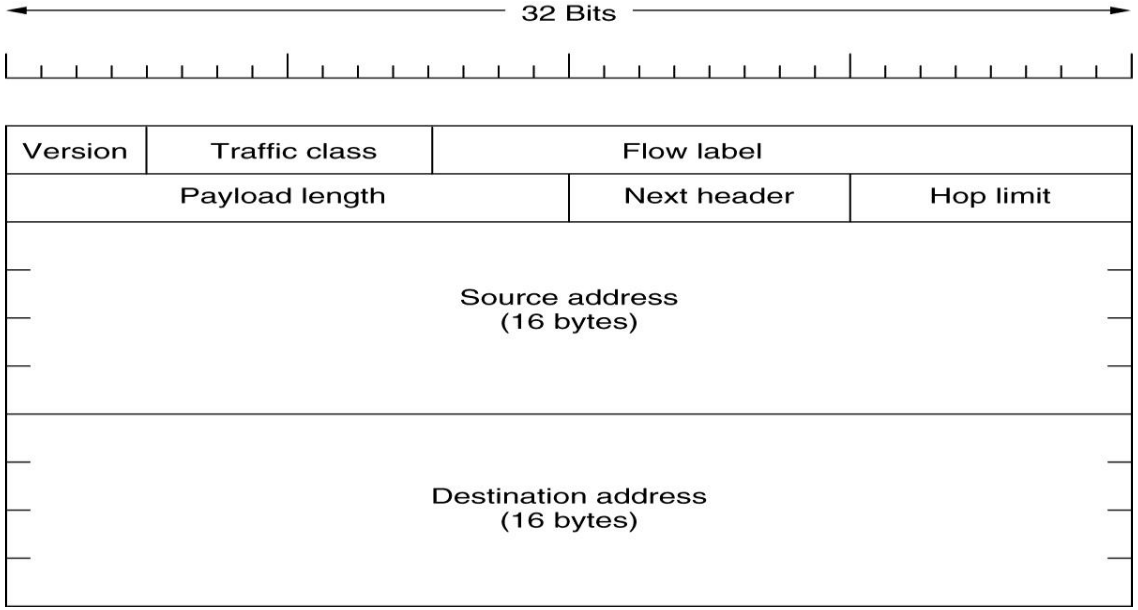
- **IP address**: It can be assigned in two ways:
    - Static
    - Dynamic
- **Static address**: It is manually inserted while joining the network. It is quite manageable as the user defines it, him/herself. It isn’t feasible in large organization.
- **Dynamic address**: Dynamic address are automatically assigned to a device. DHCP (Dynamic Host Configuration Protocol) assigns IP addresses to any new device in the network automatically. It’s quite feasible in large organization as a large amount of address can be assigned without manual intervention.

IPv6

- IPv4, defines a 32-bit address - 2^{32} (4,294,967,296) IPv4 addresses available
- The major problem with IPv4 is the eventual depletion of the IP address space with exponential growth in world population.
- Techniques like NAT and CIDR can only buy some more time for IPv4.
- IPv6 with very large address space has been introduced basically to eliminate the addressing problem
- IPv6 - 128 bits Network Layer Address
- Very large address space
    - 128-bit address => 2^128 ~ 3 Trillion trillion trillion
    - “Every grain of sand on earth can get Unique IPv6 address”

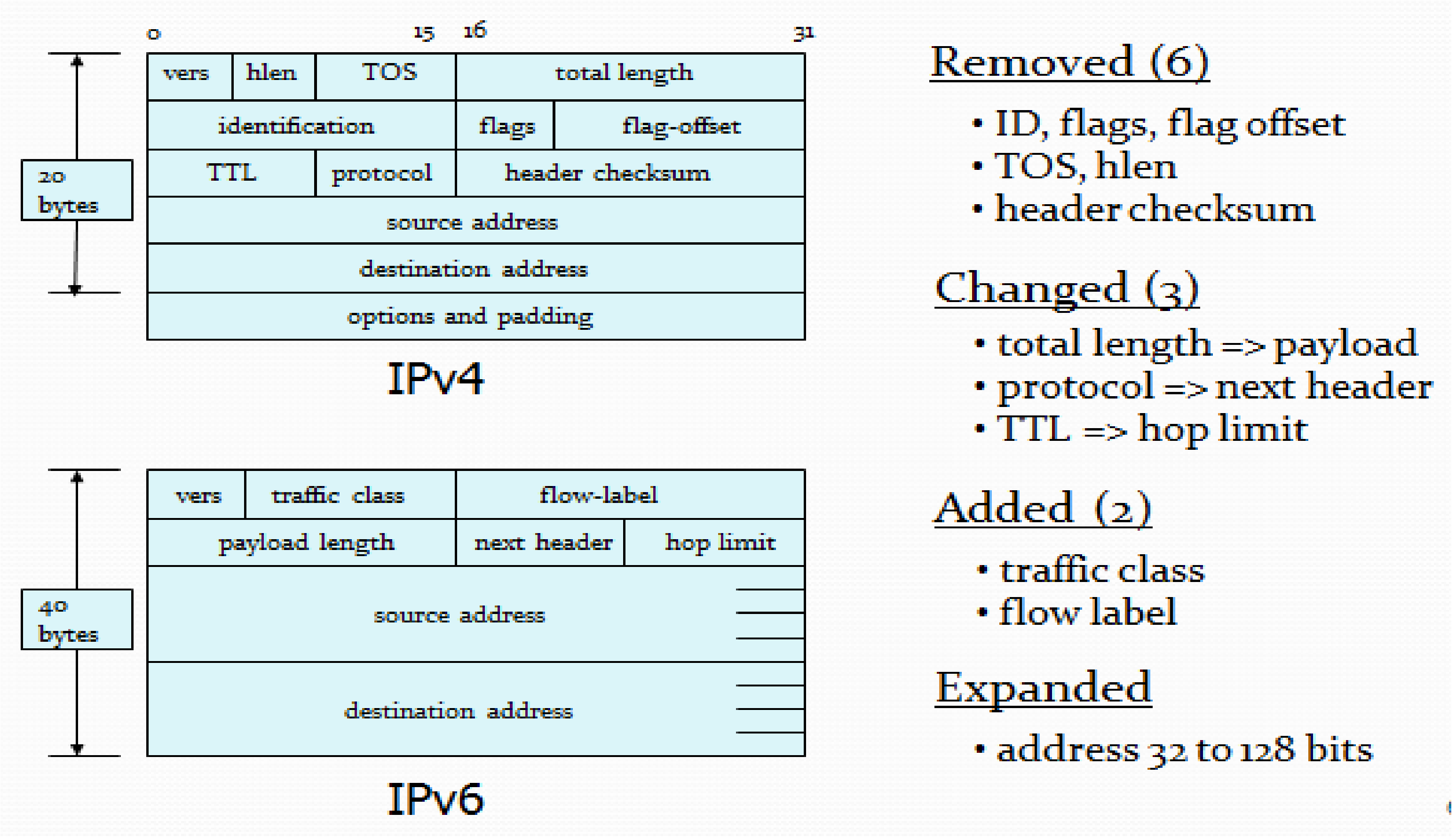
IPv6 Header Format

- **Version** (4 bits) - 4 bits are used to indicate the version of IP and is set to 6
- **Traffic Class** (8 bits) - Same function as the Type of Service field in the IPv4 , distinguish different real-time delivery requirement
- **Flow Label** (20 bits)
  - Identifies a flow and it is intended to enable the router to identify packets that should be treated in a similar way without the need for deep lookups within those packets.
  - Set by the source and should not be changed by routers along the path to destination.
  - Unique & powerful tool to IPv6



- **Payload Length** (16 bits) – Only the length of the payload (Header length is fixed to 40 bytes)
- **Next Header** (8 bits) - Indicates either the first extension header (if present) or the protocol in the upper layer PDU (such as TCP, UDP, or ICMPv6).
- **Hop Limit** (8 bits) - IPv4 TTL was appropriately renamed Hop Limit because it is a variable that is decremented at each hop, and it does not have a temporal dimension.
- **Source Address** (128 bits) - Stores the IPv6 address of the originating host.
- **Destination Address** (128 bits) - Stores the IPv6 address of the current destination host.

Header Comparison



IPv4

IPv6

Removed (6)

- ID, flags, flag offset
- TOS, hlen
- header checksum

Changed (3)

- total length => payload
- protocol => next header
- TTL => hop limit

Added (2)

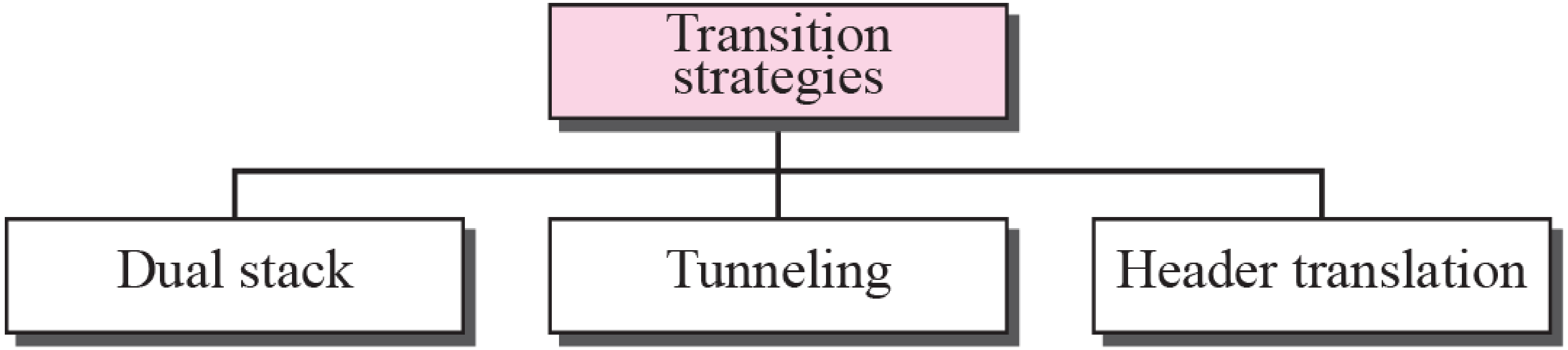
- traffic class
- flow label

Expanded

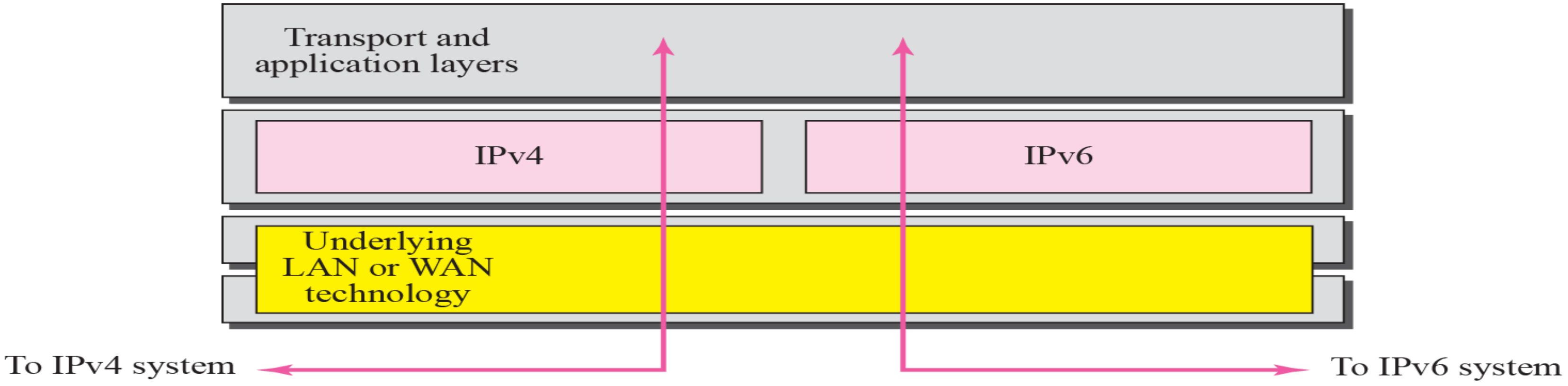
- address 32 to 128 bits

- **IPv4 to IPv6 Transition**

- Because of the huge number of systems on the Internet, the transition from IPv4 to IPv6 cannot happen suddenly.
- It will take a considerable amount of time before every system in the internet can move from IPv4 to IPv6
- The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.
- Three strategies to help the transition :

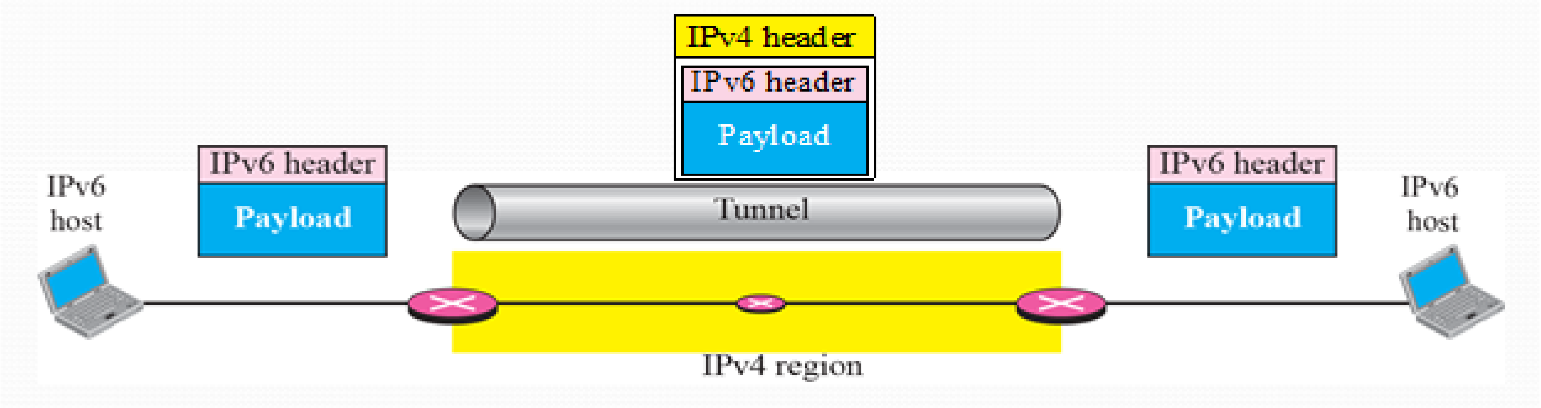


- **Dual Stack**
- Stations run both IPv6 and IPv4 simultaneously until all the Internet uses IPv6
- For received packets, uses version field to decide which stack(v4 or v6) to use
- To determine which version to use when sending a packet to a destination, the source host queries the DNS. If the DNS returns an IPv4 address, the source host sends an IPv4 packet. If the DNS returns an IPv6 address, the source host sends an IPv6 packet.


Transport and application layers
IPv4
IPv6
Underlying LAN or WAN technology
To IPv4 system
To IPv6 system

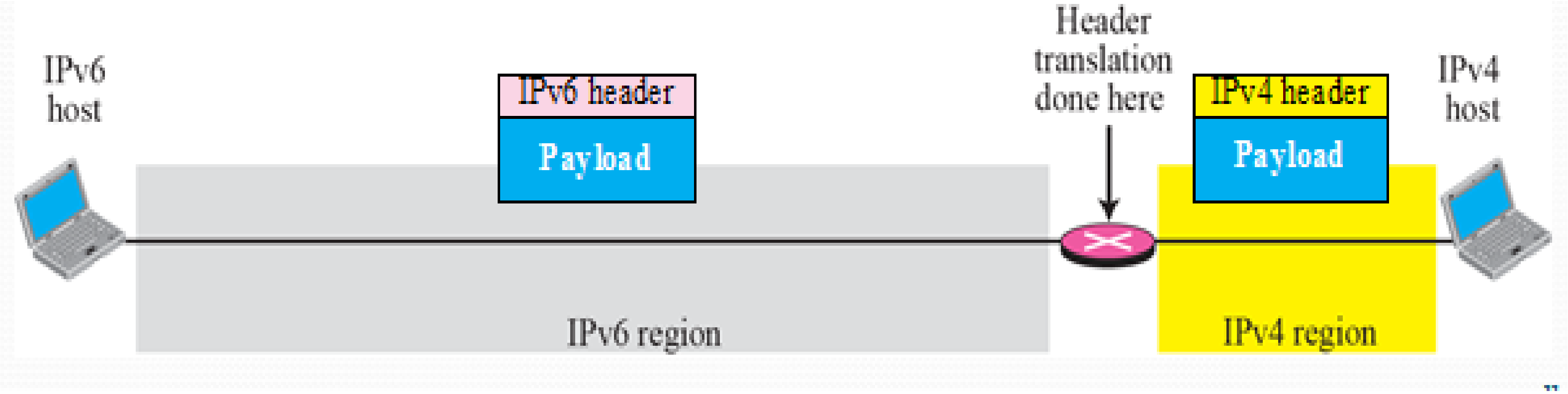
- **Tunnelling**

- Tunneling is used to deal with networks where v4 router(s) sit between two v6 routers
  - Simply encapsulate v6 packets and all of their information in v4 packets and leaves its capsule when it exist the region.



- **Header Translation**

- Header Translation is necessary when the majority of the internet has moved to IPv6 but some still use IPv4.
- Converting IPv6 Header to IPv4 and vice versa
- Used to link IPv6 and IPv4 Networks
- Some header fields may lost during translation
  - Eg. When converting IPv6 Packet to IPv4, the header information like *Flow-label* gets lost



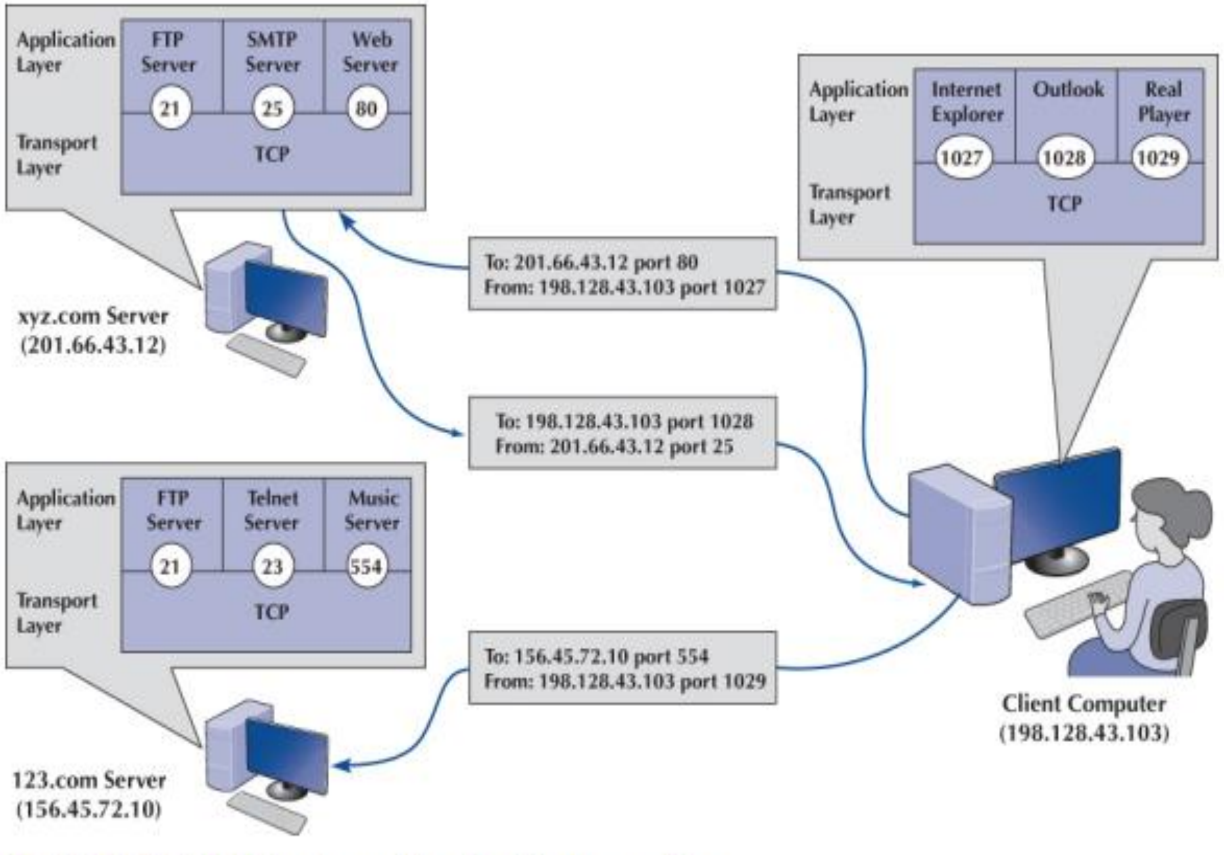
Transport Layer Functions

- **The transport layer links the application software** in the application layer with the network and is responsible for segmenting large messages into smaller ones for transmission and for managing the session (the end-to-end delivery of the message). One of the first issues facing the application layer is to find the numeric network address of the destination computer.
- Different protocols use different methods to find this address. Depending on the protocol—and which expert you ask—finding the destination address can be classified as a transport layer function, a network layer function, a data link layer function, or an application layer function with help from the operating system. T
- Three unique functions performed by the transport layer: linking the application layer to the network, segmenting and session management.

- **Linking to the Application Layer**
- **Most computers have many application** layer software packages running at the same time. Users often have Web browsers, e-mail programs, and word processors in use at the same time on their client computers. Likewise, many servers act as Web servers, mail servers, FTP servers, and so on. When the transport layer receives an incoming message, the transport layer must decide to which application program it should be delivered. It makes no sense to send a Web page request to e-mail server software.
- **With TCP/IP**, each application layer software package has a unique port address. Any message sent to a computer must tell TCP (the transport layer software) the application layer port address that is to receive the message. Therefore, when an application layer program generates an outgoing message, it tells the TCP software its own port address (i.e., the source port address) and the port address at the destination computer (i.e., the destination port address). These two port addresses are placed in the first two fields in the TCP segment.

- **Port addresses can be any 16-bit (2-byte) number**. So, how does a client computer sending a Web request to a Web server know what port address to use for the Web server? Simple. On the Internet, all port addresses for popular services such as the Web, email, and FTP have been standardized. Anyone using a Web server should set up the Web server with a port address of 80, which is called the well-known port. Web browsers, therefore, automatically generate a port address of 80 for any Web page you click on. FTP servers use port 21, Telnet 23, SMTP 25, and so on. Network managers are free to use whatever port addresses they want, but if they use a nonstandard port number, then the application layer software on the client must specify the correct port number.

Figure shows a user running three applications on the client (Internet Explorer, Outlook, and RealPlayer), each of which has been assigned a different port number, called a temporary port number (1027, 1028, and 1029, respectively). Each of these can simultaneously send and receive data to and from different servers and different applications on the same server. In this case, we see a message sent by Internet Explorer on the client (port 1027) to the Web server software on the xyz.com server (port 80). We also see a message sent by the mail server software on port 25 to the email client on port 1028. At the same time, the RealPlayer software on the client is sending a request to the music server software (port 554) at 123.com.



Linking to application layer services

- **Segmenting**
- **Some messages or blocks of application** data are small enough that they can be transmitted in one frame at the data link layer. However, in other cases, the application data in one "message" is too large and must be broken into several frames (e.g., Web pages, graphic images). As far as the application layer is concerned, the message should be transmitted and received as one large block of data. However, the data link layer can transmit only messages of certain lengths. It is therefore up to the sender’s transport layer to break the data into several smaller segments that can be sent by the data link layer across the circuit. At the other end, the receiver’s transport layer must receive all these separate segments and recombine them into one large message.
- **Segmenting means to take one outgoing** message from the application layer and break it into a set of smaller segments for transmission through the network. It also means to take the incoming set of smaller segments from the network layer and reassemble them into one message for the application layer. Depending on what the application layer software chooses, the incoming packets can either be delivered one at a time or held until all packets have arrived and the message is complete.

- **Web browsers, for example,** usually request delivery of packets as they arrive, which is why your screen gradually builds a piece at a time. Most e-mail software, on the other hand, usually requests that messages be delivered only after all packets have arrived and TCP has organized them into one intact message, which is why you usually don’t see e-mail messages building screen by screen.
- The TCP is also responsible for ensuring that the receiver has actually received all segments that have been sent. TCP therefore uses continuous ARQ.
- **One of the challenges at the transport layer** is deciding how big to make the segments. Remember, we discussed packet sizes in next topic. When transport layer software is set up, it is told what size segments it should use to make best use of its own data link layer protocols (or it chooses the default size of 536). However, it has no idea what size is best for the destination. Therefore, the transport layer at the sender negotiates with the transport layer at the receiver to settle on the best segment sizes to use. This negotiation is done by establishing a TCP connection between the sender and receiver.

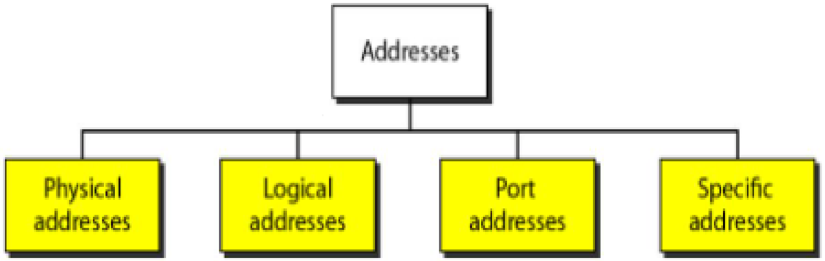
- **Session Management**
- **A session can be thought of as a conversation** between two computers. When the sending computer wants to send a message to the receiver, it usually starts by establishing a session with that computer. The sender transmits the segments in sequence until the conversation is done, and then the sender ends the session. This approach to session management is called connection-oriented messaging.
- **Sometimes,** the sender only wants to send one short information message or a request. In this case, the sender may choose not to start a session, but just send the one quick message and move on. This approach is called connectionless messaging.
- **Connection-Oriented** Messaging Connection-oriented messaging sets up a TCP connection (also called a session) between the sender and receiver. The transport layer software sends a special segment (called a SYN) to the receiver, requesting that a session be established. The receiver either accepts or rejects the session, and together they settle on the segment sizes the session will use.
- Once the connection is established, the segments flow between the sender and receiver. TCP uses the continuous ARQ (sliding window) technique described in next topic to make sure that all segments arrive and to provide flow control.

- **When the transmission is complete,** the sender sends a special segment (called a FIN) to close the session. Once the sender and receiver agree, the session is closed and all record of it is deleted.
- **Connectionless Messaging** Connectionless messaging means each packet is treated separately and makes its own way through the network. Unlike connection-oriented routing, no connection is established. The sender simply sends the packets as separate, unrelated entities, and it is possible that different packets will take different routes through the network, depending on the type of routing used and the amount of traffic. Because packets following different routes may travel at different speeds, they may arrive out of sequence at their destination. The sender’s network layer, therefore, puts a sequence number on each packet, in addition to information about the message stream to which the packet belongs. The network layer must reassemble them in the correct order before passing the message to the application layer.
- **TCP/IP can operate either** as connection-oriented or connectionless. When connection-oriented is desired, TCP is used. When connectionless is desired, the TCP segment is replaced with a User Datagram Protocol (UDP) packet. The UDP packet is much smaller than the TCP packet (only eight bytes).

- **Connectionless is most commonly used** when the application data or message can fit into one single message. One might expect, for example, that because HTTP requests are often very short, they might use UDP connectionless rather than TCP connection-oriented messaging. However, HTTP always uses TCP. All of the application layer software we have discussed so far uses TCP (HTTP, SMTP, FTP, Telnet). UDP is most commonly used for control messages such as addressing (DHCP [Dynamic Host Configuration Protocol]), routing control messages (RIP [Routing Information Protocol], discussed later in this topic), and network management.
- **Quality of Service (QoS)** routing is a special type of connection-oriented messaging in which different connections are assigned different priorities. For example, videoconferencing requires fast delivery of packets to ensure that the images and voices appear smooth and continuous; they are very time dependent because delays in routing seriously affect the quality of the service provided. E-mail packets, on the other hand, have no such requirements. Although everyone would like to receive e-mail as fast as possible, a 10-second delay in transmitting an e-mail message does not have the same consequences as a 10-second delay in a videoconferencing packet.

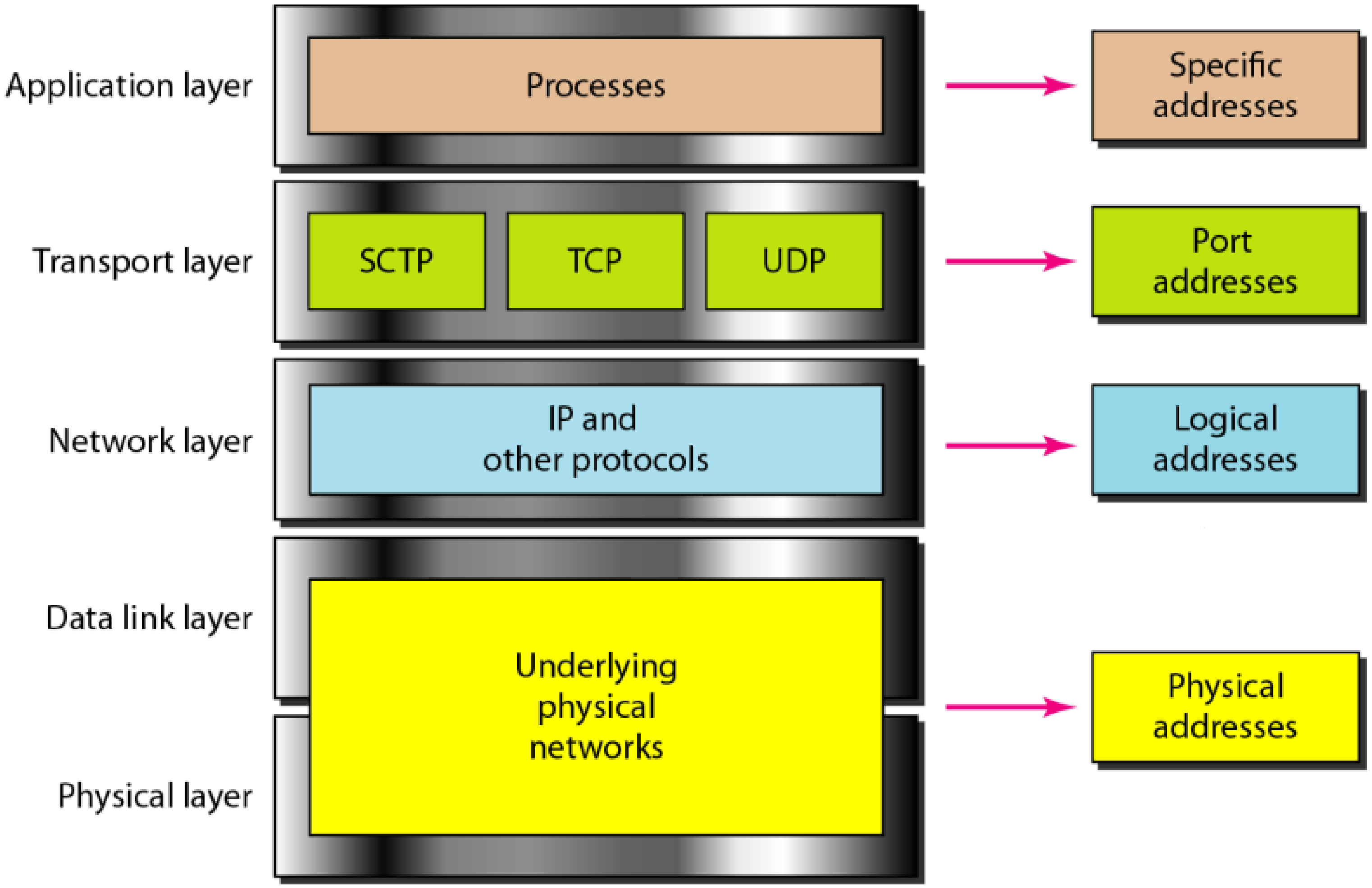
Addressing

- Four levels of addresses are used in an internet employing the TCP/IP protocols: physical, logical, port and specific.
- **Physical Addresses**
- The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN.
- The size and format of these addresses vary depending on the network. For example, Ethernet uses a 6-byte (48-bit) physical address.
- **Logical Addresses**
- Logical addresses are used by networking software to allow packets to be independent of the physical connection of the network, that is, to work with different network topologies and types of media.
- A logical address in the Internet is currently a 32-bit address that can uniquely define a host connected to the Internet. An internet address in IPv4 in decimal numbers **132.24.75.9**



- **Port Addresses**
- There are many application running on the computer. Each application run with a port no.(logically) on the computer.
- A port number is part of the addressing information used to identify the senders and receivers of messages.
- **Application-Specific Addresses**
- Some applications have user-friendly addresses that are designed for that specific application.
- Examples include the e-mail address (for example, forouzan@fhda.edu) and the Universal Resource Locator (URL) (for example, www.mhhe.com). The first defines the recipient of an e-mail; the second is used to find a document on the World Wide Web.

Assigning Addresses



- Figure shows the addressing at each layer. As the figure shows, there is a relationship between the layer, the address used in that layer, and the packet name at that layer. At the application layer, we normally use names to define the site that provides services, such as facebook.com, or the e-mail address, such as somebody@gmail.com. At the transport layer, addresses are called port numbers, and these define the application-layer programs at the source and destination. Port numbers are local addresses that distinguish between several programs running at the same time. At the network-layer, the addresses are global, with the whole Internet as the scope. A network-layer address uniquely defines the connection of a device to the Internet. The data link-layer addresses, sometimes called MAC addresses, are locally defined addresses, each of which defines a specific host or router in a network (LAN or WAN).

IPv4 Addressing

- An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet
- The IPv4 addresses are unique and universal.
- The address space of IPv4 is 232 or 4,294,967,296
- IP address: 32-bit identifier for host, router interface

- interface: connection between host, router and physical link

- router's typically have multiple interfaces

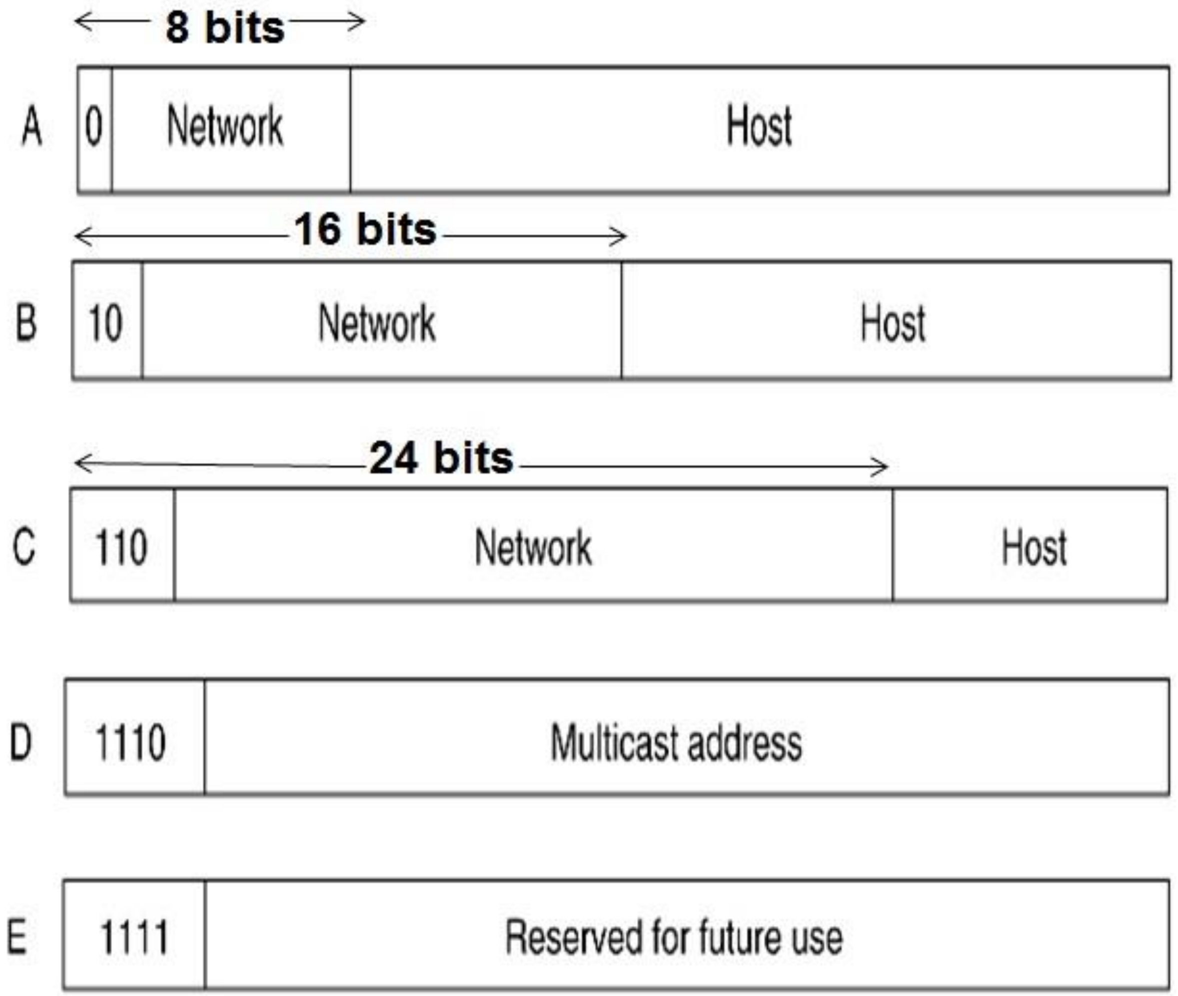
- host may have multiple interfaces

- IP addresses associated with interface, not host, or router.
- The IP address space is managed globally by the Internet Assigned Numbers Authority (IANA), and by five regional Internet registries (RIRs) responsible in their designated territories for assignment to end users and local Internet registries, such as Internet service providers

- IP address has two parts:
- The network prefix identifies a network and the host number identifies a specific host (actually, interface on the network)
- network part :high order bits
- host part :low order bits Within the IPv4 address range , there are three types of addresses:
- Network Address - The address by which we refer to the network.
- Broadcast Address - A special address used to send data to all hosts in the network.
- Host Address - The addresses assigned to the end devices in the network.

Classful Addressing (IP Address Classes):

- In classful addressing, the address space (i.e.32-bits) is divided into five classes: A, B, C, D, and E. A new architecture that was introduced in mid-1990 and proposed a concept of classless and such addressing is called classless addressing. However most of the Internet is still using classful addressing. In classful addressing IP address space is divided into five classes:
- Class A: The range of Class A IP address is 1.0.0.0 to 127.255.255.255
- Class B: The range of Class B IP address is 128.0.0.0 to 191.255.255.255
- Class C: The range of Class C IP address is 192.0.0.0 to 223.255.255.255
- Class D: The range of Class D IP address is 224.0.0.0 to 239.255.255.255
- Class E: The range of Class E IP address is 240.0.0.0 to 255.255.255.255
- These classes and their ranges can be shown in the following fig:



8 bits

A 0 Network Host

1.0.0.0 to
127.255.255.255

16 bits

B 10 Network Host

128.0.0.0 to
191.255.255.255

24 bits

C 110 Network Host

192.0.0.0 to
223.255.255.255

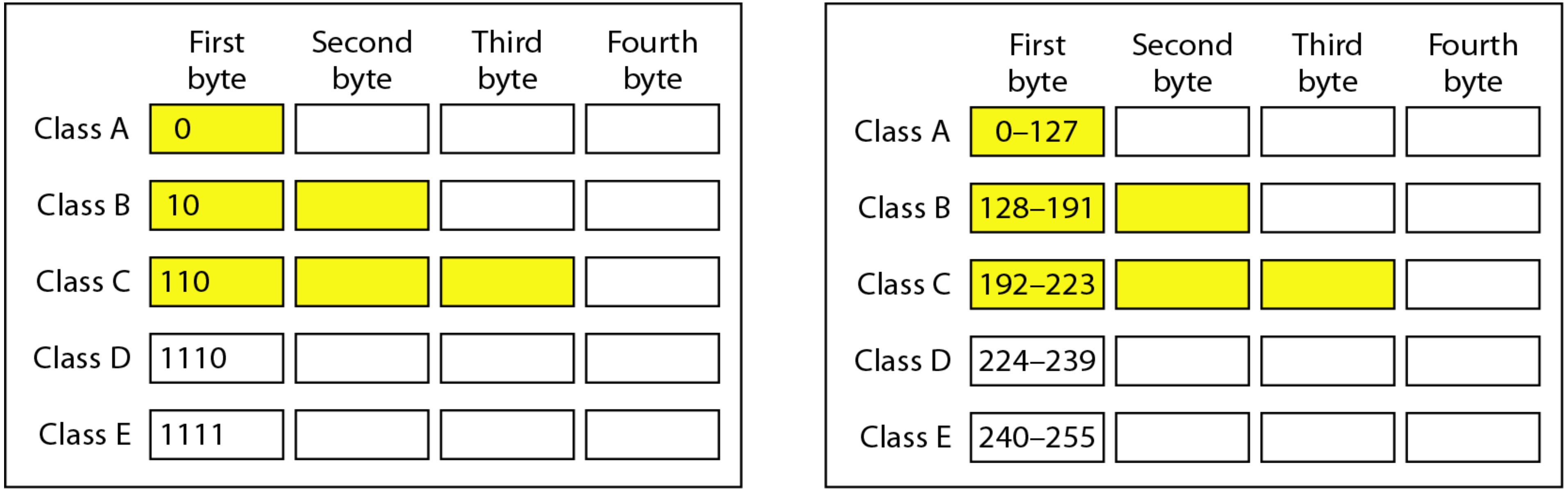
D 1110 Multicast address

224.0.0.0 to
239.255.255.255

E 1111 Reserved for future use

240.0.0.0 to
255.255.255.255

- In classful addressing, a large part of the available addresses were wasted.



	First byte	Second byte	Third byte	Fourth byte
Class A	0			
Class B	10	▨		
Class C	110	▨	▨	
Class D	1110			
Class E	1111			

a. Binary notation

	First byte	Second byte	Third byte	Fourth byte
Class A	0–127			
Class B	128–191	▨		
Class C	192–223	▨	▨	
Class D	224–239			
Class E	240–255			

b. Dotted-decimal notation

Class A Address

- The first bit of the first octet is always set to zero. So that the first octet ranges from 1 –127.
- The class A address only include IP starting from 1.x.x.x to 126.x.x.x. The IP range 127.x.x.x is reserved for loop back IP addresses.
- The default subnet mask for class A IP address is 255.0.0.0. **This means it can have 126 networks (2^7-2) and 16777214 hosts (2^{24}-2)**. (Minus 2 because 2 addresses are reserved for network and broadcast address)
- Class A IP address format is thus: **0NNNNNNN.HHHHHHHH.HHHHHHHH.HHHHHHHH**.

Class B Address

- Here the first two bits in the first two bits is set to zero.
- Class B IP Addresses range from 128.0.x.x to 191.255.x.x.
- The default subnet mask for Class B is 255.255.x.x. **Class B has 16384 (2^{14}) Network addresses and 65534 (2^{16}-2)**
- Host addresses. Class B IP address format is: **10NNNNNN.NNNNNNNN.HHHHHHHH.HHHHHHHH**

Class C Address

•The first octet of this class has its first 3 bits set to 110. Class C IP addresses range from 192.0.0.x to 223.255.255.x.

•The default subnet mask for Class C is 255.255.255.x. **Class C gives 2097152 (2^{21}) Network addresses and 254 (2^{8}-2)**

•Host addresses. Class C IP address format is: **110NNNNN.NNNNNNNN.NNNNNNNN.HHHHHHHH**

Class D Address

•The first four bits of the first octet in class D IP address are set to 1110.

•Class D has IP address rage from 224.0.0.0 to 239.255.255.255.

•Class D is reserved for Multicasting. In multicasting data is not intended for a particular host, but multiple ones. That is why there is no need to extract host address from the class D IP addresses.

•The Class D does not have any subnet mask.

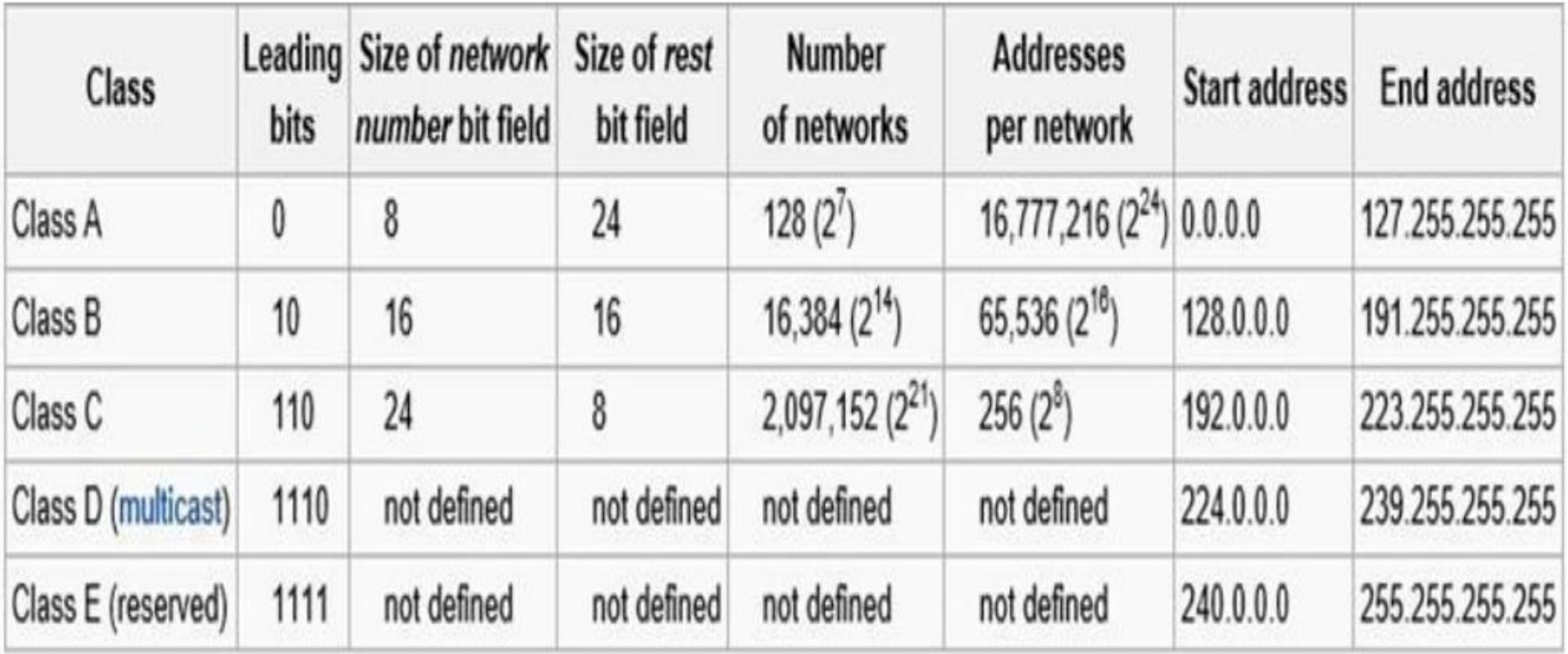
Class E Address

•The class E IP addresses are reserved for experimental purpose only for R&D or study.

•IP addresses in the class E ranges from 240.0.0.0 to 255.255.255.254.

•This class too is not equipped with any subnet mask.

Summary



Class	Leading bits	Size of *network number* bit field	Size of *rest* bit field	Number of networks	Addresses per network	Start address	End address
Class A	0	8	24	128 (2^{7})	16,777,216 (2^{24})	0.0.0.0	127.255.255.255
Class B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	128.0.0.0	191.255.255.255
Class C	110	24	8	2,097,152 (2^{21})	256 (2^{8})	192.0.0.0	223.255.255.255
Class D (multicast)	1110	not defined	not defined	not defined	not defined	224.0.0.0	239.255.255.255
Class E (reserved)	1111	not defined	not defined	not defined	not defined	240.0.0.0	255.255.255.255

- **Netid (or network part) and Hostid (or host part)**
- In classful addressing an IP address in classes A, B, and C is divided into netid and hostid. These parts are varying length depending on the class of the address.

➢ **Netid (network part):** The portion of the IP address that identifies the network is called the netid. The network part of IP address identifies the network to which the host is attached. All the hosts attached to the same network part (netid) in their IP address.

➢ **Hostid:** The portion of the IP address that identifies the host or router on the network is called the hostid. The host part identifies each host uniquely on that particular network.

- **Network Address**
- The network address is an address that defines the network itself; it cannot be assigned to a host. Network address plays a very important role in classful addressing. A network address has several properties:

➢ All hostid bytes are 0s

➢ The network address defines the network to the rest of internet. Latter we learn that router can route a packet on the network address

➢ Network address is first address in the block

➢ Given the network address we can find the class of the address

- **Network address is different from Netid . A network address has both netid and hostid with 0s for hostid.**

- **Reserved IP Address**
- Certain IP addresses are reserved and cannot be assigned to device on a network.
  - The addresses in class E are reserved for research purpose.
  - The network address is reserved: all 0s in hostid of network (network address is used to identify the network itself)
  - Broadcast address is reserved: all 1 in hostid of network (broadcast address is used for broadcasting packets to all the devices on a network)
  - In class A IP address 127.xx.yy.zz is reserved for internal loop-back
- **Private IP address**
- Some addresses are allocated to be used for private networks- called private IP address
  - Class A: 10.0.0.0 to 10.255.255.255
  - Class B: 172.16.0.0 to 172.31.255.255
  - Class C: 192.168.0.0 to 192.168.255.255

- **Classless Addressing**
- To reduce the wastage of IP addresses in a block, we use sub-netting. What we do is that we use host id bits as net id bits of a classful IP address.
- •We give the IP address and define the number of bits for mask along with it (usually followed by a ‘/’ symbol), like, 192.168.1.1/28.
- •Here, subnet mask is found by putting the given number of bits out of 32 as 1, like, in the given address, we need to put 28 out of 32 bits as 1 and the rest as 0, and so, the subnet mask would be 255.255.255.240.

Classless Inter Domain Routing (CIDR)

•CIDR is a new addressing scheme for the Internet which allows for more efficient allocation of IP addresses than the old Class A, B, and C address scheme.

•CIDR allowed for more efficient use of IPv4 address space and prefix aggregation, known as route summarization or supernetting.

CIDR introduction allowed for:

•More efficient use of IPv4 address space

•Prefix aggregation, which reduced the size of routing tables

CIDR allows routers to group routes together to reduce the bulk of routing information carried by the core routers. With CIDR, several IP networks appear to networks outside the group as a single, larger entity.

CIDR is based on variable-length subnet masking (VLSM). This allows it to define prefixes of arbitrary lengths making it much more efficient than the old system. CIDR IP addresses are composed of two sets of numbers.

With CIDR, IP addresses and their subnet masks are written as four octets, separated by periods, followed by a forward slash and a two-digit number that represents the subnet mask e.g.

- 10.1.1.0/30
- 172.16.1.16/28
- 192.168.1.32/27etc
- IP 10.0.0.0/24 equivalent to

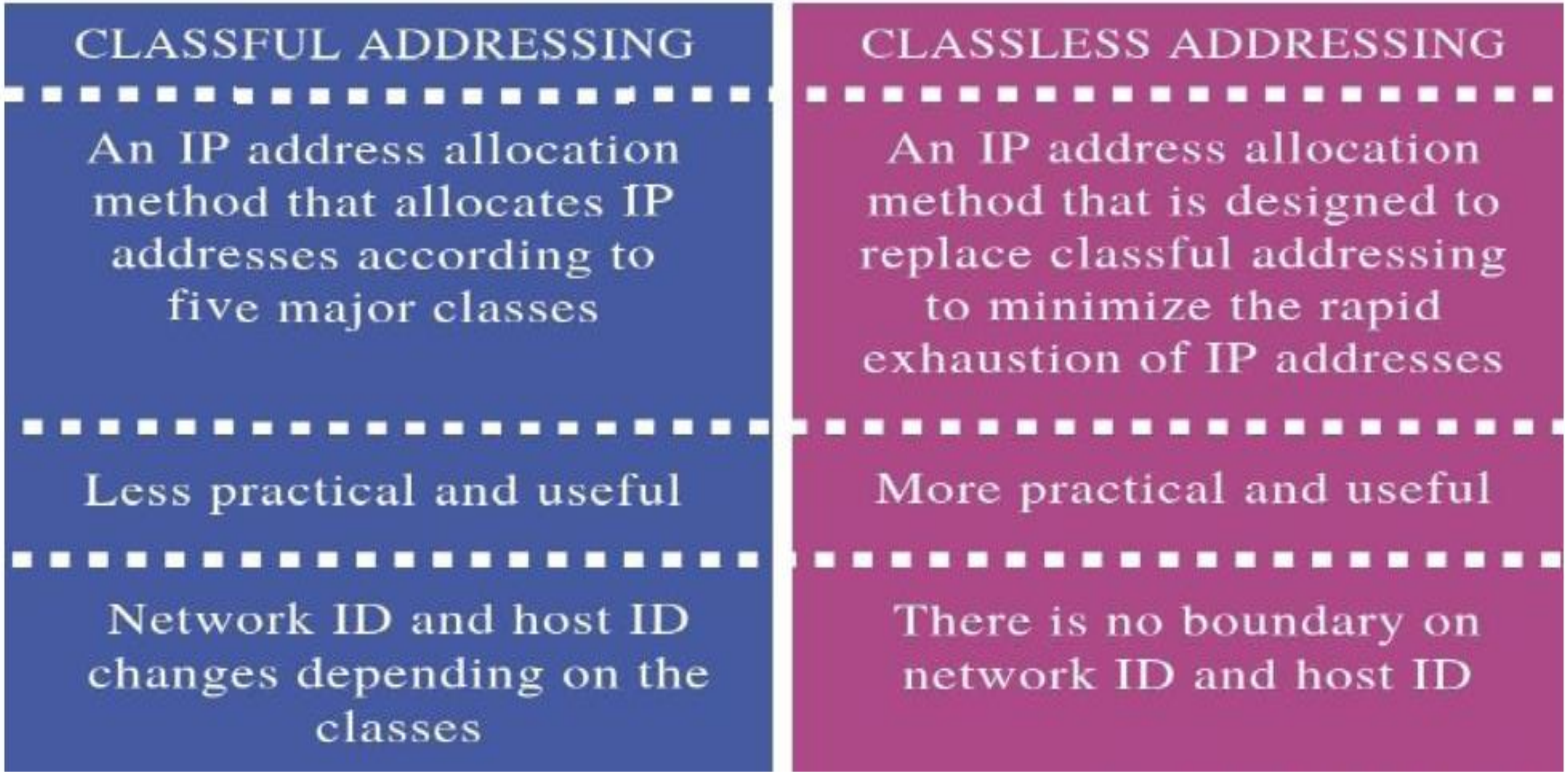
10.0.0.0-10.0.0.255(11111111.11111111.11111111.00000000)

Network mask equivalent to 255.255.255.0

The advantages of CIDR over the classful IP addressing are:

1. CIDR can be used to effectively manage the available IP address space.
2. CIDR can reduce the number of routing table entries

- **How it works**
- CIDR assigns a numerical prefix to each IP address. For example, a typical destination IP address using CIDR might be 177.67.5.44/13. The prefix 13 indicates that the first 13 bits of the IP address identify the network, while the remaining 32 - 13 = 19 bits identify the host. The prefix helps to identify the Internet destination gateway or group of gateways to which the packet will be forwarded. Prefixes vary in size, with longer prefixes indicating more specific destinations. Routers use the longest possible prefix in their routing tables when determining how to forward each packet. CIDR enables packets to be sent to groups of networks instead of to individual networks, which considerably simplifies the complex routing tables of the Internet's backbone routers.



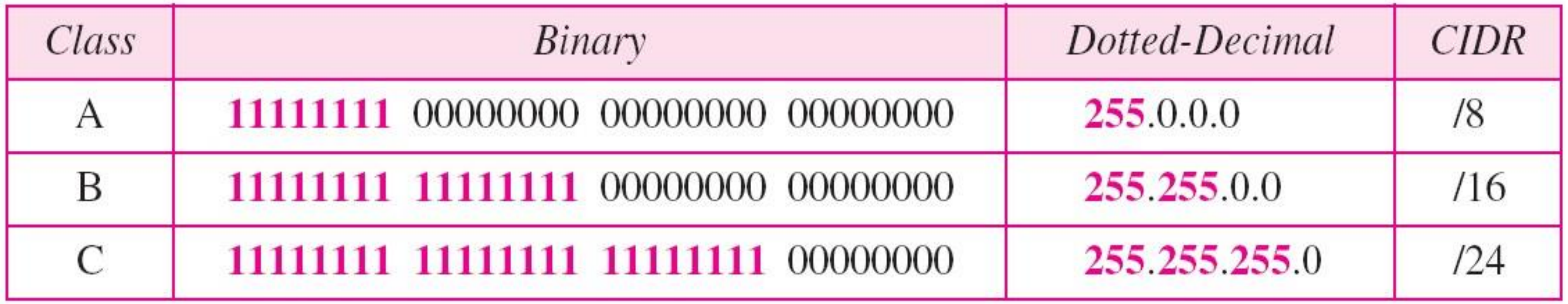
CLASSFUL ADDRESSING	CLASSLESS ADDRESSING
An IP address allocation method that allocates IP addresses according to five major classes	An IP address allocation method that is designed to replace classful addressing to minimize the rapid exhaustion of IP addresses
Less practical and useful	More practical and useful
Network ID and host ID changes depending on the classes	There is no boundary on network ID and host ID

Subnetting

- IP Subnetting is a process of dividing a large IP network in smaller IP networks.
- In Subnetting we create multiple small manageable networks from a single large IP network.
- Subnetting is the process of breaking down an IP network into smaller subnetworks called “subnets.” Each subnet is a non-physical description (or ID) for a physical sub-network .
- Subnetting is a process of segmentation of a network id into multiple broadcast domains.
- Subnetting originally referred to the subdivision of a class-based network into many subnetworks, but now it generally refers to the subdivision of a CIDR block in to smaller CIDR blocks.
- Subnetting allows single routing entries to refer either to the larger block or to its individual constituents.

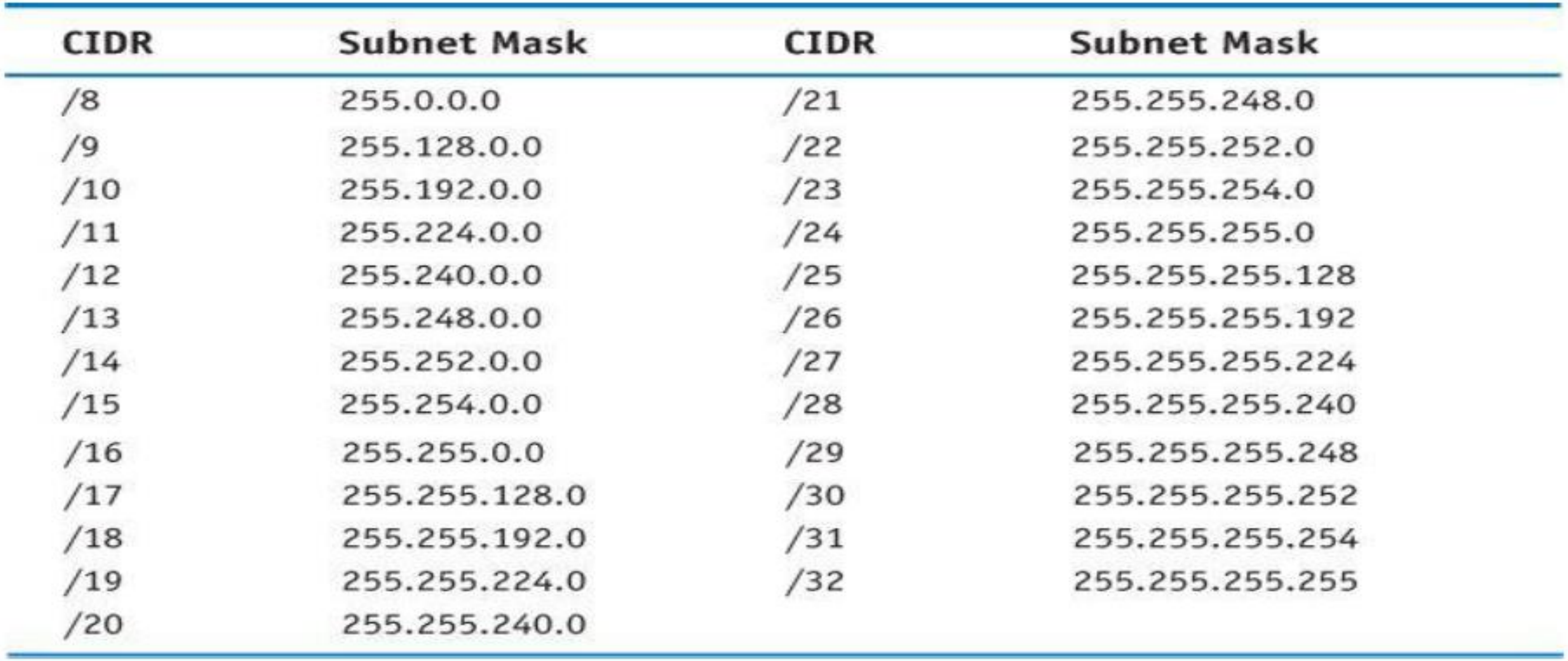
- **Advantages of Subnetting**
- Through subnetting, we can reduce network traffic and thereby improve network performance. we only allow traffic that should move to another network (subnet) to pass through the router and to the other subnet.
- Subnettiing can be used to restrict broadcast traffic on the network.
- Subnetting facilitates simplified management. we can delegate control of subnets to other administrators.
- Troubleshooting network issues is also simpler when dealing with subnets than it is in one large network.

- **Subnet Mask**
- An IP address has two components, the network address and the host address. A subnet mask separates the IP address into the network and host addresses.
- A Subnet mask is a 32-bit number that masks an IP address, and divides the IP address into network address and host address. Subnet Mask is made by setting network bits to all "1"s and setting host bits to all "0"s.



Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	**11111111** 00000000 00000000 00000000	**255**.0.0.0	/8
B	**11111111** **11111111** 00000000 00000000	**255**.**255**.0.0	/16
C	**11111111** **11111111** **11111111** 00000000	**255**.**255**.**255**.0	/24

CIDR and Subnet Mask



CIDR	Subnet Mask	CIDR	Subnet Mask
/8	255.0.0.0	/21	255.255.248.0
/9	255.128.0.0	/22	255.255.252.0
/10	255.192.0.0	/23	255.255.254.0
/11	255.224.0.0	/24	255.255.255.0
/12	255.240.0.0	/25	255.255.255.128
/13	255.248.0.0	/26	255.255.255.192
/14	255.252.0.0	/27	255.255.255.224
/15	255.254.0.0	/28	255.255.255.240
/16	255.255.0.0	/29	255.255.255.248
/17	255.255.128.0	/30	255.255.255.252
/18	255.255.192.0	/31	255.255.255.254
/19	255.255.224.0	/32	255.255.255.255
/20	255.255.240.0		

Class C Subnetting

Example: #1 Compute the network address of the IP 190.240.7.91

Here the given IP is of class B. Now here the subnet mask is not given, thus we use the default mask of class B as 255.255.0.0

To find out the network address, the router ANDs this mask with address to get 190.240.0.0

Example #2 Compute the network address (subnet address) of given IP address and Subnet mask

IP Address: 190.240.33.91

Subnet mask: 255.255.224.0/19

Solution: The Network address (or subnet address)is 190.240.32.0

Here we ANDed IP and Subnet mask

```
        190.240.33.91
 AND   255.255.224.0/19
        190.240.32.0
```

33=00100001

244=11100000 or /19 means 19 1s from the left.(2555.255.224.0/19=11111111.11111111.11100000.00000000)

ANDing (logical AND) we get the Network Address

Example #3Compute the network address (subnet address which is used by the router) of the following IP address and Subnet mask

IP Address: 130.50.15.6

Subnet mask: 255.255.252.0/22

Solution: The address that the router used is computed as in example#2 and the address is 130.50.12.0

Subnetting Steps

- Let's use the IP address 192.168.10.44 with subnet mask 255.255.255.248 (/29).

1.Total number of subnets: Using the subnet mask 255.255.255.248, number value 248 (11111000) indicates that 5 bits are used to identify the subnet. To find the total number of subnets available simply raise 2 to the power of 5 (2^5) and you will find that the result is 32 subnets.

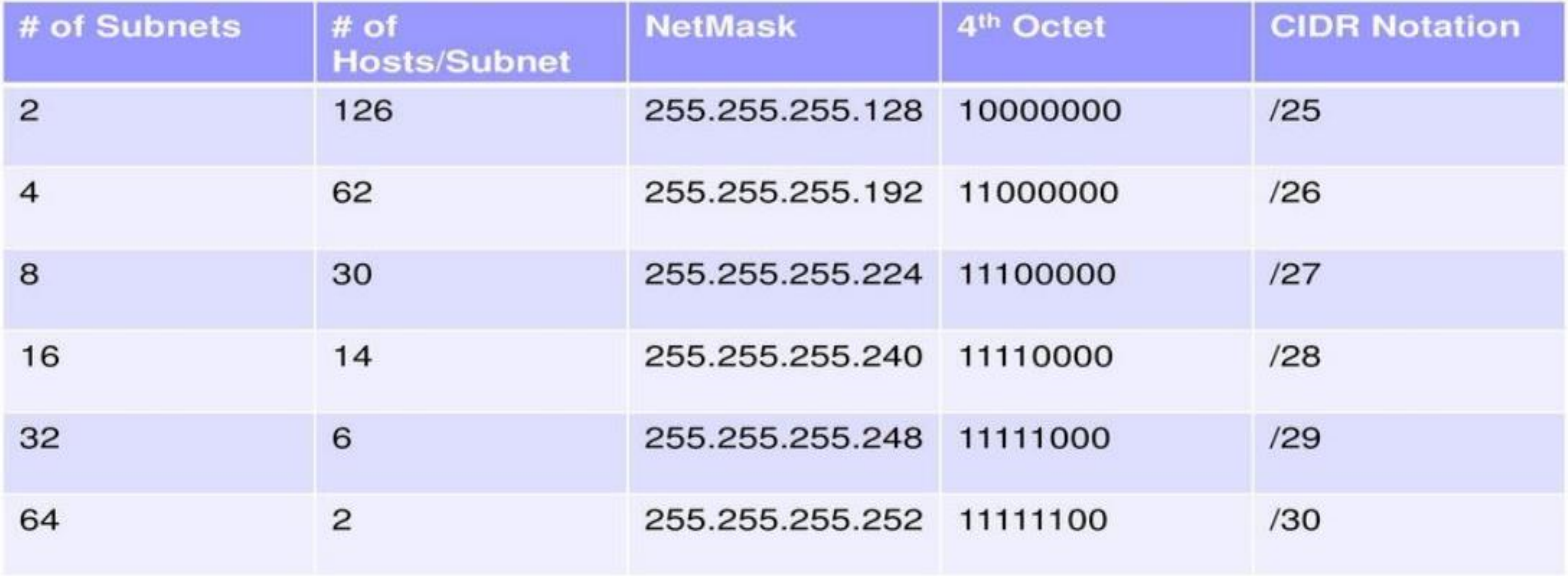
2.Hosts per subnet: 3 bits are left to identify the host therefore the total number of hosts per subnet is 2 to the power of 3 minus 2 (1 address for subnet address and another one for the broadcast address)(2^3-2) which equals to 6 hosts per subnet.

3. Subnets, hosts and broadcast addresses per subnet: To find the valid subnets for this specific subnet mask you have to subtract 248 from the value 256 (256-248=8) which is the first available subnet address.

Next subnet address is 8+8=16, next one is 16+8=24 and this goes on until we reach value 248. The following table provides all the calculated information.

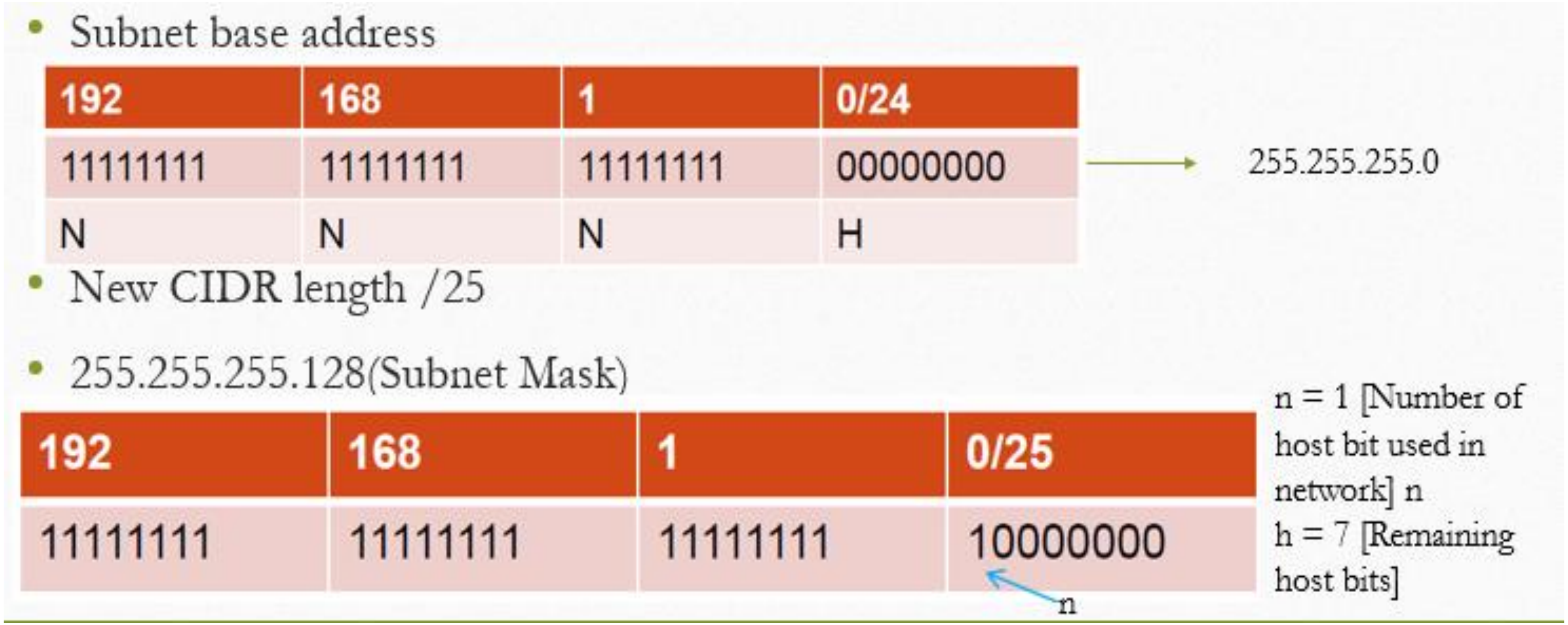
- **Formulas & Variables**
- h=number of host bits
- n=number of host bit use in network bits
- Number of new networks resulting from the subnetting: =2^n
- Number of hosts per new network: =2^h-2 where starting address used as network address and last one for broadcast address

Class C Subnetting



# of Subnets	# of Hosts/Subnet	NetMask	4th Octet	CIDR Notation
2	126	255.255.255.128	10000000	/25
4	62	255.255.255.192	11000000	/26
8	30	255.255.255.224	11100000	/27
16	14	255.255.255.240	11110000	/28
32	6	255.255.255.248	11111000	/29
64	2	255.255.255.252	11111100	/30

- Example



- Subnet base address

192	168	1	0/24
11111111	11111111	11111111	00000000
N	N	N	H

→ 255.255.255.0

- New CIDR length /25

- 255.255.255.128(Subnet Mask)

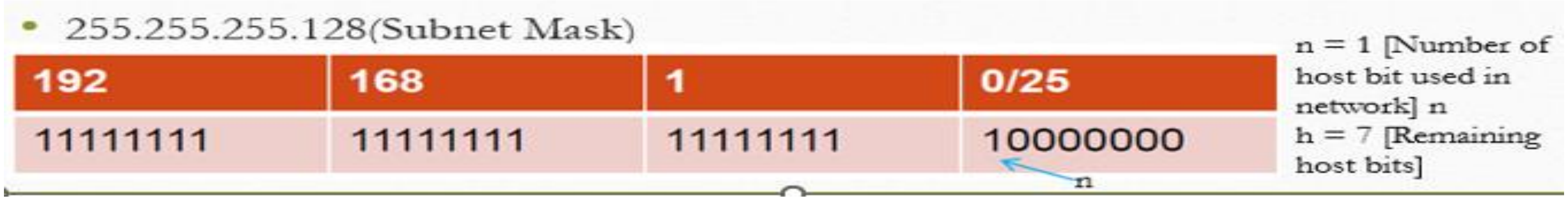
192	168	1	0/25
11111111	11111111	11111111	10000000

n

n = 1 [Number of host bit used in network] n
h = 7 [Remaining host bits]

Example 1

Suppose the IP address is 192.168.1.0/25 and subnet mask is 255.255.255.128



- 255.255.255.128(Subnet Mask)

192	168	1	0/25
11111111	11111111	11111111	10000000

n = 1 [Number of host bit used in network] n
h = 7 [Remaining host bits]

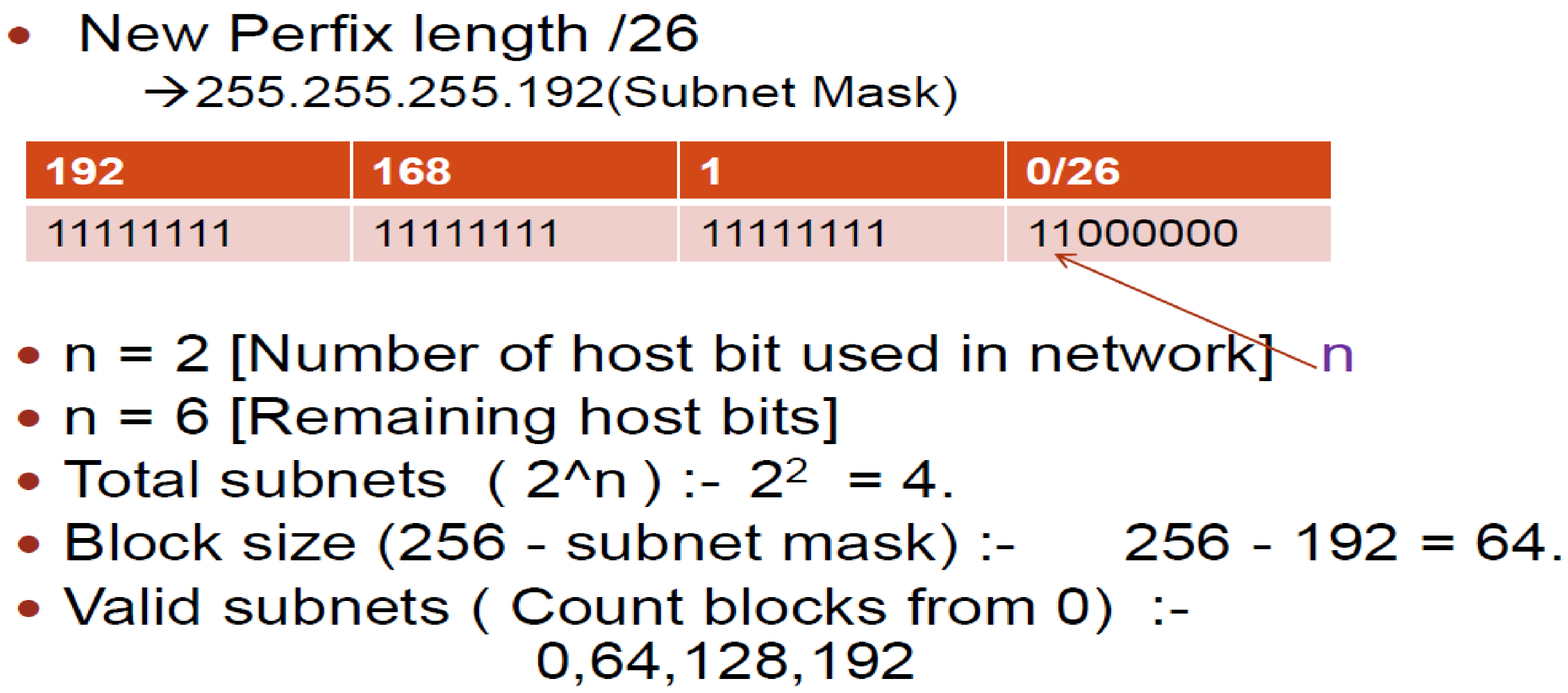
- Total subnets (2^n) :- 2^1 = 2.
- Block size (256 - subnet mask) :- 256 - 128 = 128
- Valid subnets 0,128
- Valid host per subnets (2^h-2)=2^7-2=126

Subnets	Subnet 1	Subnet 2
Network ID	192.168.1.0	192.168.1.128
First host	192.168.1.1	192.168.1.129
Last host	192.168.1.126	192.168.1.254
Broadcast ID	192.168.1.127	192.168.1.255

To find 1st address make host bit 0 and to find last address make host bit 1

Example

Suppose the IP address is 192.168.1.0/26 and subnet mask is 255.255.255.192



- New Perfix length /26

  →255.255.255.192(Subnet Mask)

192	168	1	0/26
11111111	11111111	11111111	11000000

- n = 2 [Number of host bit used in network] n
- n = 6 [Remaining host bits]
- Total subnets (2^n) :- 2^2 = 4.
- Block size (256 - subnet mask) :- 256 - 192 = 64.
- Valid subnets (Count blocks from 0) :-

  0,64,128,192

Make table as that in example 1 with same IP address

Subnets	Subnet 1	Subnet 2	Subnet 3	Subnet 4
Network ID	192.168.1.0	192.168.1.64	192.168.1.128	192.168.1.192
First host	192.168.1.1	192.168.1.65	192.168.1.129	192.168.1.193
Last host	192.168.1.62	192.168.1.126	192.168.1.190	192.168.1.254
Broadcast ID	192.168.1.63	192.168.1.127	192.168.1.191	192.168.1.255

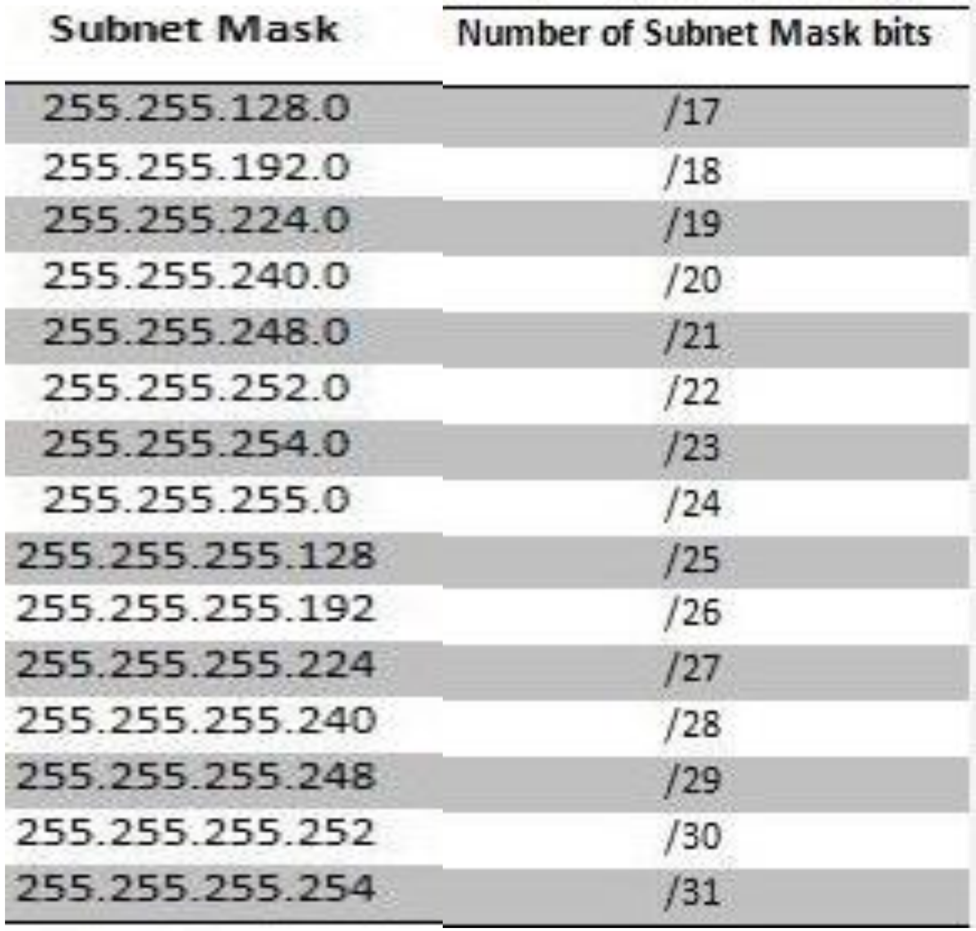
Questions

- Calculate the following

1. Total subnet
2. Block size
3. Valid subnet
4. Valid host per subnet
5. Find Network id, broadcast id, first host and last host of each subnet

Suppose the IP address is 192.168.1.0 and subnet mask is 255.255.255.224

Class B subnetting



Subnet Mask	Number of Subnet Mask bits
255.255.128.0	/17
255.255.192.0	/18
255.255.224.0	/19
255.255.240.0	/20
255.255.248.0	/21
255.255.252.0	/22
255.255.254.0	/23
255.255.255.0	/24
255.255.255.128	/25
255.255.255.192	/26
255.255.255.224	/27
255.255.255.240	/28
255.255.255.248	/29
255.255.255.252	/30
255.255.255.254	/31

- Subnet base address

172	16	0	0/16
11111111	11111111	00000000	00000000
N	N	H	H

255.255.0.0

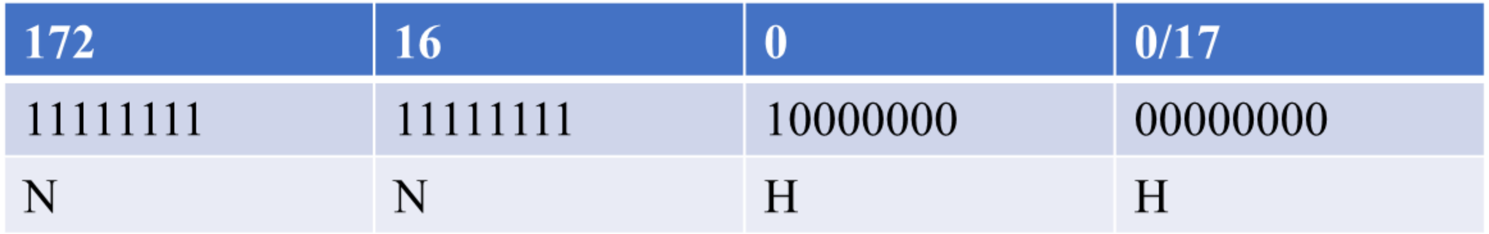
- New CIDR length /17
- 255.255.128.0(Subnet Mask)

172	16	0	0/17
11111111	11111111	10000000	00000000
N	N	H	H

n = 1 [Number of host bit used in network]
h = 15 [Remaining host bits]

Example 1

- Suppose the IP address is 172.16.0.0/17 and subnet mask is 255.255.128.0



172	16	0	0/17
11111111	11111111	10000000	00000000
N	N	H	H

n = 1 [Number of host bit used in network]
h = 15 [Remaining host bits]

- Total subnets (2^n): 2^1 = 2
- Block size (256-subnet mask) = 256-128 = 128
- Valid subnet 0,128
- Total host = 2^15 = 32768
- Valid host per subnets (2^h-2)=2^15-2 =32766

Subnets	Subnet 1	Subnet 2
Network ID	172.16.0.0	172.16.128.0
First host	172.16.0.1	172.16.128.1
Last host	172.16.127.254	172.16.255.254
Broadcast ID	172.16.127.255	172.16.255.255

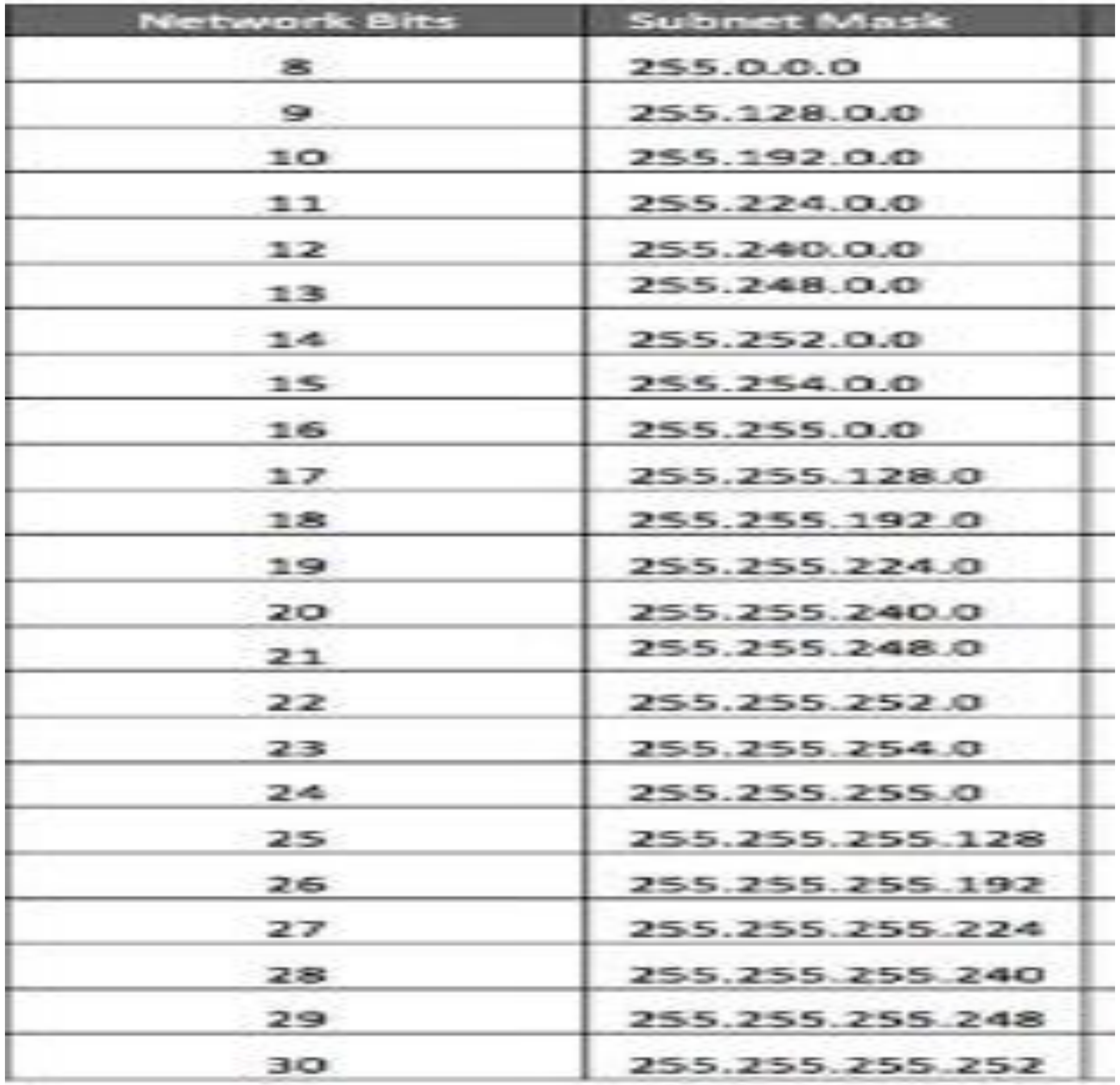
- Calculate the following

1. Total subnet
2. Block size
3. Valid subnet
4. Valid host per subnet
5. Find Network id, broadcast id, first host and last host of each subnet

Suppose the IP address is 172.16.0.0 and subnet mask is

- 255.255.192.0
- 255.255.224.0

Class A subnetting



Network Bits	Subnet Mask
8	255.0.0.0
9	255.128.0.0
10	255.192.0.0
11	255.224.0.0
12	255.240.0.0
13	255.248.0.0
14	255.252.0.0
15	255.254.0.0
16	255.255.0.0
17	255.255.128.0
18	255.255.192.0
19	255.255.224.0
20	255.255.240.0
21	255.255.248.0
22	255.255.252.0
23	255.255.254.0
24	255.255.255.0
25	255.255.255.128
26	255.255.255.192
27	255.255.255.224
28	255.255.255.240
29	255.255.255.248
30	255.255.255.252

- Subnet base address

10	0	0	0/8
11111111	00000000	00000000	00000000
N	H	H	H

255.0.0.0

- New CIDR length /9
- 255.128.0.0(Subnet Mask)

10	0	0	0/9
11111111	10000000	10000000	00000000
N	H	H	H

n = 1 [Number of host bit used in network]
h = 23 [Remaining host bits]

Example 1

- Suppose the IP address is 10.0.0.0/9 and subnet mask is 255.128.0.0

10	0	0	0/9
11111111	10000000	10000000	00000000
N	H	H	H

n = 1 [Number of host bit used in network]
h = 23 [Remaining host bits]

- Total subnets (2^n): 2^1 = 2
- Block size (256-subnet mask) = 256-128 = 128
- Valid subnet 0,128
- Total host = 2^23 = 8388608
- Valid host per subnets (2^h-2)=2^23-2 =8388606

Subnets	Subnet 1	Subnet 2
Network ID	10.0.0.0	10.128.0.0
First host	10.0.0.1	10.128.0.1
Last host	10.127.255.254	10.255.255.254
Broadcast ID	10.127.255.255	10.255.255.255

- Calculate the following

1. Total subnet
2. Block size
3. Valid subnet
4. Valid host per subnet
5. Find Network id, broadcast id, first host and last host of each subnet

Suppose the IP address is 10.0.0.0 and subnet mask is

- 255.192.0.0
- 255.224.0.0

EXAMPLE

- Suppose there are 4 Departments A(23 Hosts), B(16), C(28), D(13). Given a network 202.70.64.0/24, perform subneting in such way that IP wastage in each sub-network is minimum. Find Subnet mask, N/W ID, Broadcast ID and usable host range for each network.
    - Available Network is 202.70.64.0/24
    - Ie. Total range of available ip addresses :
        - 202.70.64.0 – 202.70.64.255
    - We proceed sub-netting with the department with highest no. of host ie, C and then A, B and D respectively.

- For Dept. C (Start with network with max^{m} hosts)
    - No. of hosts = 28
    - For No. of bits required for host(Suffix) part (H),
        - $2^{H} - 2 \geq 28$ => H = 5 (Select minimum value of H)
        - ie. Total no. of IP addresses this n/w can provide = $2^{5} = 32$
    - No. of bits for Network(Prefix) part = 32 – 5 = 27
    - No. of Subnets that can be created = 2^{27-24} = 8,
    - Let us Select Subnet for C as 202.70.64.0 / 27, then,
    - Subnet Mask = 255.255.255.[11100000] = 255.255.255.224
    - Network ID = 202.70.64.0 (The first ip address of network)
    - Broadcast ID = 202.70.64.31 (The last ip address of network)
    - Usable Host IP range = 202.70.64.1/27 – 202.70.64.30/27

- For Dept. A
    - No. of hosts = 23
    - For No. of bits required for host(Suffix) part (H),
        - 2^{H} -2 ≥ 23 => H = 5 (Select minimum value of H)
        - ie. Total no. of IP addresses this n/w can provide = 2^{5} = 32
    - No. of bits for Network(Prefix) part = 32 – 5 = 27
    - No. of Subnets that can be created = 2^{27-24} = 8
    - Let us Select Subnet for A as 202.70.64.32 / 27, then,
    - Subnet Mask = 255.255.255.[11100000] = 255.255.255.224
    - Network ID = 202.70.64.32 (The first ip address of network)
    - Broadcast ID = 202.70.64.63 (The last ip address of network)
    - Usable Host IP range = 202.70.64.33/27 – 202.70.64.62/27

- For Dept. B
  - No. of hosts = 16
  - For No. of bits required for host(Suffix) part (H),
    - 2^{H} -2 ≥ 16 => H = 5 (Select minimum value of H)
    - ie. Total no. of IP addresses this n/w can provide = 2^{5} = 32
  - No. of bits for Network(Prefix) part = 32 – 5 = 27
  - No. of Subnets that can be created = 2^{27-24} = 8,
  - Let us Select Subnet for B as 202.70.64.64 / 27, then,
  - Subnet Mask = 255.255.255.[11100000] = 255.255.255.224
  - Network ID = 202.70.64.64 (The first ip address of network)
  - Broadcast ID = 202.70.64.95 (The last ip address of network)
  - Usable Host IP range = 202.70.64.65/27 – 202.70.64.94/27

- For Dept. D
  - No. of hosts = 13
  - For No. of bits required for host(Suffix) part (H),
    - 2^{H} -2 ≥ 13 => H = 4 (Select minimum value of H)
    - ie. Total no. of IP addresses this n/w can provide = 2^{4} = 16
  - No. of bits for Network(Prefix) part = 32 – 4 = 28
  - No. of Subnets that can be created = 2^{28-24} = 16

    Let us Select Subnet for D as 202.70.64.96 / 28, then,
  - Subnet Mask = 255.255.255.[11110000] = 255.255.255.240
  - Network ID = 202.70.64.96 (The first ip address of network)
  - Broadcast ID = 202.70.64.111 (The last ip address of network)
  - Usable Host IP range = 202.70.64.97/28 – 202.70.64.110/28

EXAMPLE

- The network address given to ABC organization is 192.3.16.0 and it is divided among 8 different department. Find NID, NID and Usable Range in each department.
- Solution:

IP= 192.3.16.0

The above IP belongs to class C

We need to divide among 8 departments.

Default subnet mask of C: 255.255.255.0

Now,

Subnet = 8 =2^3

Subnet bit = 3

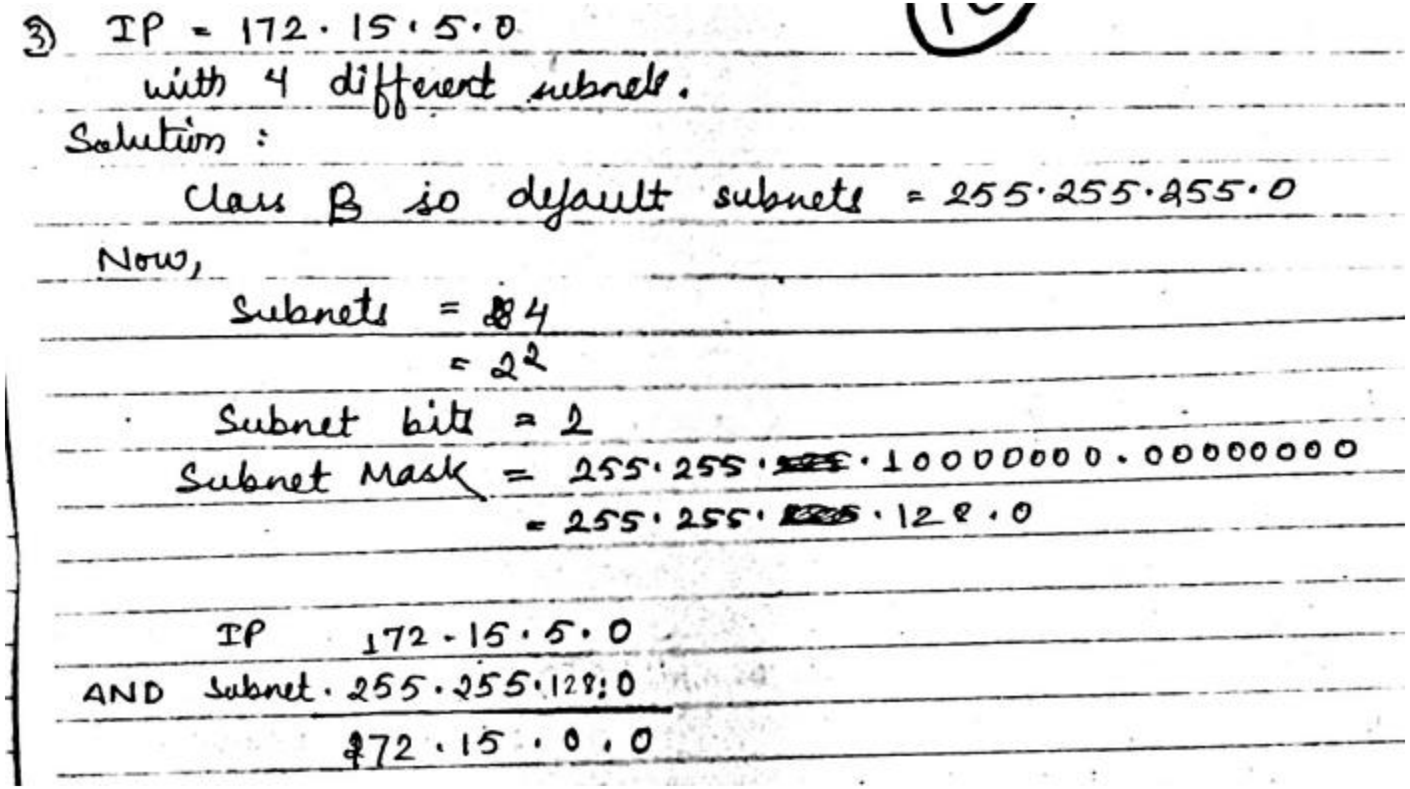
Subnet mask = 255.255.255.11100000= 255.255.255.224

- Block size (Network Jump)= 256-224 = 32
- Network bit (no. of 1’s in subnet mask)= 27
- Host bit (no. of 0s in subnet mask)=5

S.N.	NID	BID	USABLE RANGE
1	192.3.16.0	192.3.16.31	192.3.16.1-192.3.16.30
2	192.3.16.32	192.3.16.63	192.3.16.33-192.3.16.62
3	192.3.16.64	192.3.16.95	192.3.16.65-192.3.16.94
4	192.3.16.96	192.3.16.127	192.3.16.97-192.3.16.126
5	192.3.16.128	192.3.16.159	192.3.16.129-192.3.16.158
6	192.3.16.160	192.3.16.191	192.3.16.161-192.3.16.190
7	192.3.16.192	192.3.16.223	192.3.16.192-192.3.16.222
8	192.3.16.224	192.3.16.255	192.3.16.225-192.3.16.254

QUESTION

- The network address given to ABC organization is 131.215.0.0 and it is divided among 8 different department. Find NID, NID and Usable Range in each department.
- The network address given to ABC organization is 198.167.45.0 and it is divided among 6 different department. Find NID, NID and Usable Range in each department. {for 6 department we need 2^3 so subnet bit is 3}
- Suppose you are appointed as a N/w administrator at ABC organization where there are three departments A,B and C and these department requires 74, 56 and 85 computers respectively, if N/w ID is 221.22.222.0 then perform subnetting.



3) IP = 172 · 15 · 5 · 0

with 4 different subnets.

Solution :

Class B so default subnets = 255·255·255·0

Now,

Subnets = ~~8~~4

= 2^2

Subnet bits = 2

Subnet Mask = 255·255·~~255~~·10000000·00000000

= 255·255·~~255~~·128·0

IP 172·15·5·0

AND Subnet · 255·255·128:0

172·15·0·0

Make subnet table on your own

. If IP address 196 . 192 . 68 . 0 Subnet using ~~B~~ 32 host per subnet per subnet. Calculate N/W jump. Find the N/W ID, broadcast ID in all subnets. Also, calculate the usable IP per subnet.

Solution :

IP = 196 . 192 . 68 . 0

Subnet = 2^5

= 32

Subnet Mask = 255 . 255 . 255 . _

= 255 . 255 . 255 . _

= 255 . 255 . 255 . 192

Network Jump = 256 − ~~218~~ 192

= ~~8~~ 64

No of Host -

$2^H - 2 \geq 32$

$2^4 \geq 34$

$2^6 \geq$ $\boxed{H = 6}$

11000000

64 ≥

	N.ID	BID	UR
1)	196 . 192 . 68 . 0	196 . 192 . 68 . 63	196 . 192 . 68 . 1 − 196 . 192 . 68 .
2)	196 . 192 . 68 . 64	196 . 192 . 68 . 127	196 . 192 . 68 . 65 − 196 . 192 . 68 .
3)	196 . 192 . 68 . 128 .	196 . 192 . 68 . 191	196 . 192 . 68 . 129 − 196 . 192 . 68 . 1
4)	196 . 192 . 68 . 192	196 . 192 . 68 . 254	196 . 192 . 68 . 193 − 196 . 192 . 68 .

Waste IP = 64 − 34

= 30 .

Dynamic Addressing

- To this point, we have said that every computer knows its network layer address from a configuration file that is installed when the computer is first attached to the network. However, this leads to a major network management problem.
- Any time a computer is moved or its network is assigned a new address, the software on each individual computer must be updated which is very-time consuming.
- The easiest way around this is dynamic addressing. With this approach, a server is designated to supply a network layer address to a computer each time the computer connects to the network. This is commonly done for client computers but usually not for servers.

DHCP-DYNAMIC HOST CONFIGURATION PROTOCOL

- Application layer protocol used by hosts for obtaining network setup information.
- Allows a host to obtain an IP address dynamically without the network administrator having to set up an individual profile for each device.
- When using DHCP a defined range of IP addresses on a DHCP server is require.
- As hosts come online, they contact DHCP server and request the address.
- The DHCP server chooses an address and leases it to that host.
- With DHCP, the entire network configuration of a computer can be obtained in one message.
- A standard Internet protocol that enables the dynamic configuration of hosts on an Internet Protocol (IP) internetwork. Dynamic Host Configuration Protocol (DHCP) is an extension of the bootstrap protocol (BOOTP).

How it Works?

- DHCP is a client-server protocol that uses DHCP servers and DHCP clients. A DHCP server is a machine that runs a service that can lease out IP addresses and other TCP/IP information to any client that requests them. For example, on Linux System example Ubuntu you can install the DHCP Server service to perform this function. The DHCP server typically has a pool of IP addresses that it is allowed to distribute to clients, and these clients lease an IP address from the pool for a specific period of time, usually several days. Once the lease is ready to expire, the client contacts the server to arrange for renewal.
- DHCP clients are client machines that run special DHCP client software enabling them to communicate with DHCP servers. All versions of Linux and Windows include DHCP client software, which is installed when the TCP/IP protocol stack is installed on the machine.

DHCP clients obtain a DHCP lease for an IP address, a subnet mask, and various DHCP options from DHCP servers in a four-step process:

1. DHCPDISCOVER:

The client broadcasts a request for a DHCP server.

2. DHCPOFFER:

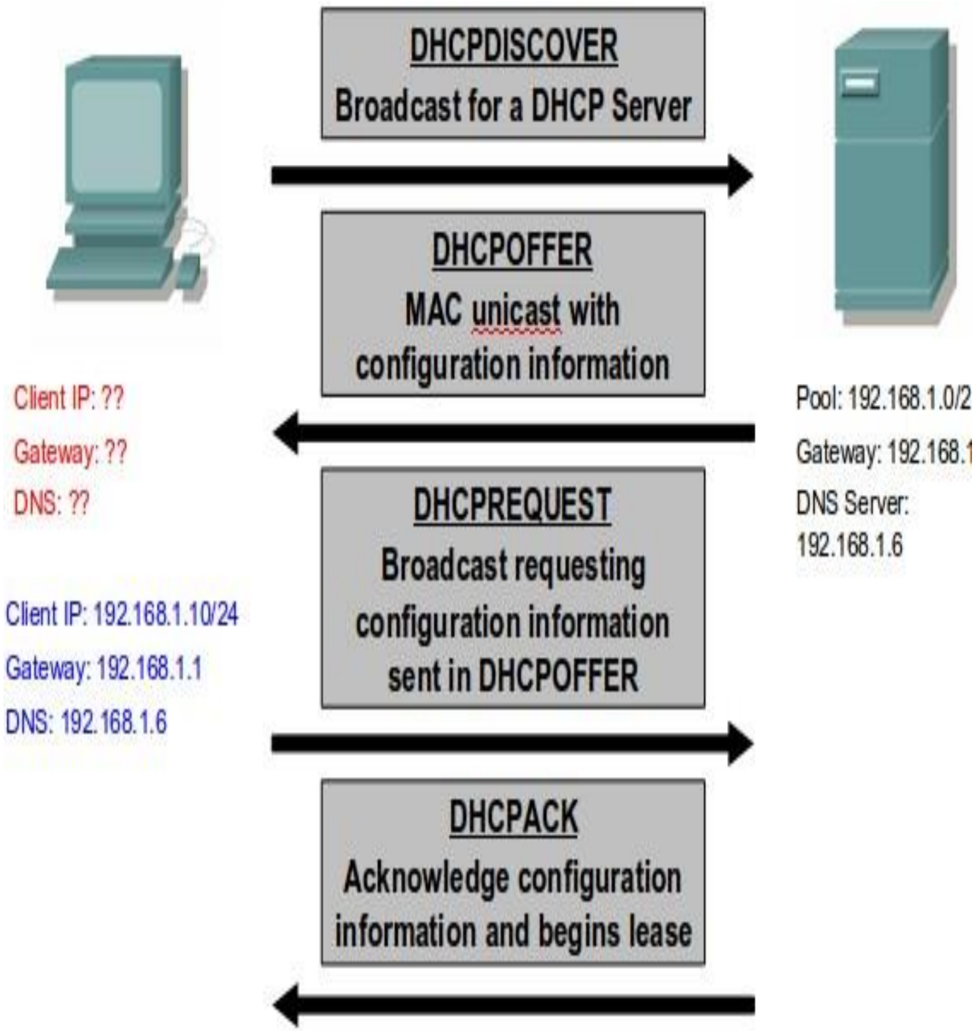
DHCP servers on the network offer an address to the client.

3. DHCPREQUEST:

The client broadcasts a request to lease an address from one of the offering DHCP servers.

4. DHCPACK:

The DHCP server that the client responds to acknowledges the client, assigns it any configured DHCP options, and updates its DHCP database. The client then initializes and binds its TCP/IP protocol stack and can begin network communication.


DHCPDISCOVER
Broadcast for a DHCP Server
DHCPOFFER
MAC unicast with configuration information
DHCPREQUEST
Broadcast requesting configuration information sent in DHCPOFFER
DHCPACK
Acknowledge configuration information and begins lease
Client IP: ??
Gateway: ??
DNS: ??
Client IP: 192.168.1.10/24
Gateway: 192.168.1.1
DNS: 192.168.1.6
Pool: 192.168.1.0/2
Gateway: 192.168.1
DNS Server: 192.168.1.6

Address Resolution

- To send a message, the sender must be able to translate the application layer address (or server name) of the destination into a network layer address and in turn translate that into a data link layer address. This process is called address resolution. TCP/IP uses two different approaches, one for resolving application layer addresses into IP addresses and a different one for resolving IP addresses into data link layer addresses.
- **Server Name Resolution**
- Server name resolution is the translation of application layer addresses into network layer addresses (e.g., translating an Internet address such as www.facebook.com into an IP address such as 157.240.15.35). This is done using the **Domain Name Service (DNS).** DNS is a global system for translating IP addresses to human-readable domain names. When a user tries to access a web address like "example.com", their web browser or application performs a DNS Query against a DNS server, supplying the hostname. The DNS server takes the hostname and resolves it into a numeric IP address, which then web browser can connect to.

When a user wants to use a file transfer client to access the file transfer server running on a remote host while the user is only aware of file transfer name. To establish the connection the TCP/IP suite must need IP address of the file transfer server. The given figure illustrates the working of the DNS step by step.

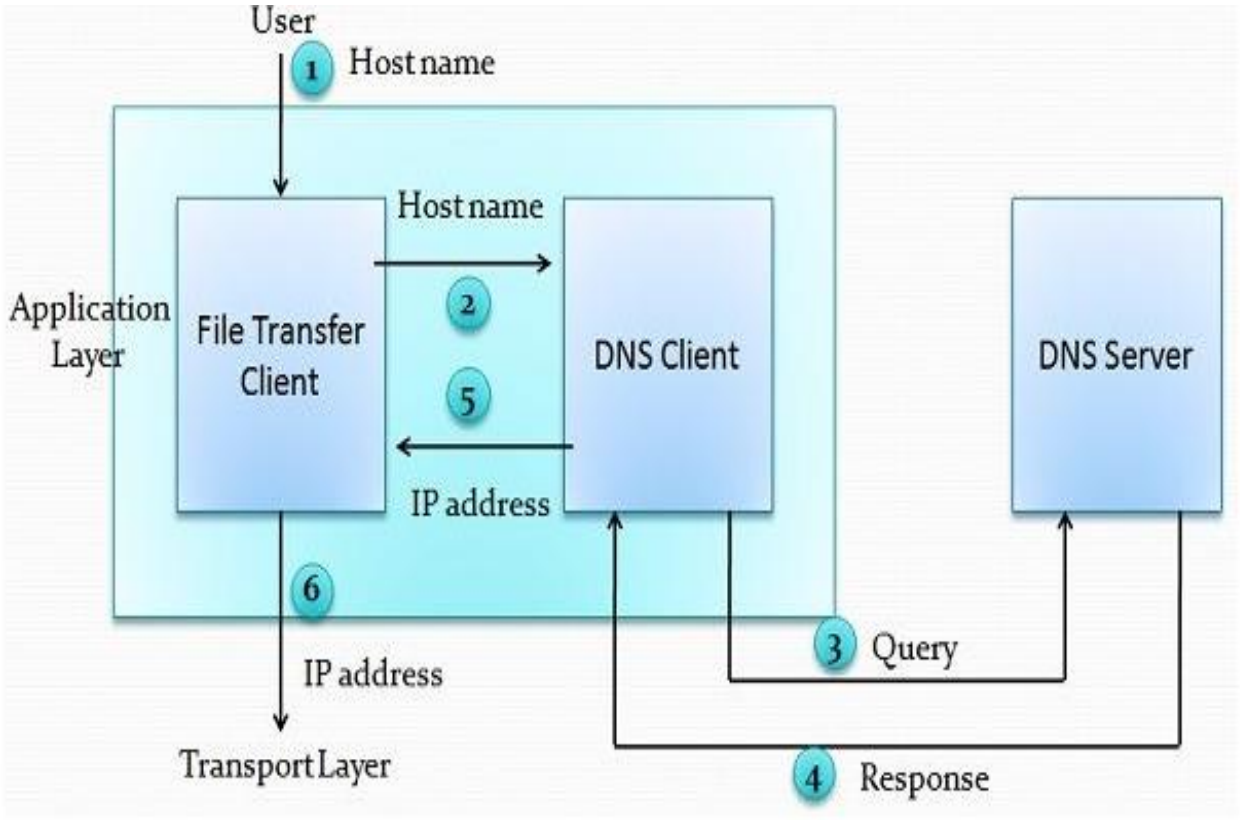
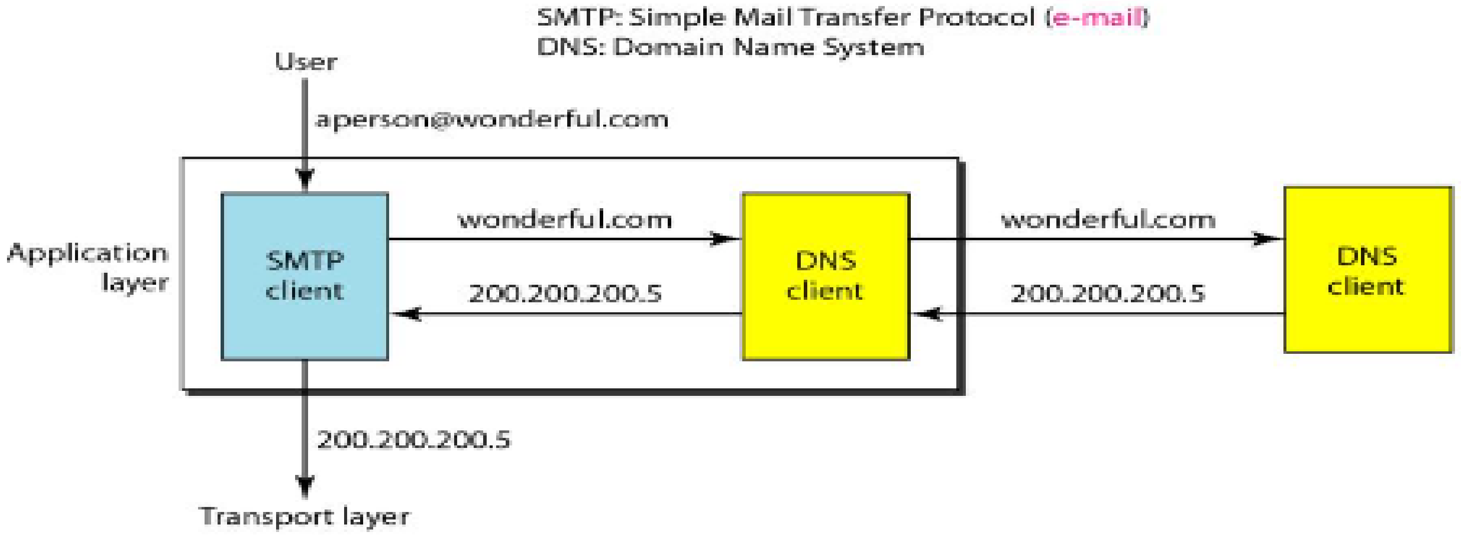


Figure: Purpose of DNS

1. The hostname is passed to the file transfer client by the user.
2. The file transfer client transits the hostname to the DNS client.
3. The DNS client sends the query to the DNS server which gives file transfer server name by utilizing known IP address of the DNS server.
4. DNS server sends the response with the IP address of the required file transfer server.
5. The DNS client passes the IP address to the file transfer server.
6. The received IP address is used by file transfer client to access the file transfer server.

- Note that the purpose of accessing the Internet is to make a connection between the file transfer client and server, but before this can happen, another connection needs to be made between the DNS client and DNS server. In other words, we need at least two connections in this case. The first is for mapping the name to an IP address; the second is for transferring files.
- Figure shows an example of how DNS client/server program.



- There are three types of queries in the DNS system:
- **Recursive Query:** In a recursive query, a DNS client provides a hostname, and the DNS Resolver must provide an answer-it responds with either a relevant resource record, or an error message if it can't be found. The resolver starts a recursive query process, starting from the DNS Root Server, until it finds the Authoritative Name Server (for more on Authoritative Name Servers see DNS Server Types below) that holds the IP address and other information for the requested hostname.
- **Iterative Query:** In an iterative query, a DNS client provides a hostname, and the DNS Resolver returns the best answer it can. If the DNS resolver has the relevant DNS records in its cache, it returns them. If not, it refers the DNS client to the Root Server, or another Authoritative Name Server which is nearest to the required DNS zone. The DNS client must then repeat the query directly against the DNS server it was referred to.

- **Non-Recursive Query:** A non-recursive query is a query in which the DNS Resolver already knows the answer. It either immediately returns a DNS record because it already stores it in local cache, or queries a DNS Name Server which is authoritative for the record, meaning it definitely holds the correct IP for that hostname. In both cases, there is no need for additional rounds of queries(like in recursive or iterative queries). Rather, a response is immediately returned to the client.

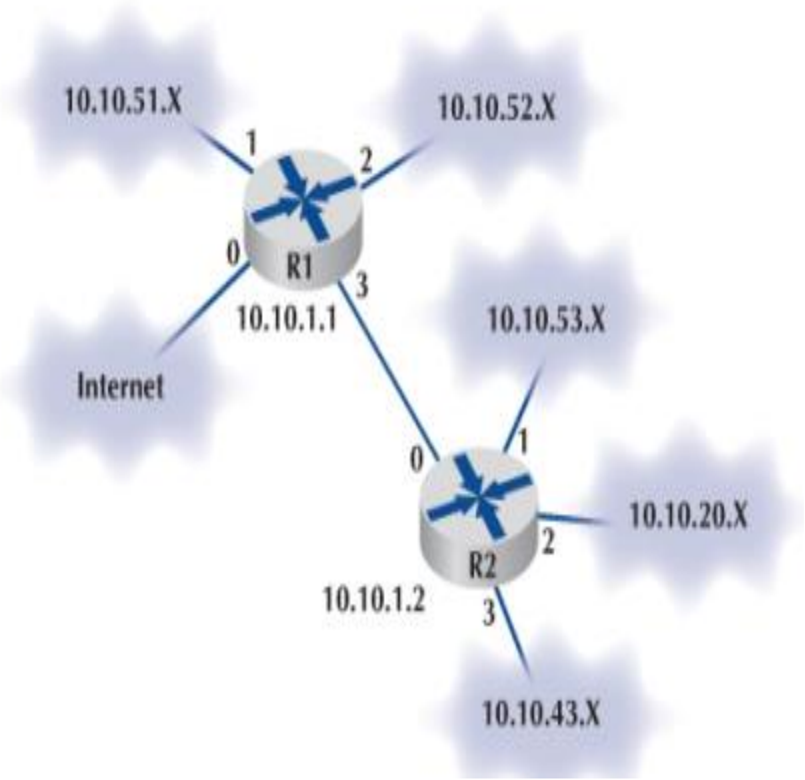
- **Data Link Layer Address Resolution**
- The Address Resolution Protocol (ARP) is a protocol that connects an ever-changing Internet Protocol (IP) address to a fixed physical machine address, also known as a media access control (MAC) address, in a LAN. This mapping procedure is important because the lengths of the IP and MAC addresses differ, and a translation is needed so that the systems can recognize one another. The MAC address is also known as the data link layer, which establishes and terminates a connection between two physically connected devices so that data transfer can take place. The IP address is also referred to as the network layer or the layer responsible for forwarding packets of data through different routers. ARP works between these layers.
- When a new computer joins a LAN, it will receive a unique IP address to use for identification and communication. Packets of data arrive at a gateway, destined for a particular host machine. The gateway asks the ARP program to find a MAC address that matches the IP address. There are three basic ARP terms:

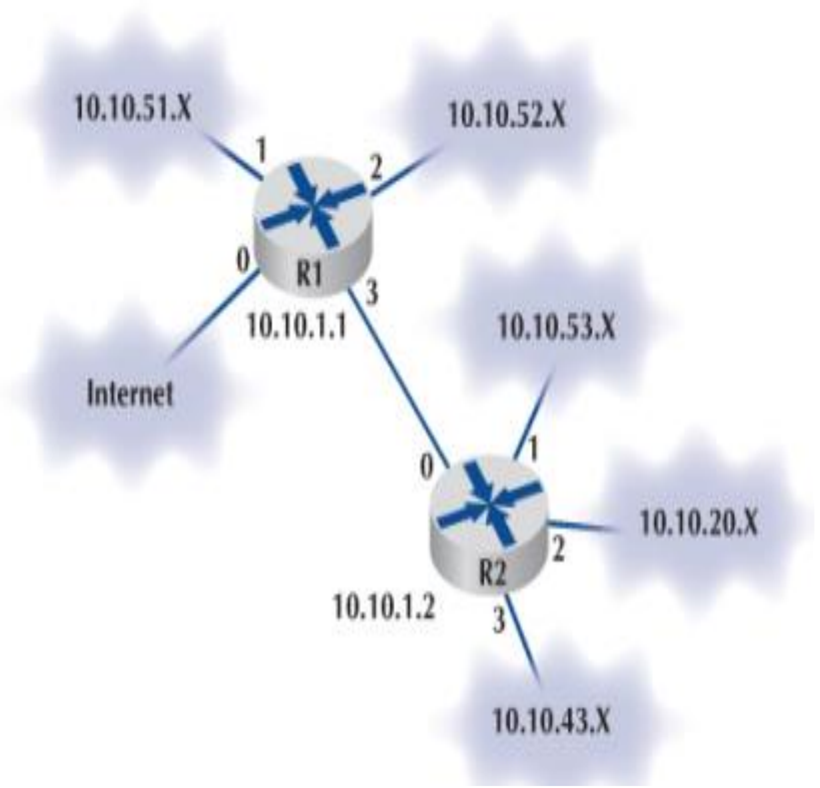
- **Reverse ARP:** Reverse Address Resolution Protocol is used in local area networks LAN) by client machines for requesting IP Address from Router's ARP Table.
- **Proxy ARP:** Proxy Address Resolution Protocol work to enable devices that are separated into network segments connected through the router in the same IP to resolve IP Address to MAC Address.
- **Inverse ARP**: Inverse Address Resolution Protocol uses MAC Address to find the IP Address.

Routing

- Routing is the process of determining the route or path through the network that a message will travel from the sending computer to the receiving computer. In some networks (e.g., the Internet), there are many possible routes from one computer to another. In other networks (e.g., internal company networks), there may only be one logical route from one computer to another. In either case, some device has to route messages through the network.
- Routing is done by special devices called routers. Routers are usually found at the edge of subnets because they are the devices that connect subnets together and enable messages to flow from one subnet to another as the messages move through the network from sender to receiver.

- Figure shows a small network with two routers, R1 and R2. This network has five subnets, plus a connection to the Internet. Each subnet has its own range of addresses (e.g., 10.10.51.x), and each router has its IP address (e.g., 10.10.1.1). The first router (R1) has four connections, one to the Internet, one to router R2, and one to each of two subnets. Each connection, called an interface, is numbered from 0 to 3. The second router (R2) also has four interfaces, one that connects to R1 and three that connect to other subnets.
- Every router has a routing table that specifies how messages will travel through the network. In its simplest form, the routing table is a two-column table. The first column lists every network or computer that the router knows about, and the second column lists the interface that connects to it





Router R1's Routing Table

Network Address	Interface
10.10.51.0–10.10.51.255	1
10.10.52.0–10.10.52.255	2
10.10.53.0–10.10.53.255	3
10.10.20.0–10.10.20.255	3
10.10.43.0–10.10.43.255	3
10.10.1.2	3
All other addresses	0

Router R2's Routing Table

Network Address	Interface
10.10.1.1	0
10.10.53.0–10.10.53.255	1
10.10.20.0–10.10.20.255	2
10.10.43.0–10.10.43.255	3
All other addresses	0

- **Types of Routing**
- There are three fundamental approaches to routing: centralized routing, static routing, and dynamic routing.
- **Centralized Routing**
- With centralized routing, all routing decisions are made by one central computer or router. Centralized routing is commonly used in host-based networks, and in this case, routing decisions are rather simple. All computers are connected to the central computer, so any message that needs to be routed is simply sent to the central computer, which in turn retransmits the message on the appropriate circuit to the destination.
- **Static Routing**
- Static routing is a process in which we have to manually add routes in routing table. Static routing does not involve any change in routing table unless the network administrator changes or modify them manually. Static routing algorithms function well where the network traffic is predictable. This is simple to design and easy to implement. There is no requirement of complex routing protocols.

Advantages of Static Routing

- No routing overhead for router CPU which means a cheaper router can be used to do routing.
- It adds security because only administrator can allow routing to particular networks only.
- No bandwidth usage between routers.

Disadvantage of Static Routing

- For a large network, it is a hectic task for administrator to manually add each route for the network in the routing table on each router.
- The administrator should have good knowledge of the topology. If a new administrator comes, then he has to manually add each route so he should have very good knowledge of the routes of the topology.

- **Dynamic Routing**
- Dynamic routing makes automatic adjustment of the routes according to the current state of the route in the routing table. Dynamic routing uses protocols to discover network destinations and the routes to reach it. RIP and OSPF are the best examples of dynamic routing protocol. Automatic adjustment will be made to reach the network destination if one route goes down.
- A dynamic protocol have following features:
- The routers should have the same dynamic protocol running in order to exchange routes.
- When a router finds a change in the topology then router advertises it to all other routers.

Advantages of Dynamic Routing

- Easy to configure.
- More effective at selecting the best route to a destination remote network and also for discovering remote network.

Disadvantage of Dynamic Routing

- Consumes more bandwidth for communicating with other neighbors.
- Less secure than static routing

- **Distance Vector Routing**
- It is a dynamic routing algorithm in which each router computes distance between itself and each possible destination i.e. its immediate neighbors.
- The router share its knowledge about the whole network to its neighbors and accordingly updates table based on its neighbors.
- The sharing of information with the neighbors takes place at regular intervals.
- It makes use of **Bellman Ford Algorithm** for making routing tables.
- RIP and IGRP is a commonly used distance vector protocol that uses hop counts or its routing metrics.
- Problems – Count to infinity problem which can be solved by splitting horizon. – Persistent looping problem i.e. loop will be there forever

- The Bellman Ford algorithm operates by having each router maintain a table (vector). The routing table contains two parts. They are:

a) The preferred outgoing line to use for that destination.

b) Estimate of the time or distance to that destination.

Procedure of Distance Vector Routing:

1. Initially, the router makes a list of which networks it can reach, and how many hops it will cost. In the outset this will be the two or more networks to which this router is connected. The number of hops for these networks will be 1. This table is called a routing table.

2. Periodically (typically every 30th second) the routing table is shared with other routers on each of the connected networks via some specified inter-router protocol. These routers will add 1 to every hop-count in the table, as it associates a hop cost of 1 for reaching the router that sent the table. This information is just shared in between physically connected routers ("neighbors"), so routers on other networks are not reached by the new routing tables yet.

3. A new routing table is constructed based on the directly configured network interfaces, as before, with the addition of the new information received from other routers. The hop-count is used as a cost measure for each path. The table also contains a column stating which router offered this hop count. so that the router knows who is next in line for reaching a certain network.

4. Bad routing paths are then removed from the new routing table. If two identical paths to the same network exists, only the one with the smallest hop-count is kept. When the new table has been cleaned up, it may be used to replace the existing routing table used for packet forwarding.

5. The new routing table is then communicated to all neighbors of this router. This way the routing information will spread and eventually all routers know the routing path to each network which router it shall use to reach this network, and to which router it shall route next.

Example

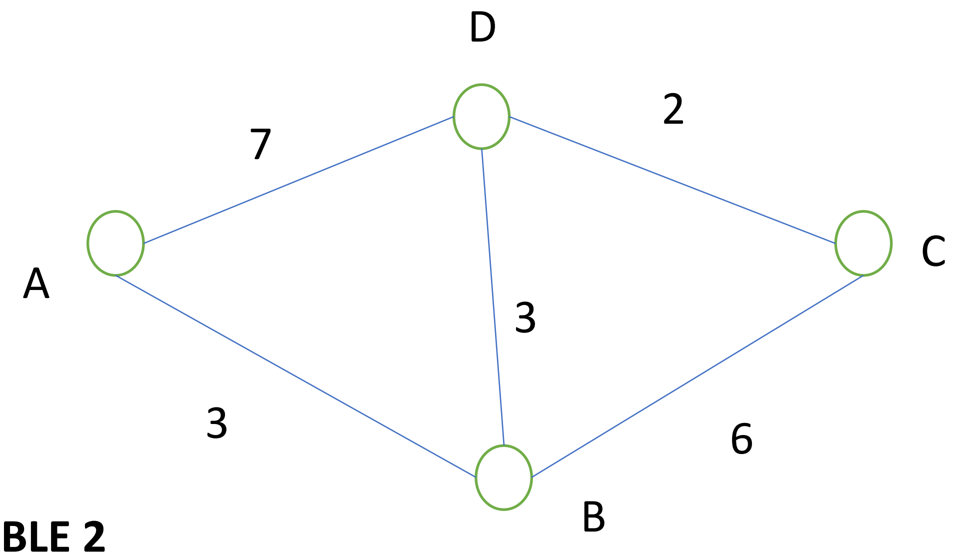


TABLE 1

	A	**B**	**C**	**D**
A	0	∞	∞	∞
B	∞	0	∞	∞
C	∞	∞	0	∞
D	∞	∞	∞	0

TABLE 2

	A	**B**	**C**	**D**
A	0	3A	∞	7A
B	3B	0	6B	3B
C	∞	6C	0	2C
D	7D	3D	2D	0

Example

TABLE 3

	A	**B**	**C**	**D**
A	0	3A	8B	6B
B	3B	0	6B	3B
C	9B	5D	0	2C
D	7D	3D	2D	0

TABLE 4

	A	**B**	**C**	**D**
A	0	3A	8B	6B
B	3B	0	6B	3B
C	8D	5D	0	2C
D	6B	3D	2D	0

- **Link State Routing**
- It is a dynamic routing algorithm in which each router shares knowledge of its neighbors with every other router in the network.
- A router sends its information about its neighbours only to all the routers through flooding.
- Information sharing takes place only whenever there is a change.
- It makes use of Dijkastra’s Algorithm for making routing tables.
- Problems – Heavy traffic due to flooding of packets. – Flooding can result in infinite looping which can be solved by using Time to leave (TTL) field.

- **The Link State Routing Algorithm:**

Step 1: Discover its neighbors and learn their network addresses:

As soon as router is booted it sends a hello packet on each point to point line and the router at the other end sends back a reply telling who they are.

Step 2: Measure the delay or cost to each of its neighbors:

A special Echo packet is send by the source and the destination has to send back immediately. So. the delay is calculated by measuring the round trip time and dividing it by 2. This can be carried out multiple times for better results

Step 3: Build link state packets:

Link state packets are build which contains identity of the sender, sequence number., age and a list of neighbor with cost.

Step 4: Distribute this packet to all routers:

- It uses 32 bit sequence number to avoid confusion. It uses the concept of flooding to distribute the link state packets. Each packet is assigned a sequence number so that duplicates can be discarded. If router crashes or if sequence number is corrupted, then all the next packets will be rejected. So, age filed is attached to a packet and is decremented once per second. When age hits zero , the packet is discarded. Age field is to make sure no packet can get lost and live for indefinite period of time.

Step 5: Compute the shortest path to every other node:

- Dijkstra's algorithm can be run to construct the shortest path to all the possible destinations. Once all the routers has received all the LSP’s, the routers then construct a topological map of the network which is used to determine the best route to destination. In this algorithm, routers can independently determine the shortest path to every network.

Dijkstra’s Algorithm:

- finds shortest paths from given source node s to all other nodes
- **Dijkstra's Algorithm** has three steps;

➢ Step 1 [Initialization]

□ N = set of nodes in the network

□ s = source node

□ T = {s} Set of nodes so far incorporated

□ $w(i, j)$ = link cost from node i to node j;

□ $w(i, i) = 0$;

□ $w(i, j) = \infty$ if two nodes not directly connected;

□ $w(i, j) \geq 0$ if two nodes are directly connected

□ $L(n)$ = cost of the least-cost path from node s to node n that is currently known to the algorithm; at termination, this is the cost of the least-cost path from s to n.

□ $L(n) = w(s, n)$ for $n \neq s$

□ initial path costs to neighboring nodes are simply link costs

- Step 2 [Get Next Node]

  - find neighboring node not in T with least-cost path from s

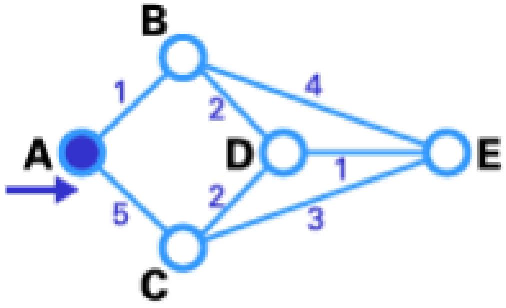
  - incorporate node into T

  - also incorporate the edge that is incident on that node and a node in T that contributes to the path

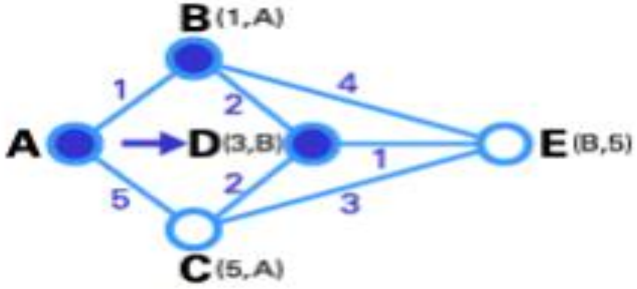
- Step 3 [Update Least-Cost Paths]

- Steps 2 and 3 are repeated until T = N. That is, steps 2 and 3 are repeated until final paths have been assigned to all nodes in the network. The algorithm terminates when all nodes have been added to T.
- When the shortest path from source to another node is discovered, it is made permanent and never changed thereafter

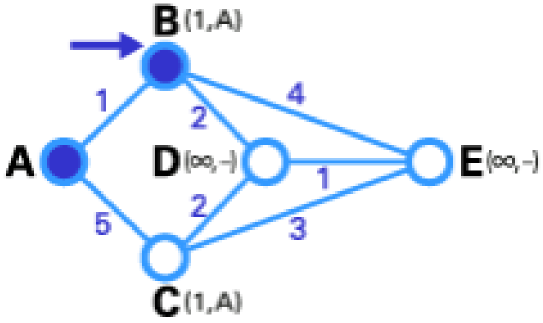
EXAMPLE



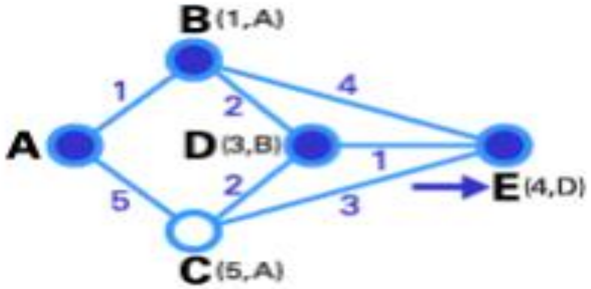
Step 1



Step 3



Step 2



Step 4

EXAMPLE

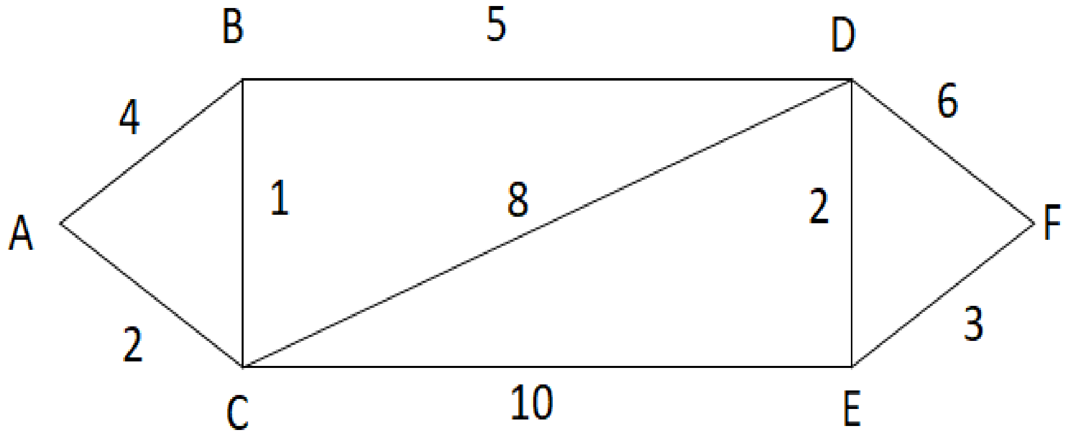
A-C = 2

A-C-B = 2+1 = 3

A-C-B-D = 2+1+5= 8

A-C-B-D-E = 2+1+5+2 = 10

A-C-B-D-E-F = 2+1+5+2+3 = 13



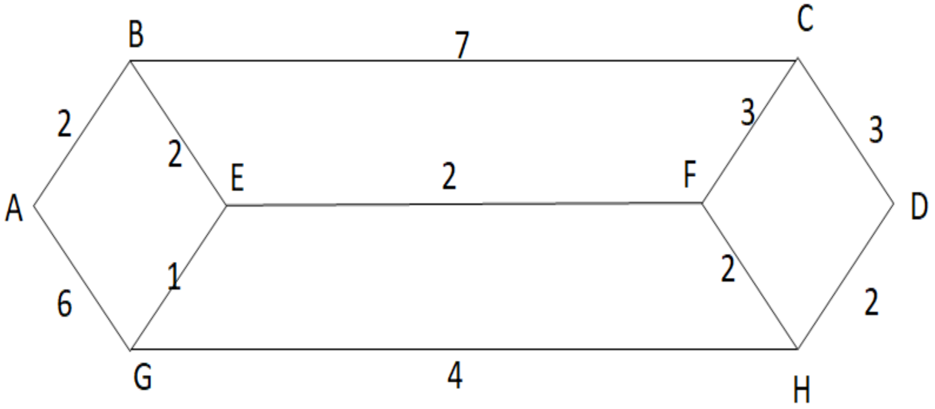
A- B = 2

A-B-E =2+2 =4

A-B-E-F = 2+2+2 =6

A-B-E-F-H = 2+2+2+2 =8

A-B-E-F-H-D = 2+2+2+2+2 = 10



LS vs DV algorithm

Distance vector algorithms **(also known as Bellman-Ford algorithms):** Key features of the distance vector routing are as follows:

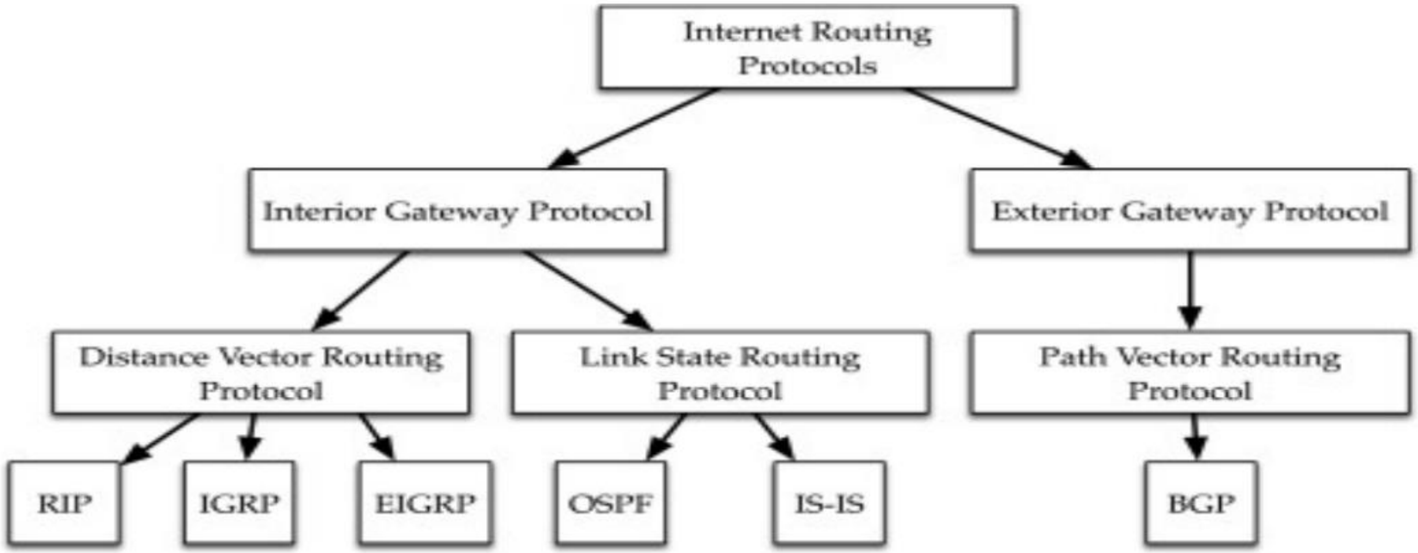
- The routers keep information such as its distance vector, cost to its neighbor and its neighbor distance
- Sharing of information takes place only with the neighbors
- Sharing of information takes place at fixed regular intervals, say every 30

Link-state algorithms **(also known as shortest path first algorithms)** have the following key feature

- The routers share keeps information about entire topology and the link cost between the routers.
- Sharing of information takes place with all the routers in the internet, by sending small updates using flooding compared to sending larger updates to their neighbors
- Sharing of information takes place only when there is a change, which leads to lesser internet traffic compared to distance vector routing

Distance Vector	Link State
• Router sends (destination, distance) information from routing table to all neighbors (all entries or changed entries only) • each router simply inform its neighbors of its routing table • Simple to implement • Simple to configure • Bad convergence (count-to infinity)	• Router sends list of direct neighbors to direct neighbors • When a network link changes state, *link state advertisement* (LSA) is *flooded,* All the routers note the change, and re-compute their routes accordingly • more reliable, easier to debug and less bandwidth-intensive • more complex and more compute and memory-intensive • Faster convergence • Generates less traffic (may be better controlled) • Fast reaction to topology changes • Fast reaction to communication breaks •

Different types of routing Protocols


Internet Routing Protocols
Interior Gateway Protocol
Exterior Gateway Protocol
Distance Vector Routing Protocol
Link State Routing Protocol
Path Vector Routing Protocol
RIP
IGRP
EIGRP
OSPF
IS-IS
BGP

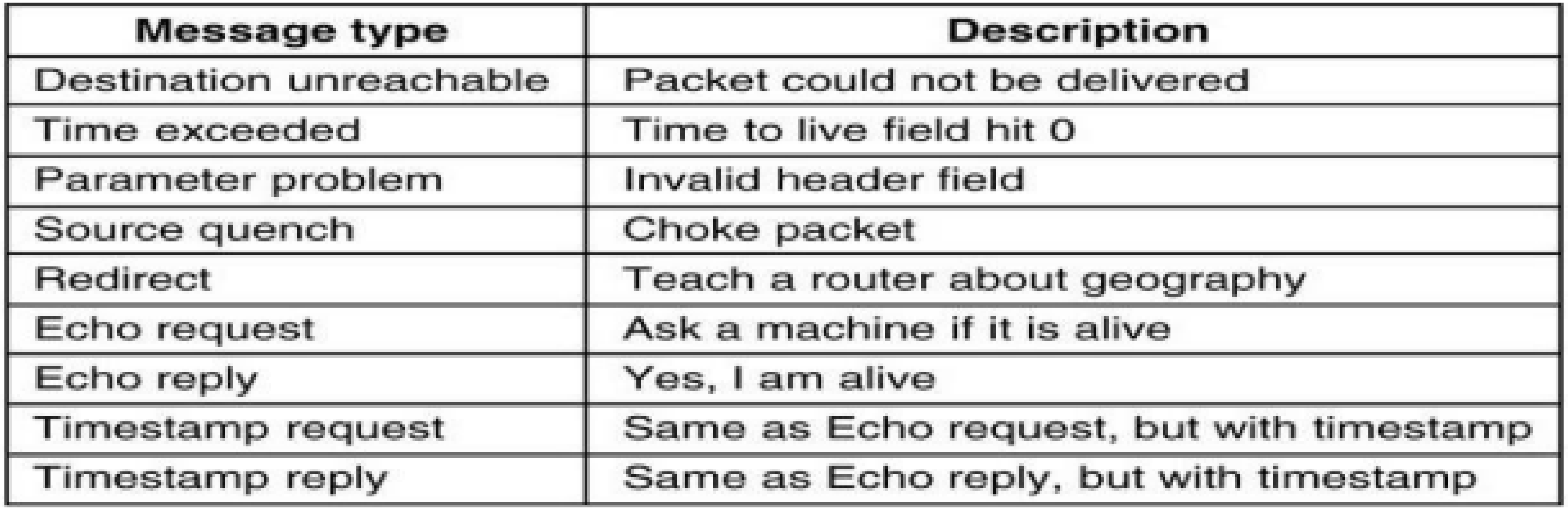
Five are commonly used on the Internet: Border Gateway Protocol (BGP), Internet Control Message Protocol (ICMP), Routing Information Protocol (RIP), Intermediate System to Intermediate System (IS-IS) Open Shortest Path First (OSPF), and Enhanced Interior Gateway Routing Protocol (EIGRP).

- **Interior gateway protocols:**
- as the Internet community calls them, are typically used in small, cooperative set of networks such as might be found on a university campus.
- One of the oldest interior protocols is Routing Information Protocol, or RIP.
- Newer interior protocols include Interior Gateway Routing Protocol, or IGRP, and Open Shortest Path First, or OSPF.
- Cisco network devices can also use Cisco’s proprietary Enhanced Interior Gateway Routing Protocol, or EIGRP.
- Interior protocols are fairly easy to set up, but do not scale well to large networks.
- **Exterior Gateway Protocol (EGP):**
- is a protocol or exchanging routing information between two neighbor gateway hosts (each with its own router) in a network of autonomous systems.
- EGP is commonly used between hosts on the Internet to exchange routing table information.
- The routing table contains a list of known routers, the addresses they can reach, and a cost metric associated with the path to each router so that the best available route is chosen.

- **BGP (Border Gateway Protocol)**
- Border Gateway Protocol (BGP) is an Internet Engineering Task Force (IETF) standard, and the most scalable of all routing protocols.
- BGP (Border Gateway Protocol) is protocol that manages how packets are routed across the internet through the exchange of routing and reachability information between edge routers.
- BGP directs packets between autonomous systems (AS) -- networks managed by a single enterprise or service provider.
- BGP is the routing protocol of the global Internet, as well as for Service Provider private networks. BGP has expanded upon its original purpose of carrying Internet reachability information, and can now carry routes for Multicast, IPv6, VPNs, and a variety of other data.
- Traffic that is routed within a single network AS is referred to as internal BGP, or iBGP. More often, BGP is used to connect one AS to other autonomous systems, and it is then referred to as an external BGP, or eBGP.

- **Internet Control message Protocol (ICMP)**
- The IP protocol has no error-reporting or error-correcting mechanism. The IP protocol also lacks a mechanism for host and management queries.
- The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol.
- The ICMP provides a means for transferring message from routers and other hosts to a host.
- ICMP provides feedback about problems in the communication environment. ICMP uses the messages and these messages are divided into two types Error-reporting and Query message.
- One of the main responsibilities of ICMP is to report errors. ICMP always reports error message to the original source. Error reporting including the messages in additional to error reporting ICMP can diagnose some network problems. This accomplished through the query message, thus the query messages are used to diagnose the network problems. Each message type has their format.

ICMP Message Types



Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

Note: Choke packets are used for congestion and flow control over a network

A **timestamp** is a sequence of characters or encoded information identifying when a certain event occurred, usually giving date and time of day, sometimes accurate to a small fraction of a second.

- **Open Shortest Path First(OSPF)**
- Open Shortest Path First is a link state and hierarchical IGP routing algorithm.
- It is an enhanced version of RIP, which comprises features like multipath routing, least cost routing, and load balancing. Its major metric is the cost to determine the best path.
- OSPF involves the type of service routing which means multiple routes can be installed according to the priority or type of service.
- OSPF offers load balancing in which it distributes overall traffic routes equally. It also allows networks and routers partitioned into subsets and areas which enhance the growth and ease of management.
- OSPF is more efficient than RIP because it normally doesn’t use broadcast messages. Instead, it selectively sends status update messages directly to selected computers or routers. OSPF is the preferred interior routing protocol used by TCP/IP.

- **Routing Information Protocol (RIP)**
- RIP is a dynamic distance vector interior routing protocol that is commonly used in smaller networks, such as those operated by one organization. The network manager uses RIP to develop the routing table. When new computers are added, RIP simply counts the number of computers in the possible routes to the destination and selects the route with the least number. Computers using RIP send broadcast messages every minute or so (the timing is set by the network manager) announcing their routing status to all other computers. RIP is used by both TCP/IP and IPX/SPX.
- **Intermediate System to Intermediate System (IS-IS)**
- **IS-IS** is a link state interior routing protocol that is commonly used in large networks. IS-IS is an ISO protocol that has been added to many TCP/IP networks.

- **Enhanced Interior Gateway Routing Protocol (EIGRP)**
- EIGRP is a dynamic hybrid interior routing protocol developed by Cisco and is commonly used inside organizations. Hybrid means that it has some features that act like distance vector protocols and some other features that act like link state protocols. As you might expect, EIGRP is an improved version of Interior Gateway Routing Protocol (IGRP). EIGRP records information about a route’s transmission capacity, delay, reliability, and load. EIGRP is unique in that computers or routers store their own routing tables as well as the routing tables for all of their neighbors so they have a more accurate understanding of the network.

Multicasting

- The most common type of message in a network is the transmission between two computers. One computer sends a message to another computer (e.g., a client requesting a Web page). This is called a **unicast message**. Earlier in the chapter, we introduced the concept of a **broadcast message** that is sent to all computers on a specific LAN or subnet. A third type of message called a **multicast message** is used to send the same message to a group of computers.
- Consider a videoconferencing situation in which four people want to participate in the same conference. Each computer could send the same voice and video data from its camera to the computers of each of the other three participants using unicasts. In this case, each computer would send three identical messages, each addressed to the three different computers. This would work but would require a lot of network capacity. Alternately, each computer could send one broadcast message. This would reduce network traffic (because each computer would send only one message), but every computer on the network would process it, distracting them from other tasks. Broadcast messages usually are transmitted only within the same LAN or subnet, so this would not work if one of the computers were outside the subnet.

- The solution is multicast messaging. Computers wishing to participate in a multicast send a message to the sending computer or some other computer performing routing along the way using a special type of packet called **Internet Group Management Protocol (IGMP).** Each multicast group is assigned a special IP address to identify the group. Any computer performing routing knows to route all multicast messages with this IP address onto the subnet that contains the requesting computer.
- The routing computer sets the data link layer address on multicast messages to a matching multicast data link layer address. Each requesting computer must inform its data link layer software to process incoming messages with this multicast data link layer address. When the multicast session ends (e.g., the videoconference is over), the client computer sends another IGMP message to the organizing computer or the computer performing routing to remove it from the multicast group.

The Anatomy of a Router

- There is a huge array of software and hardware that makes the Internet work, but the one device that is indispensable is the router.
- The router has three main functions:

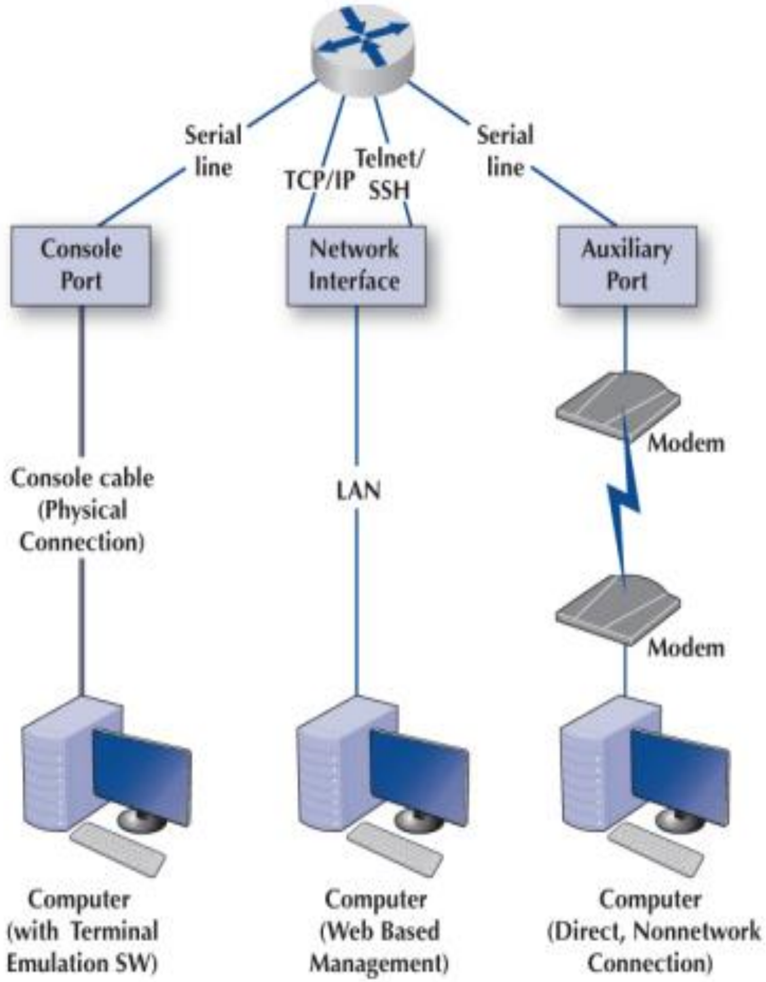
(1) it determines a path for a packet to travel over,

(2) it transmits the packet across the path, and

(3) it supports communication between a wide variety of devices and protocols.

- Routers are essentially special-purpose computers that consist of a CPU (central processing unit), memory (both volatile and nonvolatile), and ports or interfaces that connect to them to the network and/or other devices so that a network administrator can communicate with them.

- There are three ways that a network manager can connect to a router and configure and maintain it: *(1) console port, (2) network interface port, and (3) auxiliary port.*
- When the router is turned on for the very first time, it does not have an IP address assigned, so it cannot communicate on the network. Because of this, the console port, also called the management port, is used to configure it.
- A network manager would use a blue rollover cable (not the Ethernet cable) to connect the router's console port to a computer that has terminal emulation software on it. The network manager would use this software to communicate with the router and perform the basic setup (e.g., IP address assignment, routing protocol selection).

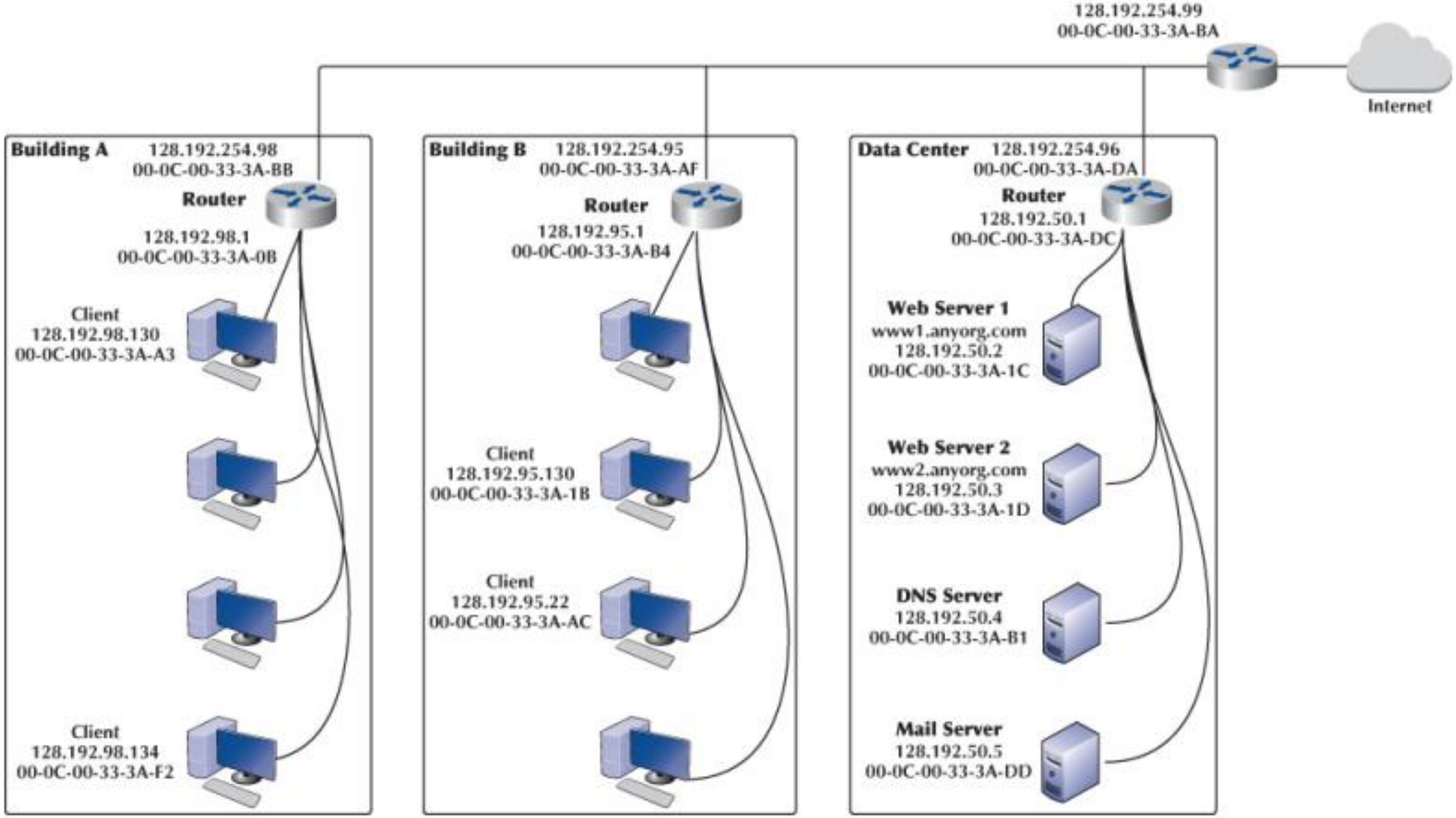


Serial line
TCP/IP
Telnet/ SSH
Serial line
Console Port
Network Interface
Auxiliary Port
Modem
Console cable (Physical Connection)
LAN
Modem
Computer (with Terminal Emulation SW)
Computer (Web Based Management)
Computer (Direct, Nonnetwork Connection)

- Once the basic setup is done, the network manager can log in to the router from any computer using the network interface using TCP/IP and Telnet with Secure Shell (SSH). Although routers come with an auxiliary port that allows an administrator to log via a direct, nonnetwork connection (e.g., using modems), this connection is rarely used today.
- A router, just like a computer, must have an operating system so that it can be configured. The operating system that is used in about 90% of routers is the Cisco Internetwork Operating Systems (IOS), although other operating systems exist as well. IOS uses a command line interface rather than a graphical user interface. The network manager uses IOS commands to create a configuration file (also called a config file) that defines how the router will operate. The config file can contain the type of routing protocol to be used, the interfaces that are active/enabled and those that are down, and what type of encryption is used. The config file is central to a router’s operation, and the IOS refers to it hundreds of times per second to tell the router how to do its job.
- The other important file is the Access Control List (ACL), which plays an important role in network security. The ACL defines what types of packets should be routed and what types of packets should be discarded.

TCP/IP Example

- As discussed the functions of the transport and network layers: linking to the application layer, segmenting, session management, addressing, and routing. Here we tie all of these concepts together to take a closer look at how these functions actually work using TCP/IP.
- When a computer is installed on a TCP/IP network (or dials into a TCP/IP network), it must be given four pieces of network layer addressing and routing information before it can operate. This information can be provided by a configuration file or via a DHCP server.
- The information is
- Its IP address
- A subnet mask, so it can determine what addresses are part of its subnet .
- The IP address of a DNS server, so it can translate application layer addresses into IP addresses .
- The IP address of an IP gateway (commonly called a router) leading outside of its subnet, so it can route messages addressed to computers outside of its subnet.


128.192.254.99
00-0C-00-33-3A-BA
Internet
Building A
128.192.254.98
00-0C-00-33-3A-BB
Router
128.192.98.1
00-0C-00-33-3A-0B
Client
128.192.98.130
00-0C-00-33-3A-A3
Client
128.192.98.134
00-0C-00-33-3A-F2
Building B
128.192.254.95
00-0C-00-33-3A-AF
Router
128.192.95.1
00-0C-00-33-3A-B4
Client
128.192.95.130
00-0C-00-33-3A-1B
Client
128.192.95.22
00-0C-00-33-3A-AC
Data Center
128.192.254.96
00-0C-00-33-3A-DA
Router
128.192.50.1
00-0C-00-33-3A-DC
Web Server 1
www1.anyorg.com
128.192.50.2
00-0C-00-33-3A-1C
Web Server 2
www2.anyorg.com
128.192.50.3
00-0C-00-33-3A-1D
DNS Server
128.192.50.4
00-0C-00-33-3A-B1
Mail Server
128.192.50.5
00-0C-00-33-3A-DD

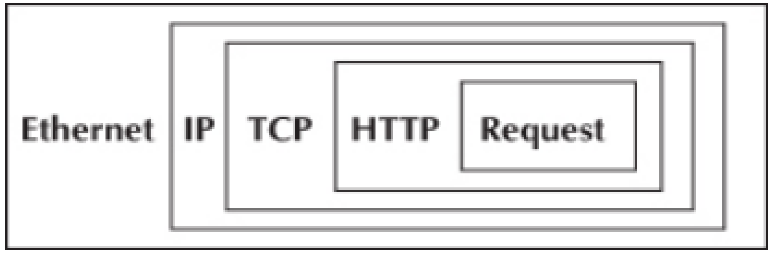
- These four pieces of information are the minimum required. A server would also need to know its application layer address.
- This figure shows an organization that has three LANs connected by a BN. The BN also has a connection to the Internet. Each building is configured as a separate subnet. For example, Building A has the 128.192.98.x subnet, whereas Building B has the 128.192.95.x subnet. The data center has the 128.192.50.x subnet. The BN is its own subnet: 128.192.254.x. Each building is connected to the BN via a router that has two IP addresses and two data link layer addresses, one for the connection into the building and one for the connection onto the BN. The organization has couple Web servers, one DNS server, and one Mail server located in the data center. All networks use Ethernet as the data link layer and only focus on Web requests at the application layer.

- In TCP/IP, it is important to remember that the TCP segments and IP packets are created by the sending computer and never change until the message reaches its final destination. The IP packet contains the original source and ultimate destination address for the packet. The sending computer also creates a data link layer frame (e.g., Ethernet) for each message. This frame contains the data link layer address of the current computer sending the packet and the data link layer address of the next computer in the route through the network.
- The data link layer frame is removed and replaced with a new frame at each computer at which the message stops as it works its way through the network. Thus, the source and destination data link layer addresses change at each step along the route, whereas the IP source and destination addresses never change.

Known Addresses

- Suppose that a client computer in Building A (e.g., 128.192.98.130) wanted to get a Web page from a Web server A located in the data center (www2.anyorg.com). We will assume that this computer knows the network layer and data link layer addresses of the Web server 1 (www1.anyorg.com) in the data center (e.g., it has previously requested pages from this server, so the addresses are stored in appropriate tables in the memory of the computer). Because the computer knows the IP address of the server, it uses its IP address, not its application layer address.
- In this case, the application layer software would pass an HTTP packet to the transport layer software (TCP) with the Internet address of the destination www1.anyorg.com: 128.192.50.2. The transport layer software (TCP) would make sure that the request fits in one segment and hands it to the network layer. The network layer software (IP) would then check the subnet mask and would recognize that the Web server is located outside of its subnet. Any messages going outside the subnet must be sent to the router (128.192.98.1), whose job it is to process the message and send the message on its way into the outside network.

- The network layer software would check its address table and find the Ethernet address for the router. It would therefore set the data link layer address to the router’s Ethernet address on this subnet (00-0C-00-33-3A-0B) and pass the IP packet to the data link layer for transmission. The data link layer would surround the frame with an Ethernet frame and transmit it over the physical layer to the Web server as shown in figure below:



Packet nesting. HTTP = Hypertext Transfer Protocol; IP = Internet Protocol; TCP = Transmission Control Protocol

- The router would receive the message and its data link layer would perform error checking before passing the packet to the network layer software (IP). The network layer software would read the IP address to determine the final destination. The router would recognize that this address (128.192.50.2) needed to be sent to the 128.192.50.x subnet. It knows that the router for this subnet is 128.192.254.98. It would pass the packet back to its data link layer, giving the Ethernet address of the router (00-0C-00-33-3A-DA).
- This router in the data center would receive the message (do error checking, etc.) and read the IP address to determine the final destination. The router would recognize that this address (128.192.50.2) was inside its 128.192.50.x subnet and would search its data link layer address table for this computer. It would then pass the packet to the data link layer along with the Ethernet address (00-0C-00-33-3A-1C) for transmission.

- The www1.anyorg.com Web server would receive the message and process it. This would result in a series of TCP/IP packets addressed to the requesting client (128.192.98.130). These would make their way through the network in reverse order. The Web server would recognize that this IP address is outside its subnet and would send the message to the 128.192.50.1 router using its Ethernet address (00-0C-00-33-3A-DC). This router would then send the message to the router for the 128.192.98.x subnet (128.192.254.98) using its Ethernet address (00-0C-00-33-3A-BB). This router would in turn send the message back to the client (128.192.98.130) using its Ethernet address (00-0C-00-33-3A-A3).
- This process would work in the same way for Web servers located outside the organization on the Internet. In this case, the message would go from the client to the 128.192.98.1 router, which would send it to the Internet router (128.192.254.99), which would send it to its Internet connection. The message would be routed through the Internet, from router to router, until it reached its destination. Then the process would work in reverse to return the requested page.

Unknown Addresses

- Suppose that the client computer in Building A (128.192.98.130) wants to retrieve a Web page from the www1.anyorg.com Web server but does not know the IP address of the Web server. For simplicity, we will start by assuming that the client knows the data link layer address of its subnet router, but after you read through this example, you will realize that obtaining a data link layer address is straightforward.
- The Web browser realizes that it does not know the IP address after searching its IP address table and not finding a matching entry. Therefore, it issues a DNS request to the name server (128.192.50.4). The DNS request is passed to the transport layer (TCP), which attaches a UDP datagram and hands the message to the network layer.
- Using its subnet mask, the network layer (IP) will recognize that the DNS server is outside of its subnet. It will attach an IP packet and set the data link layer address to its router’s address.

- The router will process the message and recognize that to reach the 128.192.50.4 IP address, it must send the packet to the data center router, 128.192.254.96 and does this by using this router’s MAC address (00-0-00-33-3A-DA). When the data center router receives this packet, it will realize that it has a direct connection to the network the DNS server is on and will transmit the packet using the DNS server’s Ethernet address (00-0C00-CC-3A-B1).
- The name server will process the DNS request and send the matching IP address back to the client via the 128.198.98.x subnet router. The IP address for the desired computer makes its way back to the application layer software, which stores it in its IP table.
- The application layer then issues the HTTP request using the IP address for the Web server (128.192.50.2) and passes it to the transport layer, which in turn passes it to the network layer. The network layer uses its subnet mask and recognizes that this computer is not on its subnet. Therefore, it will route the packet to its default gateway, 128.192.98.1, which will then send the HTTP request to the data center’s router, 128.192.254.96, which will deliver the HTTP request to Web server 1.

- This process works the same for a Web server outside the subnet, whether in the same organization or anywhere on the Internet. If the Web server is far away (e.g., Australia), the process will likely involve searching more than one name server, but it is still the same process.
- What would happen if client in building A (128.192.98.130) did not know its router’s Ethernet address, which it needs to send the message to the router? It would broadcast an ARP request to all computers on its subnet, requesting that the computer whose IP address is 128.192.98.1 respond with its Ethernet address.
- This request is processed by all computers on the subnet, but only the router responds with an ARP packet giving its Ethernet address. The network layer software on the client stores this address in its data link layer address table (called ARP cache). Then the client computer could send the message.

- This same ARP request/response process can occur at any point as a message moves through the network. For example, suppose that the router in the data center (128.192.254.96) did not know the Ethernet address of the DNS server (128.192.50.4). The DNS request would flow through the network in exactly the same way as described earlier (because no computer knows whether the router knows or doesn’t know the Ethernet address) until the DNS request arrived at the data center router. This router would try to address the message to the DNS server and would realize that it did not have the server’s Ethernet address, so it would issue an ARP request. The DNS server would respond with an ARP response containing its Ethernet address, and the router would put that address on the message and send it to the server.

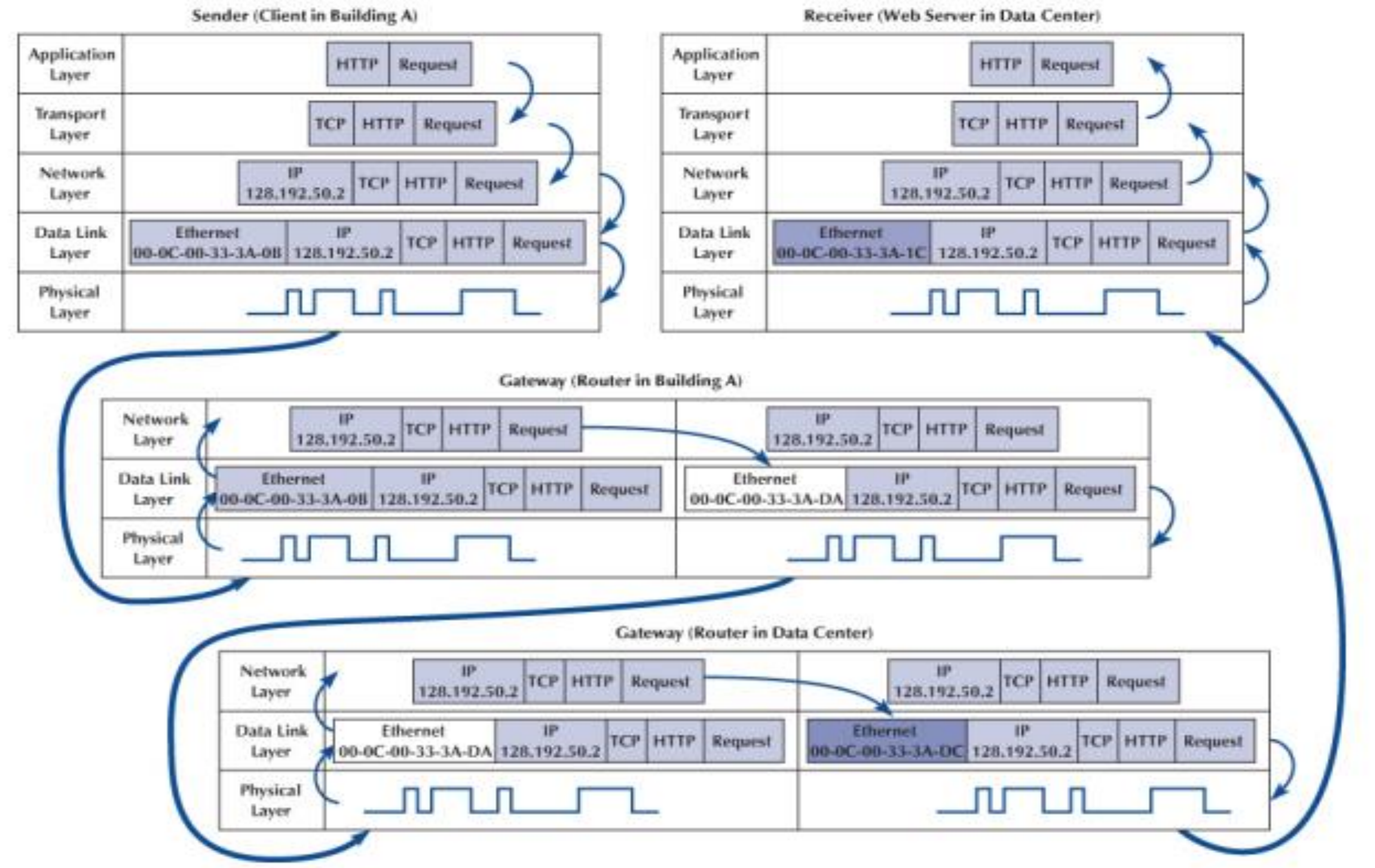
TCP Connections

- Whenever a computer transmits data to another computer, it must choose whether to use a connection-oriented service via TCP or a connectionless service via UDP. Most application layer software such as Web browsers (HTTP), email (SMTP), FTP, and Telnet use connection-oriented services. This means that before the first packet is sent, the transport layer first sends a SYN segment to establish a session (also known as the threeway handshake). Once the session is established, then the data packets begin to flow. Once the data are finished, the session is closed with a FIN segment (also known as the four-way handshake).
- In the preceding examples, this means that the first packet sent is really a SYN segment, followed by a response from the receiver accepting the connection, and then the packets as described earlier. There is nothing magical about the SYN and FIN segments; they are addressed and routed in the same manner as any other packets. But they do add to the complexity and length of the example.

- A special word is needed about HTTP packets. When HTTP was first developed, Web browsers opened a separate TCP session for each HTTP request. That is, when they requested a page, they would open a session, send the single packet requesting the Web page, and close the session at their end. The Web server would open a session, send as many packets as needed to transmit the requested page, and then close the session. If the page included graphic images, the Web browser would open and close a separate session for each request. This requirement to open and close sessions for each request was time consuming and not really necessary. With the newest version of HTTP, Web browsers open one session when they first issue an HTTP request and leave that session open for all subsequent HTTP requests to the same server.

TCP/IP and Network Layers

- In the network model we will look at how messages flow through the layers. Figure shows how a Web request message from a client computer in Building A would flow through the network layers in the different computers and devices on its way to the Web server (www1.anyorg.com, 128.192.50.2) in the Data Center.


Sender (Client in Building A)
Application Layer
HTTP | Request
Transport Layer
TCP | HTTP | Request
Network Layer
IP 128.192.50.2 | TCP | HTTP | Request
Data Link Layer
Ethernet 00-0C-00-33-3A-0B | IP 128.192.50.2 | TCP | HTTP | Request
Physical Layer
Receiver (Web Server in Data Center)
Application Layer
HTTP | Request
Transport Layer
TCP | HTTP | Request
Network Layer
IP 128.192.50.2 | TCP | HTTP | Request
Data Link Layer
Ethernet 00-0C-00-33-3A-1C | IP 128.192.50.2 | TCP | HTTP | Request
Physical Layer
Gateway (Router in Building A)
Network Layer
IP 128.192.50.2 | TCP | HTTP | Request
Data Link Layer
Ethernet 00-0C-00-33-3A-0B | IP 128.192.50.2 | TCP | HTTP | Request
Ethernet 00-0C-00-33-3A-DA | IP 128.192.50.2 | TCP | HTTP | Request
Physical Layer
Gateway (Router in Data Center)
Network Layer
IP 128.192.50.2 | TCP | HTTP | Request
Data Link Layer
Ethernet 00-0C-00-33-3A-DA | IP 128.192.50.2 | TCP | HTTP | Request
Ethernet 00-0C-00-33-3A-DC | IP 128.192.50.2 | TCP | HTTP | Request
Physical Layer

How messages move through the network layers.

***Note*: The addresses in this example are destination addresses**

- The message starts at the application layer of the sending computer (the client in Building A, 128.192.98.130), shown in the upper left corner of the figure, which generates an HTTP packet. This packet is passed to the transport layer, which surrounds the HTTP packet with a TCP segment. This is then passed to the network layer, which surrounds it with an IP frame that includes the IP address of the final destination (128.192.50.2). This in turn is passed to the data link layer, which surrounds it within an Ethernet frame that also includes the Ethernet address of the next computer to which the message will be sent (00-0C-00-33- 3A-0B). Finally, this is passed to the physical layer, which converts it into electrical impulses for transmission through the cable to its next stop—the router that serves as the gateway in Building A.
- When the message arrives at the router in Building A, its physical layer translates it from electrical impulses into digital data and passes the Ethernet frame to the data link layer. The data link layer checks to make sure that the Ethernet frame is addressed to the router, performs error detection, strips off the Ethernet frame, and passes its contents (the IP packet) to the network layer.

- The routing software running at the network layer looks at the destination IP address, determines the next computer to which the packet should be sent, and passes the outgoing packet down to the data link layer for transmission. The data link layer surrounds the IP packet with a completely new Ethernet frame that contains the destination address of the next computer to which the packet will be sent (00-0C-00-33- 3A-DA). In Figure, this new frame is shown in a different color. This is then passed to the physical layer, which transmits it through the network cable to its next stop—the router that serves as the gateway in the Data Center.
- When the message arrives at the router in the Data Center, it goes through the same process. The physical layer passes the incoming packet to the data link layer, which checks the destination Ethernet address, performs error detection, strips off the Ethernet frame, and passes the IP packet to the network layer software. The software determines the next destination and passes the IP packet back to the data link layer, which adds a completely new Ethernet frame with the destination address of its next stop (00-0C-00-33-3A-DC)— its final destination.

- The physical layer at the server receives the incoming packet and passes it to the data link layer, which checks the Ethernet address, performs error detection, removes the Ethernet frame, and passes the IP packet to the network layer. The network layer examines the final destination IP address on the incoming packet and recognizes that the server is the final destination. It strips off the IP packet and passes the TCP segment to the transport layer, which in turn strips off the TCP segment and passes the HTTP packet to the application layer (the Web server software).
- There are two important things to remember from this example. First, at all gateways (i.e., routers) along the way, the packet moves through the physical layer and data link layer up to the network layer, but no higher. The routing software operates at the network layer, where it selects the next computer to which the packet should be sent, and passes the packet back down through the data link and physical layers. These three layers are involved at all computers and devices along the way, but the transport and application layers are only involved at the sending computer (to create the application layer packet and the TCP segment) and at the receiving computer (to understand the TCP segment and process the application layer packet). Inside the TCP/IP network itself, messages only reach layer 3—no higher.

- Second, at each stop along the way, the Ethernet frame is removed and a new one is created. The Ethernet frame lives only long enough to move the message from one computer to the next and then is destroyed. In contrast, the IP packet and the packets above it (TCP and application layer) never change while the message is in transit. They are created and removed only by the original message sender and the final destination.