5) TCP/IP Protocols

5.1 Network Access/Link layer protocols :Ethernet, Token Ring, Network access to Internet layer

Mapping: ARP and RARP protocol

Network Access/Link layer protocols :Ethernet, Token Ring

Ethernet:

1. Ethernet is a family of computer networking technologies for local area networks (LANs) and metropolitan area networks (MANs).
2. It is the most popular LAN technology in the world. It is an easy, relatively inexpensive way to provide high-performance networking to all different types of computer equipment.
3. Ethernet was invented at Xerox PARC and developed jointly by Digital Equipment Corporation, Intel and Xerox.
4. Ethernet has largely replaced competing wired LAN technologies such as token ring, FDDI, and ARCNET.
5. The Ethernet standards comprise several wiring and signaling variants of the OSI physical layer in use with Ethernet.
6. Over time Ethernet data transfer rates have been increased from the original three megabits per second (Mbit/s) to the latest 100 gigabits per second (Gbit/s)
7. Systems communicating over Ethernet divide a stream of data into shorter pieces called frames. Each frame contains source and destination addresses and error-checking data so that damaged data can be detected and re-transmitted.
8. As per the OSI model, Ethernet provides services up to and including the data link layer.
9. The CSMA/CD approach is used by any form of Ethernet operating in half-duplex mode-that is, the mode in which transmit (Tx) and receive (Rx) signals can be sent on the same wire or data path. In full-duplex mode, transmit and receive signals are separated onto dedicated, one-way channels.
10. The original Ethernet was created in 1976 at Xerox's Palo Alto Research Center (PARC). Since then, it has gone through four generations: Standard Ethernet (lot Mbps), Fast Ethernet (100 Mbps), Gigabit Ethernet (l Gbps), and Ten-Gigabit Ethernet (l0 Gbps), as shown in Figure.

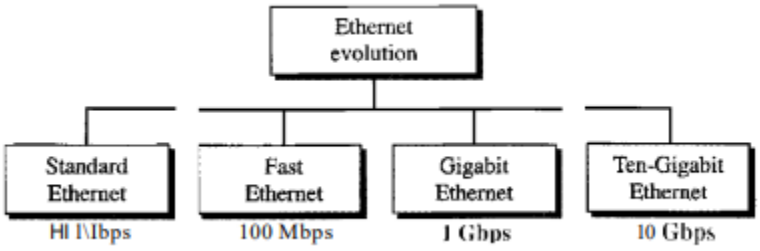


Figure: Ethernet evolution through four generations

Token Ring:

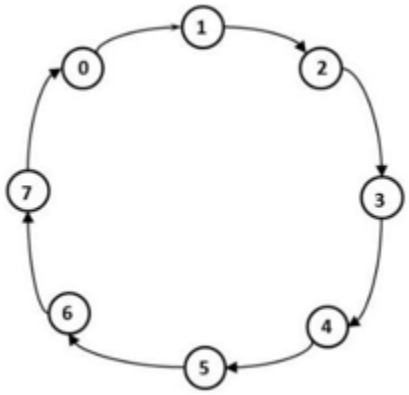


Figure: Token Ring

1. In this algorithm it is assumed that all the processes in the system are organized in a logical ring. The figure below describes the structure.
2. The ring positions may be allocated in numerical order of network addresses and is unidirectional in the sense that all messages are passed only in clockwise or anti-clockwise direction.
3. When a process sends a request message to the current coordinator and does not receive a reply within a fixed timeout, it assumes the coordinator has crashed. It then initializes the ring and process Pi is given a token.
4. The token circulates around the ring. It is passed from process k to k+1 in point to point messages. When a process acquires the token from its neighbor it checks to see if it is attempting to enter a critical region. If so the process enters the region does all the execution and leaves the region. After it has exited it passes the token along the ring. It is not permitted to enter a second critical region using the same token.
5. If a process is handed the token by its neighbor and is not interested in entering a critical region it just passes along. When no processes want to enter any critical regions the token just circulates at high speed around the ring.
6. Only one process has the token at any instant so only one process can actually be in a critical region. Since the token circulates among the process in a well-defined order, starvation cannot occur.
7. Once a process decides it wants to enter a critical region, at worst it will have to wait for every other process to enter and leave one critical region.
8. The disadvantage is that if the token is lost it must be regenerated. But the detection of lost token is difficult. If the token is not received for a long time it might not be lost but is in use.

Network access to Internet layer Mapping: ARP and RARP protocol

ARP Protocol:

1. ARP stands for Address Resolution Protocol. It is used for mapping Logical to Physical Address.The Address Resolution Protocol (ARP) is used to associate a logical address with a physical address.
2. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address, usually imprinted on the network interface card (NIC). ARP is used to find the physical address of the node when its Internet address is known.

ARP Operation

1. Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver. The logical (IP) address is obtained from the DNS if the sender is the host or it is found in a routing table if the sender is a router. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network.
2. This means that the sender needs the physical address of the receiver. The host or the router sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network **(see Figure).**
3. Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and physical addresses. The packet is unicast directly to the inquirer by using the physical address received in the query packet.

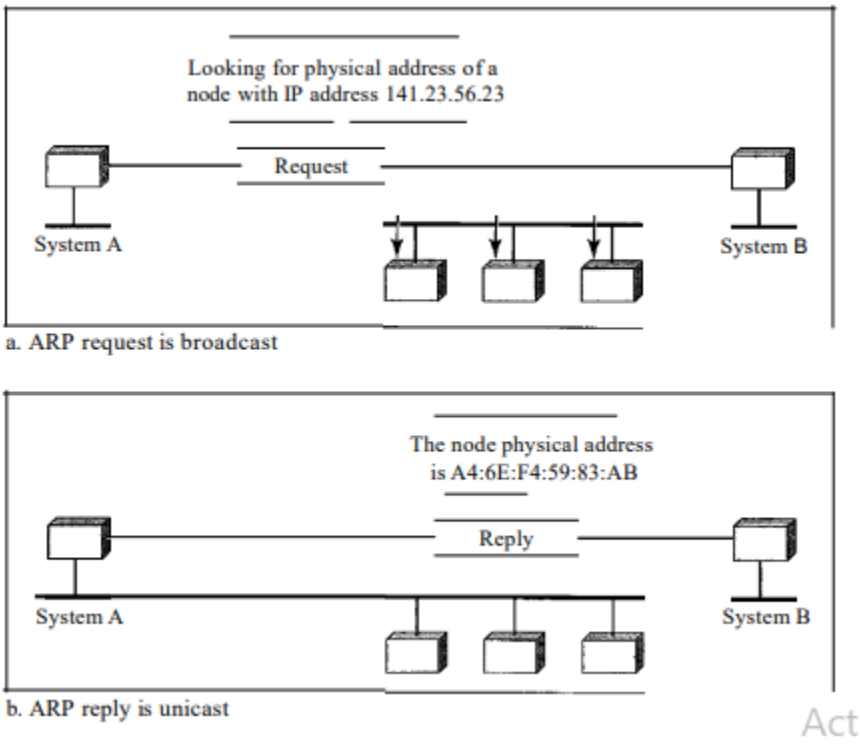


Figure a & b: ARP Operation

4. **In Figure a,** the system on the left (A) has a packet that needs to be delivered to another system (B) with IP address 141.23.56.23. System A needs to pass the packet to its data link layer for the actual delivery, but it does not know the physical address of the recipient. It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IF address of 141.23.56.23.
5. This packet is received by every system on the physical network, but only system B will answer it, as shown **in Figure b.** System B sends an ARP reply packet that includes its physical address. Now system A can send all the packets it has for this destination by using the physical address it received.

Cache Memory

1. Using ARP is inefficient if system A needs to broadcast an ARP request for each IP packet it needs to send to system B. It could have broadcast the IP packet itself. ARP can be useful if the ARP reply is cached (kept in cache memory for a while) because a system normally sends several packets to the same destination.
2. A system that receives an ARP reply stores the mapping in the cache memory and keeps it for 20 to 30 minutes unless the space in the cache is exhausted. Before sending an ARP request, the system first checks its cache to see if it can find the mapping.

Packet Format

1. **Figure shows the format of an ARP packet.**

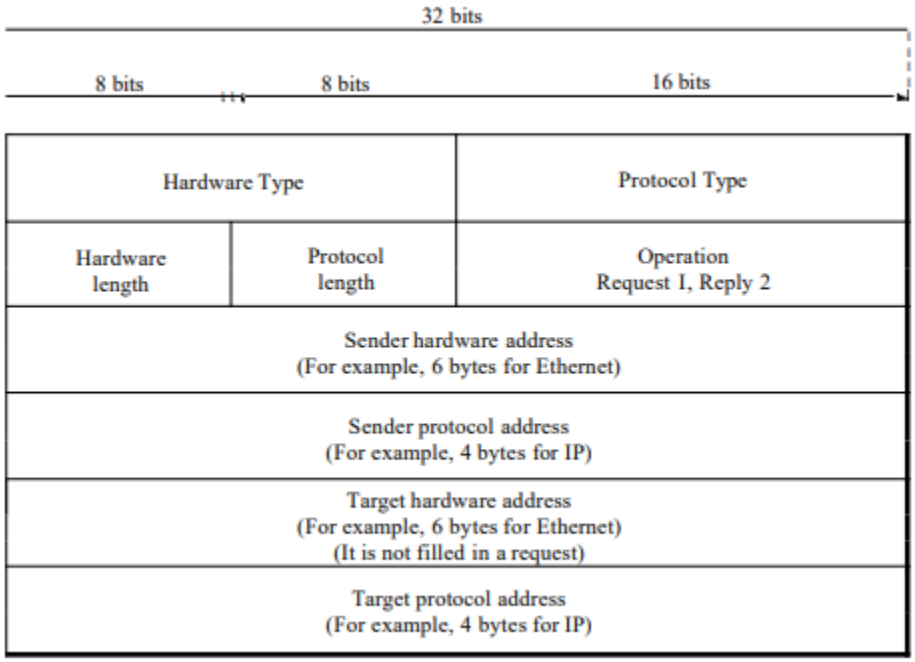


Figure: ARP Packet Format

2. **The fields are as follows:**
    - **Hardware type:** This is a 16-bit field defining the type of the network on which ARP is running. Each LAN has been assigned an integer based on its type. For example, Ethernet is given type 1. ARP can be used on any physical network.
    - **Protocol type:** This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 080016, ARP can be used with any higher-level protocol.
    - **Hardware length:** This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.
    - **Protocol length:** This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.
    - **Operation:** This is a 16-bit field defining the type of packet. Two packet types are defined: ARP request (1) andARP reply (2).
    - **Sender hardware address:** This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.
    - **Sender protocol address:**This is a variable-length field defining the logical (for example, IP) address of the sender. For the IP protocol, this field is 4 bytes long.
    - **Target hardware address:** This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long. For an ARP request message, this field is all Os because the sender does not know the physical address of the target.
    - **Target protocol address:** This is a variable-length field defining the logical (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long.

Encapsulation

1. An ARP packet is encapsulated directly into a data link frame. For example, in Figure, an ARP packet is encapsulated in an Ethernet frame.
2. Note that the type field indicates that the data carried by the frame are an ARP packet.

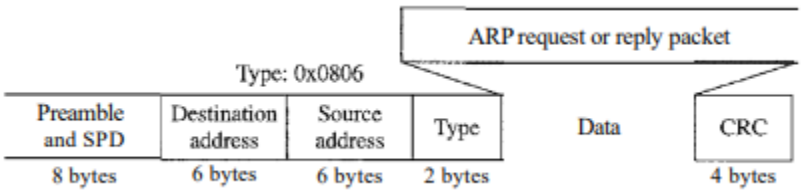
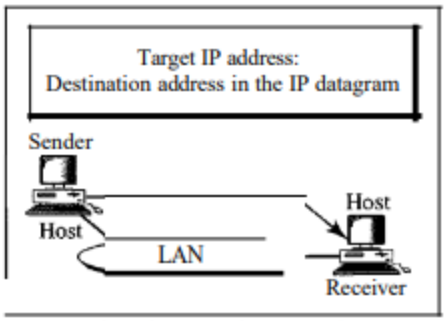


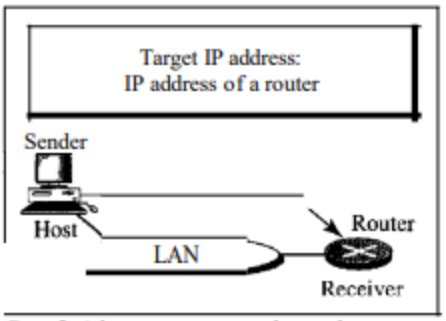
Figure: Encapsulation ofARP packet

Four Different Cases

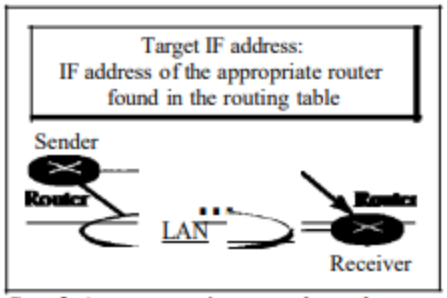
The following are four different cases in which the services of ARP can be used (see Figure).



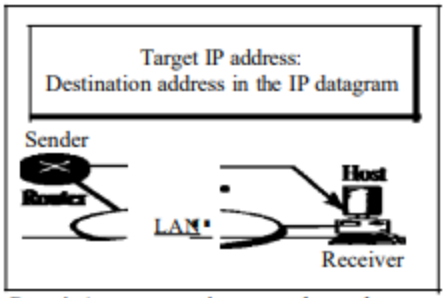
Case 1. A host has a packet to send to another host on the same network.



Case 2. A host wants to send a packet to another host on another network. It must first be delivered to a router.



Case 3. A router receives a packet to be sent to a host on another network. It must first be delivered to the appropriate router.



Case 4. A router receives a packet to be sent to a host on the same network.

RARP protocol:

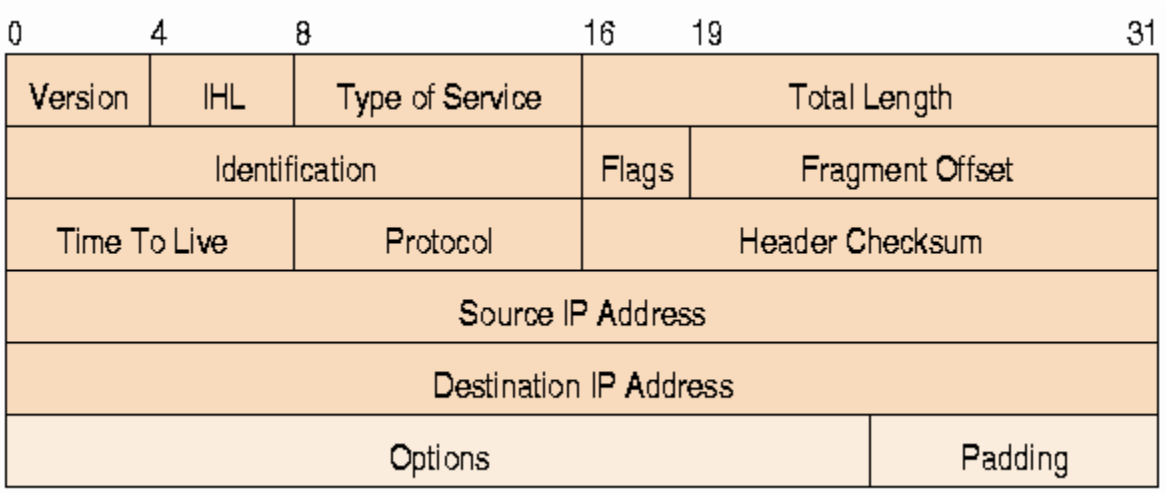
1. RARP Reverse Address Resolution Protocol (RARP) finds the logical address for a machine that knows only its physical address.
2. Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the physical (hardware) address of the machine.
3. To create an IP datagram, a host or a router needs to know its own IP address or addresses. The IP address of a machine is usually read from its configuration file stored on a disk file.
4. However, a diskless machine is usually booted from ROM, which has minimum booting information. The ROM is installed by the manufacturer. It cannot include the IP address because the IP addresses on a network are assigned by the network administrator.
5. The machine can get its physical address (by reading its NIC, for example), which is unique locally. It can then use the physical address to get the logical address by using the RARP protocol.
6. A RARP request is created and broadcast on the local network. Another machine on the local network that knows all the IP addresses will respond with a RARP reply.
7. The requesting machine must be running a RARP client program; the responding machine must be running a RARP server program. There is a serious problem with RARP: Broadcasting is done at the data link layer.
8. The physical broadcast address, allis in the case of Ethernet, does not pass the boundaries of a network.
9. This means that if an administrator has several networks or several subnets, it needs to assign a RARP server for each network or subnet.
10. This is the reason that RARP is almost obsolete. The two protocols, BOOTP and DHCp, are replacing RARP.

5.2 Internet Layer: IP Protocol, IP Address, Classful and Classless Addressing, IPV4 and IPV6 Protocol. DHCP Protocol,Network Address Translation(NAT) protocol, ICMP protocol.

Internet Protocol (IP Protocol) -

1. It is a connectionless datagram protocol.
2. It is a host to host network layer delivery protocol designed for the internet.
3. It is a set of requirements for addressing and routing data on the Internet.
4. It is an unreliable protocol because it does not provide any error or flow control.
5. is the method or protocol by which data is sent from one computer to another on the internet.

Header Format of Internet Protocol



1. **Version :-** It is the version of IP Protocol. It's bit value is 4 in static.
2. **Header length :-** It is the length of the header in 32-bit words. Minimum is 20 bytes and maximum is 60 bytes.
3. **Types of Service :-** It specifies how the datagram should be handled.
4. **Total Length :-** It is the length of the entire (header + data). The minimum length is 20 bytes, and the maximum is 65537 bytes.
5. **Identification :-** It is used to differentiate fragmented packets from different datagrams.
6. **Flags :-** It is used to control or identify fragments.
7. **Fragmented Offset :-** It is used for fragmentation and reassembly if the packet is too large to put in a frame.
8. **Time to Live :-** It limits a datagram's lifetime.
9. **Protocol :-** It defines the protocol used in the data portion of the IP datagram.
10. **Header Checksum :-** It is used for error-checking of the header. If a packet arrives at a router and the router calculates a different checksum than the one specified in this field, the packet will be discarded.
11. **Source IP address :-** It is the IP address of the host (sender).

12. **Destination IP address :-** It is the IP address of the host (receiver).
13. **Options :-** It is used for network testing. It is usually not required for every datagram.
14. **Padding :-** It is variable size bit field.

Functions of Internet Protocol -

1. **Reassembly :-** Internet Protocol keeps track of the way messages between computers are broken into packets. Since most messages are too big to fit in one packet, and since packets aren't sent in any organized order, they must be reassembled as they arrive at the recipient.
2. **Addressing :-** IP packet headers contain addresses that identify the sending computer and the receiving computer. Routers use this information to guide each packet across communication networks and connect the sending and receiving computers.
3. **Options :-** IP includes optional features such as allowing the sending computer to decide the path its packets take to get to the receiving computer, to trace the path they take or to include added security in the packets.
4. **Timeouts :-** Each IP packet contains a self-destructive counter that limits its lifetime. If a packet's defined lifetime expires, the packet is destroyed so that the Internet doesn't get overloaded with broken packets wandering aimlessly.
5. **Types of Service :-** IP supports traffic prioritization by allowing packets to be labeled with an abstract type of service.

IPv4 :

1. The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols. IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service.
2. The term best-effort means that IPv4 provides no error control or flow control (except for error detection on the header). IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees. If Reliability is important, IPv4 must be paired with a reliable protocol such as TCP.
3. An example of a more commonly understood best-effort delivery service is the post office. The post office does its best to deliver the mail but does not always succeed. If an unregistered letter is lost, it is up to the sender or would-be recipient to discover the loss and rectify the problem.
4. The post office itself does not keep track of every letter and cannot notify a sender of loss or damage. IPv4 is also a connectionless protocol for a packet-switching network that uses the datagram approach.
5. This means that each datagram is handled independently, and each datagram can follow a different route to the destination. This implies that datagrams sent by the same source to the same destination could arrive out of order. Also, some could be lost or corrupted during transmission. Again, IPv4 relies on a higher-level protocol to take care of all these problems.

Datagram

1. Packets in the IPv4 layer are called datagrams. Figure shows the IPv4 datagram format.
2. A datagram is a variable-length packet consisting of two parts: header and data. The header is 20 to 60 bytes in length and contains information essential to routing and delivery.

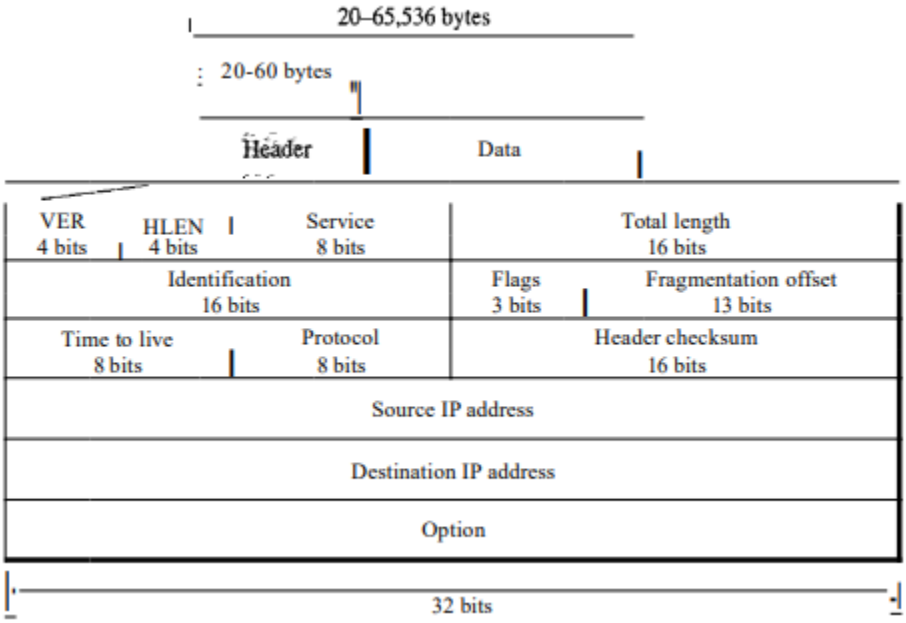


Figure: IPv4 datagram format

3. **It is customary in TCP/IP to show the header in 4-byte sections. A brief description of each field is in order.**
    - **Version (VER):** This 4-bit field defines the version of the IPv4 protocol. Currently the version is 4. However, version 6 (or IPng) may totally replace version 4 in the future. This field tells the IPv4 software running in the processing machine that the datagram has the format of version 4. All fields must be interpreted as specified in the fourth version of the protocol. If the machine is using some other version of IPv4, the datagram is discarded rather than interpreted incorrectly.
    - **Header length (HLEN):** This 4-bit field defines the total length of the datagram header in 4-byte words. This field is needed because the length of the header is variable (between 20 and 60 bytes). When there are no options, the header length is 20 bytes, and the value of this field is 5 (5 x 4 = 20). When the option field is at its maximum size, the value of this field is 15 (15 x 4 = 60).
    - **Services:** IETF has changed the interpretation and name of this 8-bit field. This field, previously called service type, is now called differentiated services. We show both interpretations in Figure.

1. **Service Type**

- In this interpretation, the first 3 bits are called precedence bits. The next 4 bits are called type of service (TOS) bits and the last bit is not used.

    a. Precedence is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary). The precedence defines the priority of the datagram in issues such as congestion. If a router is congested and needs to discard some datagrams, those datagrams with lowest

precedence are discarded first. Some datagrams in the Internet are more important than others.

b. TOS bits are a 4-bit subfield with each bit having a special meaning. Although a bit can be either 0 or 1, one and only one of the bits can have the value of 1 in each datagram.The bit patterns and their interpretations are given in Table . With only 1 bit set at a time, we can have five different types of services.

TOS Bits	*Description*
0000	Normal (default)
0001	Minimize cost
0010	Maximize reliability
0100	Maximize throughput
1000	Minimize delay

Figure: Types of service

- Application programs can request a specific type of service. The defaults for some applications are shown in Table.

Protocol	*TOS Bits*	*Description*
ICMP	0000	Normal
BOOTP	0000	Normal
NNTP	0001	Minimize cost
IGP	0010	Maximize reliability
SNMP	0010	Maximize reliability
TELNET	1000	Minimize delay
FTP (data)	0100	Maximize throughput
FTP (control)	1000	Minimize delay
TFTP	1000	Minimize delay
SMTP (command)	1000	Minimize delay
SMTP (data)	0100	Maximize throughput
DNS (UDP query)	1000	Minimize delay
DNS (TCP query)	0000	Normal
DNS (zone)	0100	Maximize throughput

Figure: Default types of service

2. Differentiated Services

In this interpretation, the first 6 bits make up the codepoint subfield, and the last 2 bits are not used. The codepoint subfield can be used in two different ways.

a. When the 3 rightmost bits are Os, the 3 leftmost bits are interpreted the same as the precedence bits in the service type interpretation. In other words, it is compatible with the old interpretation.

b. When the 3 rightmost bits are not all Os, the 6 bits define 64 services based on the priority assignment by the Internet or local authorities according to Table 20.3. The first category contains 32 service types; the second and the third each contain 16. The first category (numbers 0, 2,4, ... ,62) is assigned by the Internet authorities (IETF). The second category (3, 7, 11, 15, , 63) can be used by local authorities (organizations). The third category (1, 5, 9, ,61) is temporary and can be used for experimental purposes. Note that the numbers are not contiguous. Ifthey were, the first category would range from 0 to 31, the second from 32 to 47, and the third from 48 to 63. This would be incompatible with the TOS interpretation because XXXOOO (which includes 0, 8, 16, 24, 32, 40, 48, and 56) would fall into all three categories. Instead, in this assignment method all these services belong to category 1. Note that these assignments have not yet been finalized.

Category	*Codepoint*	*Assigning Authority*
1	XXXXXO	Internet
2	XXXX11	Local
3	XXXXOI	Temporary or experimental

Figure: Values for codepoints

- **Total length:** This is a In-bit field that defines the total length (header plus data) of the IPv4 datagram in bytes. To find the length of the data coming from the upper layer, subtract the header length from the total length. The header length can be found by multiplying the value in the HLEN field by 4. Length of data =total length - header length Since the field length is 16 bits, the total length of the IPv4 datagram is limited to 65,535 (216 - 1) bytes, of which 20 to 60 bytes are the header and the rest is data from the upper layer. The total length field defines the total length of the datagram including the header.
- **Identification:** This 16 - bit field is used in fragmentation.
- **Flags:** This 3-bit field is used in fragmentation.
- **Fragmentation offset.** This 13-bit field is used in fragmentation .
- **Time to live:** A datagram has a limited lifetime in its travel through the internet. This 8-bit field was originally designed to hold a timestamp, which was decremented by each visited router.
- **Protocol:** This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer. An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP.

Value	*Protocol*
1	ICMP
2	IGMP
6	TCP
17	UDP
89	OSPF

Figure: Protocol Values

- **Checksum:** The checksum 16 bit defines header checksum.
- **Source address:** This 32-bit field defines the IPv4 address of the source. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.
- **Destination address:** This 32-bit field defines the IPv4 address of the destination. This field must remain unchanged during the time the IPv4 datagram travels from the source host to the destination host.

Fragmentation

1. A datagram can travel through different networks. Each router decapsulates the IPv4 datagram from the frame it receives, processes it, and then encapsulates it in another frame.
2. The format and size of the received frame depend on the protocol used by the physical network through which the frame has just traveled.
3. The format and size of the sent frame depend on the protocol used by the physical network through which the frame is going to travel.
4. For example, if a router connects a LAN to a WAN, it receives a frame in the LAN format and sends a frame in the WAN format.

Checksum

1. The implementation of the checksum in the IPv4 packet follows the same principles. First, the value of the checksum field is set to O. Then the entire header is divided into 16-bit sections and added together.
2. The result (sum) is complemented and inserted into the checksum field. The checksum in the IPv4 packet covers only the header, not the data. There are two good reasons for this.
3. First, all higher-level protocols that encapsulate data in the IPv4 datagram have a checksum field that covers the whole packet. Therefore, the checksum for the IPv4 datagram does not have to check the encapsulated data.
4. Second, the header of the IPv4 packet changes with each visited router, but the data do not. So the checksum includes only the part that has changed. If the data were included, each router must recalculate the checksum for the whole packet, which means an increase in processing time.

5. Example checksum calculation for an IPv4 header without options. The header is divided into 16-bit sections. All the sections are added and the sum is complemented. The result is inserted in the checksum field.

Options

1. The header of the IPv4 datagram is made of two parts: a fixed palt and a variable part. The fixed part is 20 bytes long and was discussed in the previous section.
2. The variable part comprises the options that can be a maximum of 40 bytes. Options, as the name implies, are not required for a datagram. They can be used for network testing and debugging.

#End ofOption

An end-of-option option is a l-byte option used for padding at the end of the option field. It, however, can only be used as the last option.

#Record Route

A record route option is used to record the Internet routers that handle the datagram. It can list up to nine router addresses. It can be used for debugging and management purposes.

#Strict Source Route

A strict source route option is used by the source to predetermine a route for the datagram as it travels through the Internet. Dictation of a route by the source can be useful for several purposes.

#Loose Source Route

A loose source route option is similar to the strict source route, but it is less rigid. Each router in the list must be visited, but the datagram can visit other routers as well.

#Timestamp

1. A timestamp option is used to record the time of datagram processing by a router. The time is expressed in milliseconds from midnight, Universal time or Greenwich mean time. Knowing the time a datagram is processed can help users and managers track the behavior of the routers in the Internet.
2. We can estimate the time it takes for a datagram to go from one ~outer to another. We say estimate because, although all routers may use Universal time, their local clocks may not be synchronized.

IPv6

1. IPv4 has some deficiencies (listed below) that make it unsuitable for the fast-growing Internet.
   - Despite all short-term solutions, such as subnetting, classless addressing, and NAT, address depletion is still a long-term problem in the Internet.
   - The Internet must accommodate real-time audio and video transmission. This type of transmission requires minimum delay strategies and reservation of resources not provided in the IPv4 design.
   - The Internet must accommodate encryption and authentication of data for some applications. No encryption or authentication is provided by IPv4.
2. To overcome these deficiencies, IPv6 (Internetworking Protocol, version 6), also known as IPng (Internetworking Protocol, next generation), was proposed and is now a standard.
3. In IPv6, the Internet protocol was extensively modified to accommodate the unforeseen growth of the Internet.
4. The format and the length of the IP address were changed along with the packet format. Related protocols, such as ICMP, were also modified. Other protocols in the network layer, such as ARP, RARP, and IGMP, were either deleted or included in the ICMPv6 protocol.

Packet Format

1. The IPv6 packet is shown in Figure 20.15. Each packet is composed of a mandatory base header followed by the payload.
2. The payload consists of two parts: optional extension headers and data from an upper layer. The base header occupies 40 bytes, whereas the extension headers and data from the upper layer contain up to 65,535 bytes of information.

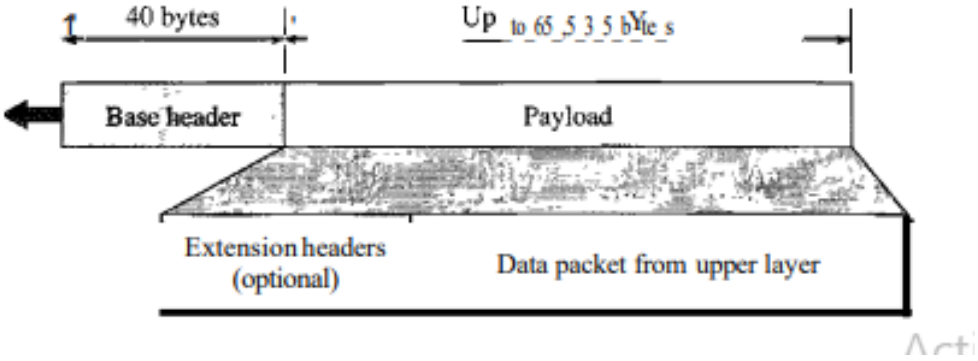


Figure: IPv6 datagram header and payload

Base Header

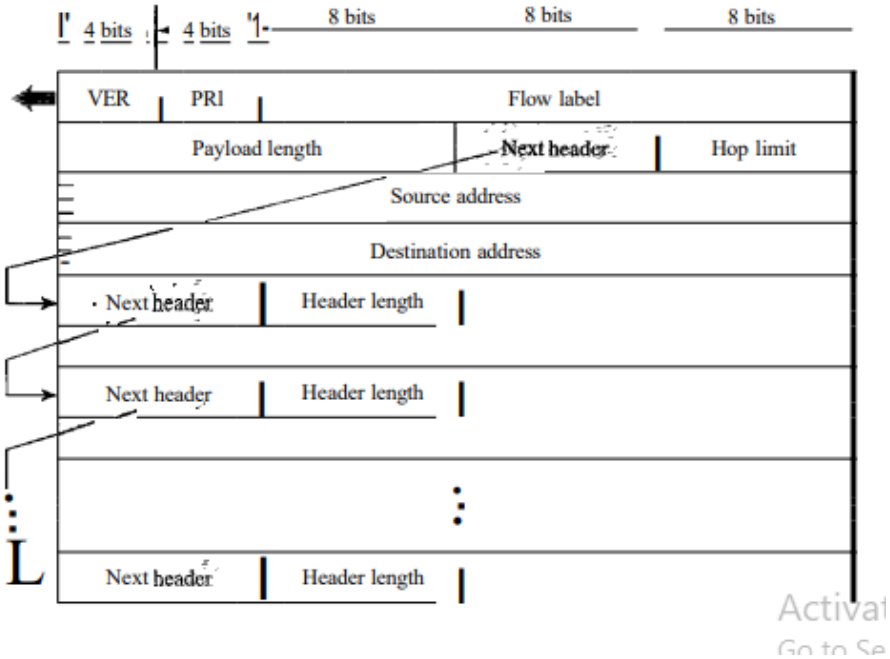


Figure:

Figure shows the base header with its eight fields. These fields are as follows:

1. **Version.** This 4-bit field defines the version number of the IP. For IPv6, the value is 6.
2. **Priority.** The 4-bit priority field defines the priority of the packet with respect to traffic congestion. We will discuss this field later.
3. **Flow label.** The flow label is a 3-byte (24-bit) field that is designed to provide special handling for a particular flow of data.
4. **Payload length.** The 2-byte payload length field defines the length of the IP datagram excluding the base header.
5. **Next header.** The next header is an 8-bit field defining the header that follows the base header in the datagram. The next header is either one of the optional extension headers used by IP or the header of an encapsulated packet such as UDP or TCP. Each extension header also contains this field. Note that this field in version 4 is called the protocol.

Code	*Next Header*
0	Hop-by-hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

6. **Hop limit.** This 8-bit hop limit field serves the same purpose as the TIL field in IPv4.
7. **Source address.** The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram.
8. **Destination address.** The destination address field is a 16-byte (128-bit) Internet address that usually identifies the final destination of the datagram. However, if source routing is used, this field contains the address of the next router.

Priority

1. The priority field of the IPv6 packet defines the priority of each packet with respect to other packets from the same source.
2. For example, if one of two consecutive datagrams must be discarded due to congestion, the datagram with the lower packet priority will be discarded. IPv6 divides traffic into two broad categories: congestion-controlled and noncongestion-controlled.

Congestion-Controlled

1. Traffic If a source adapts itself to traffic slowdown when there is congestion, the traffic is referred to as congestion-controlled traffic.
2. For example, TCP, which uses the sliding window protocol, can easily respond to traffic. In congestion-controlled traffic, it is understood that packets may arrive delayed, lost, or out of order.
3. Congestion-controlled data are assigned priorities from 0 to 7, as listed in Table.
4. A priority of 0 is the lowest; a priority of 7 is the highest.

Priority	*Meaning*
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Attended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

Figure: Priorities For congestion-controlled traffic

Extension Headers

1. The length of the base header is fixed at 40 bytes. However, to give greater functionality to the IP datagram, the base header can be followed by up to six extension headers.
2. Many of these headers are options in IPv4. Six types of extension headers have been defined, as shown in Figure.

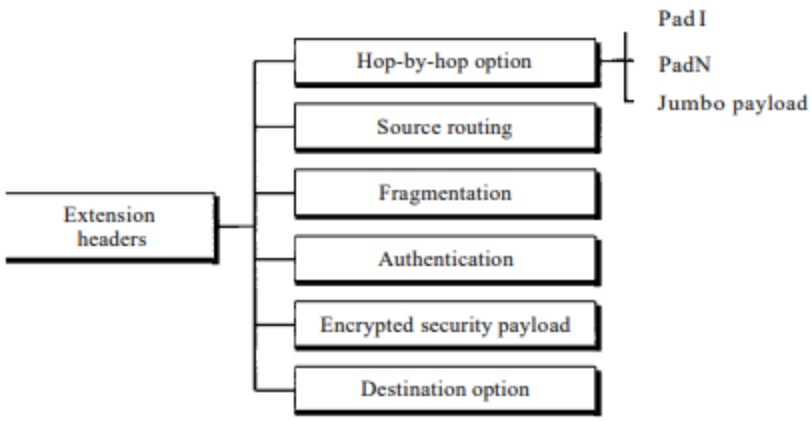


Figure: Extension header types

- **Hop-by-Hop Option**: The hop-by-hop option is used when the source needs to pass information to all routers visited by the datagram. So far, only three options have been defined: PadI, PadN, and jumbo payload. The PadI option is 1 byte long and is designed for alignment purposes. PadN is similar in concept to Padi. The difference is that PadN is used when 2 or more bytes is needed for alignment. The jumbo payload option is used to define a payload longer than 65,535 bytes.
- **Source Routing:** The source routing extension header combines the concepts of the strict source route and the loose source route options of IPv4.
- **Fragmentation:** The concept of fragmentation is the same as that in IPv4. However, the place where fragmentation occurs differs. In IPv4, the source or a router is required to fragment if the size of the datagram is larger than the MTU of the network over which the datagram travels. In IPv6, only the original source can fragment. A source must use a path MTU discovery technique to find the smallest MTU supported by any network on the path. The source then fragments using this knowledge.
- **Authentication:** The authentication extension header has a dual purpose: it validates the message sender and ensures the integrity of data.
- **Encrypted Security:** Payload The encrypted security payload (ESP) is an extension that provides confidentiality and guards against eavesdropping.
- **Destination Option:** The destination option is used when the source needs to pass information to the destination only. Intermediate routers are not permitted access to this information.

Advantages

The next-generation IP, or IPv6, has some advantages over IPv4 that can be summarized as follows:

1. Larger address space. An IPv6 address is 128 bits long, as we discussed in Chapter 19. Compared with the 32-bit address of IPv4, this is a huge (296) increase in the address space.
2. Better header format. IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
3. New options. IPv6 has new options to allow for additional functionalities.
4. Allowance for extension. IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.

5. Support for resource allocation. In IPv6, the type-of-service field has been removed, but a mechanism (called Flow Label) has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
6. Support for more security. The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

IP Address:

1. Computers communicate through the Internet. The packet transmitted by the sending computer may pass through several LANs or WANs before reaching the destination computer.
2. For this level of communication, we need a global addressing scheme; we called this logical addressing. Today, we use the term IP address to mean a logical address in the network layer of the TCP/IP protocol suite.
3. The Internet addresses are 32 bits in length; this gives us a maximum of 2^{32} addresses. These addresses are referred to as IPv4 (IP version 4) addresses or simply IP addresses if there is no confusion.
4. The Internet uses 128-bit addresses that give much greater flexibility in address allocation. These addresses are referred to as IPv6 (IP version 6) addresses.

Definition:

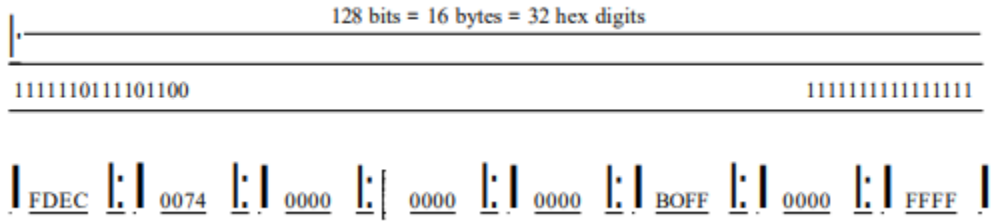
- It can be defined as the communication between computers through the internet for transmitting packets through several LANs or WANs is referred to as Logical Address ie IP Address.
- The Internet addresses are 32 bits in length; this gives us a maximum of 2^{32} addresses ie IPv 4 Address.
- The Internet uses 128-bit addresses that give much greater flexibility in address allocation ie IPv 6 Address .

IPv4 ADDRESSES:

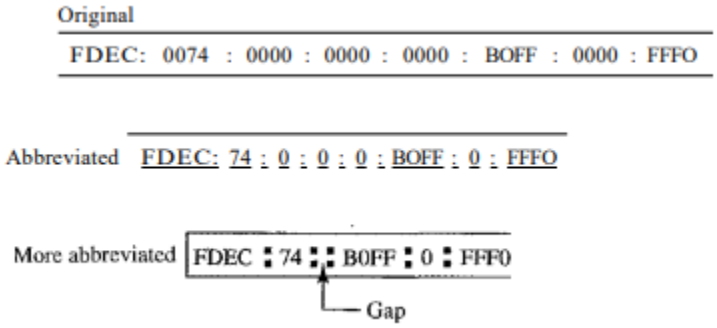
1. An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet.
2. An IPv4 address is **32 bits long.**
3. The IPv4 addresses are unique and universal.
4. The **address space** of IPv4 is 232 or 4,294,967,296.
5. There are two prevalent notations to show an IPv4 address: **binary notation(01110101 10010101 00011101 00000010) and dotted decimal notation(117.149.29.2).**
6. IPv4 ADDRESSES consist of **Classful Addressing, Classless Addressing & NAT.**

IPv6 ADDRESSES:

1. An IPv6 address is 128 bits long.
2. IPv6 address in binary and hexadecimal colon notation



3. Although the IP address, even in hexadecimal format, is very long, many of the digits are zer9s. In this case, we can **abbreviate the address.** The leading zeros of a section (four digits between two colons) can be omitted.



4. A **Unicast Address** defines a single computer. The packet sent to a unicast address must be delivered to that specific computer. IPv6 defines two types of unicast addresses: **Geographically based and Provider-based.**
5. **Multicast addresses** are used to define a group of hosts instead of just one. A packet sent to a multicast address must be delivered to each member of the group.
6. IPv6 also defines **Anycast Addresses**. An anycast address, like a multicast address, also defines a group of nodes. It also defines **Reserved Address and Logical Address.**

Classful Addressing:

1. IPv4 addressing, at its inception, used the concept of classes. This architecture is called classful addressing.
2. In classful addressing, the address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the address space.
3. We can find the class of an address when given the address in binary notation or dotted-decimal notation. If the address is given in binary notation, the first few bits can immediately tell us the class of the address.
4. If the address is given in decimal-dotted notation, the first byte defines the class. Both methods are shown in Figure.

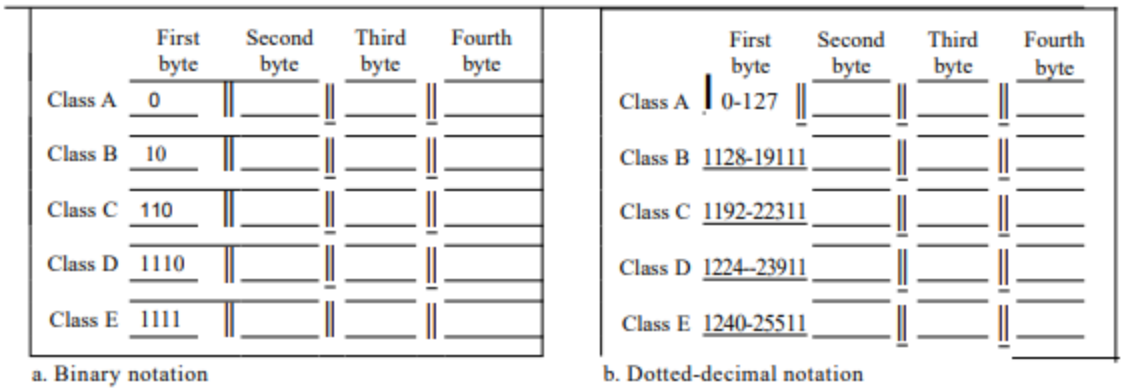


Figure: Finding the classes in binary and dotted-decimal notation

Example

Find the class of each address.

a. 00000001 00001011 00001011 11101111
b. 11000001 10000011 00011011 11111111
c. 14.23.120.8
d. 252.5.15.111

Solution

a. The first bit is 0. This is a class A address.
b. The first 2 bits are 1; the third bit is 0. This is a class C address.
c. The first byte is 14 (between 0 and 127); the class is A.
d. The first byte is 252 (between 240 and 255); the class is E.

Classes and Blocks

Class	*Number of Blocks*	*Block Size*	*Application*
A	128	16,777,216	Unicast
B	16,384	65,536	Unicast
C	2,097,152	256	Unicast
D	1	268,435,456	Multicast
E	1	268,435,456	Reserved

Figure: Number of blocks and block size in classfullPv4 addressing

Netid and Hostid

1. In classful addressing, an IP address in class A, B, or C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address.
2. Table some netid and hostid bytes. The netid is in color, the hostid is in white. Note that the concept does not apply to classes D and E.
3. **In class A, one byte defines the netid and three bytes define the hostid. In class B, two bytes define the netid and two bytes define the hostid. In class C, three bytes define the netid and one byte defines the hostid.**

Mask

1. Although the length of the netid and hostid (in bits) is predetermined in classful addressing, we can also use a mask (also called the default mask), a 32-bit number made of contiguous Is followed by contiguous as.
2. The masks for classes A, B, and C are shown in Table. The concept does not apply to classes D and E.

Class	*Binary*	*Dotted-Decimal*	*CIDR*
A	11111111 00000000 00000000 00000000	255.0.0.0	18
B	11111111 11111111 00000000 00000000	255.255.0.0	116
C	11111111 11111111 11111111 00000000	255.255.255.0	124

Figure: Concept of NetId , HostId & Net Mask

Subnetting

1. During the era of classful addressing, subnetting was introduced. If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called subnets) or, in rare cases, share part of the addresses with neighbors.
2. Subnetting increases the number of Is in the mask.

Supernetting

1. The time came when most of the class A and class B addresses were depleted; however, there was still a huge demand for midsize blocks. The size of a class C block with a maximum number of 256 addresses did not satisfy the needs of most organizations. Even a midsize organization needed more addresses. One solution was supernetting.
2. In supernetting, an organization can combine several class C blocks to create a larger range of addresses. In other words, several networks are combined to create a supernetwork or a supemet. An organization can apply for a set of class C blocks instead of just one.
3. For example, an organization that needs 1000 addresses can be granted four contiguous class C blocks. The organization can then use these addresses to create one supernetwork. Supernetting decreases the number of Is in the mask.

Address Depletion

1. The flaws in classful addressing scheme combined with the fast growth of the Internet led to the near depletion of the available addresses.
2. Yet the number of devices on the Internet is much less than the 2^{32} address space. We have run out of class A and B addresses, and a class C block is too small for most midsize organizations.
3. One solution that has alleviated the problem is the idea of classless addressing. Classful addressing, which is almost obsolete, is replaced with classless addressing.

Classless Addressing:

To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented. In this scheme, there are no classes, but the addresses are still granted in blocks.

Address Blocks

1. In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block (the number of addresses) varies based on the nature and size of the entity.
2. For example, a household may be given only two addresses; a large organization may be given thousands of addresses.
3. An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.
4. **Restriction To simplify the handling of addresses, the Internet authorities impose three restrictions on classless address blocks:**
    - The addresses in a block must be contiguous, one after another.
    - The number of addresses in a block must be a power of 2 (I, 2, 4, 8, ...).
    - The first address must be evenly divisible by the number of addresses.

Mask

1. In 1Pv4 addressing, a block of addresses can be defined as x.y.z.tln in which x.y.z.t defines one of the addresses and the In defines the mask.

First Address

The first address in the block can be found by setting the 32 - n rightmost bits in the binary notation of the address to Os.

Last Address

The last address in the block can be found by setting the 32 - n rightmost bits in the binary notation of the address to Is.

Number of Addresses

The number of addresses in the block is the difference between the last and first address. It can easily be found using the formula 232- n.

Network Addresses

The first address is called the network address and defines the organization network. It defines the organization itself to the rest of the world.

Example:

1. A block of addresses is granted to a small organization. We know that one of the addresses is 205.16.37.39/28. Find First address the last address for the block & Number of Address.

Solution:

Given address is 205.16.37.39/28.

- The binary representation of the given address is 11001101 00010000 00100101 00100 I 11.

We know that the first address in the block can be found by setting the rightmost 32 - n bits to Os.

- If we set 32 - 28 rightmost bits = 4 bits to 0,
- we get 11001101 000100000100101 0010000
- Convert it to Decimal, 205.16.37.32.

We know that the last address in the block can be found by setting the rightmost 32 - n bits to Is.

- If we set 32 - 28 rightmost bits = 4 bit to 1
- we get, 11001101 00010000 001001010010 1111
- Convert it to Decimal, 205.16.37.47.

We know that the number of addresses in the block can be found by using the formula 2^{32-n} •

- The value of n is 28
- Number of Network Address=2^32-28

=2^4

= 16

2. A block of addresses is granted to a small organization. We know that one of the addresses is 207.16.47.29/28. Find First address the last address for the block & Number of Address.(Self Practice).

Network Address Translation (NAT) Protocol

1. Many are not happy with one address; many have created small networks with several hosts and need an IP address for each host. With the shortage of addresses, this is a serious problem.
2. A quick solution to this problem is called network address translation (NAT). NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally. The traffic inside can use the large set; the traffic outside, the small set.
3. To separate the addresses used inside the home or business and the ones used for the Internet, the Internet authorities have reserved three sets of addresses as private addresses, shown in Table.

Range			Total
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

Figure: Addresses For private networks

A NAT implementation

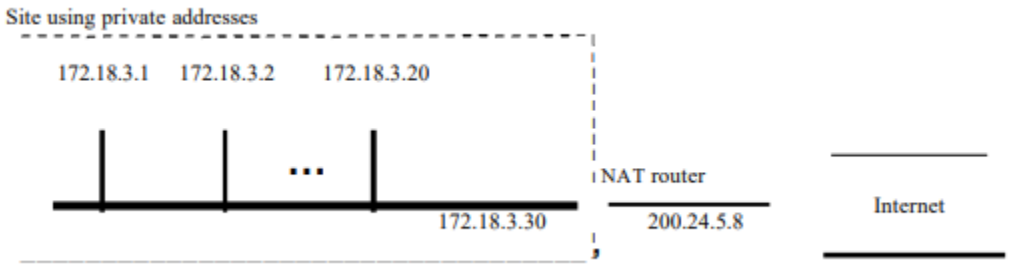


Figure: A NAT implementation

1. Figure shows a simple implementation of NAT. As Figure shows, the private network uses private addresses.
2. The router that connects the network to the global address uses one private address and one global address.
3. The private network is transparent to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8.

Address Translation

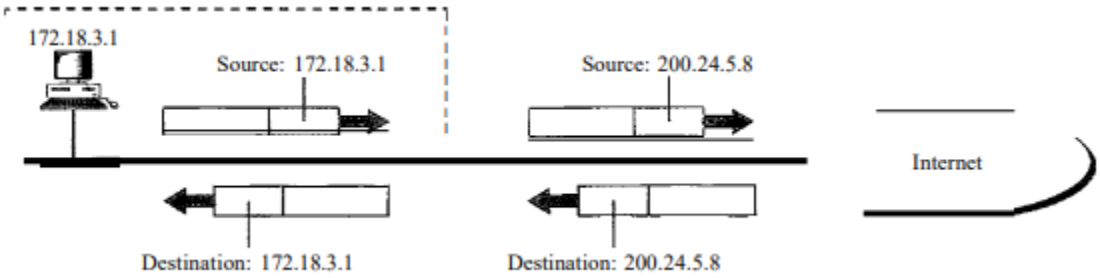


Figure: Addresses in a NAT

1. All the outgoing packets go through the NAT router, which replaces the source address in the packet with the global NAT address.
2. All incoming packets also pass through the NAT router, which replaces the destination address in the packet (the NAT router global address) with the appropriate private address.
3. Figure shows an example of address translation.

NAT and ISP

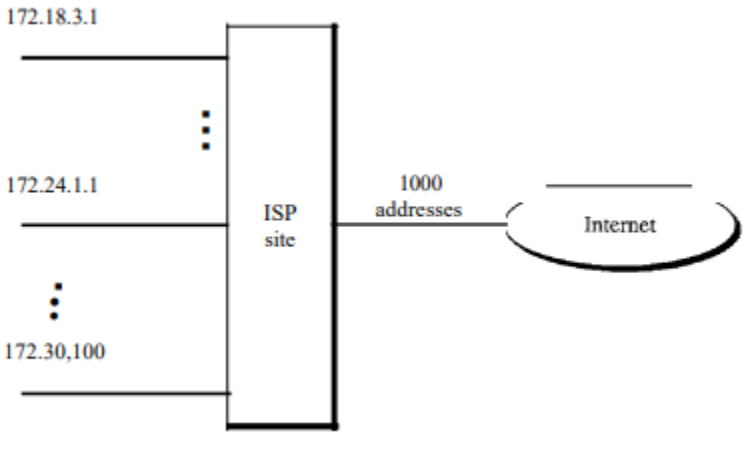


Figure: An ISP and NAT

1. An ISP that serves dial-up customers can use NAT technology to conserve addresses. For example, suppose an ISP is granted 1000 addresses, but has 100,000 customers.
2. Each of the customers is assigned a private network address. The ISP translates each of the 100,000 source addresses in outgoing packets to one of the 1000 global addresses; it translates the global destination address in incoming packets to the corresponding private address.

DHCP Protocol

The Dynamic Host Configuration Protocol (DHCP) has been devised to provide static and dynamic address allocation that can be manual or automatic.

Static Address Allocation

1. In this capacity DHCP acts as BOOTP does. It is backwardcompatible with BOOTP, which means a host running the BOOTP client can request a static address from a DHCP server.
2. A DHCP server has a database that statically binds physical addresses to IP addresses.

Dynamic Address Allocation

1. DHCP has a second database with a pool of available IP addresses. This second database makes DHCP dynamic.
2. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time.
3. When a DHCP client sends a request to a DHCP server, the server first checks its static database. If an entry with the requested physical address exists in the static database, the permanent IP address of the client is returned.
4. On the other hand, if the entry does not exist in the static database, the server selects an IP address from the available pool, assigns the address to the client, and adds the entry to the dynamic database.

5. The dynamic aspect of DHCP is needed when a host moves from network to network or is connected and disconnected from a network (as is a subscriber to a service provider).
6. DHCP provides temporary IP addresses for a limited time. The addresses assigned from the pool are temporary addresses.
7. The DHCP server issues a lease for a specific time. When the lease expires, the client must either stop using the IP address or renew the lease. The server has the option to agree or disagree with the renewal. If the server disagrees, the client stops using the address.

Manual and Automatic Configuration

1. One major problem with the BOOTP protocol is that the table mapping the IP addresses to physical addresses needs to be manually configured. This means that every time there is a change in a physical or IP address, the administrator needs to manually enter the changes.
2. DHCP, on the other hand, allows both manual and automatic configurations. Static addresses are created manually~ dynamic addresses are created automatically.

ICMP Protocol

1. The IP protocol is a best-effort delivery service that delivers a datagram from its original source to its final destination. However, **it has two deficiencies: lack of error control and lack of assistance mechanisms.**
2. The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol.

Types of Messages

1. ICMP messages are divided into two broad categories: error-reporting messages and query messages.
2. The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.
3. The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host. For example, nodes can discover their neighbors. Also, hosts can discover and learn about routers on their network, and routers can help a node redirect its messages.

Message Format

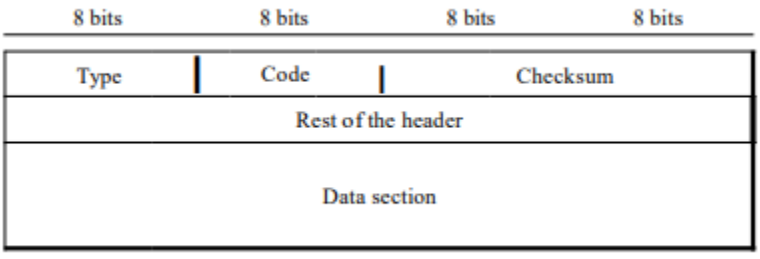


Figure: General Format of ICMP messages

1. An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all. As
2. The Figure shows the first field, ICMP type, defines the type of the message.
3. The code field specifies the reason for the particular message type.
4. The rest of the header is specific for each message type.The data section in error messages carries information for finding the original packet that had the error.
5. In query messages, the data section carries extra information based on the type of the query.

Error Reporting

1. ICMP always reports error messages to the original source.
2. Five types of errors are handled: destination unreachable, source quench, time exceeded, parameter problems, and redirection.

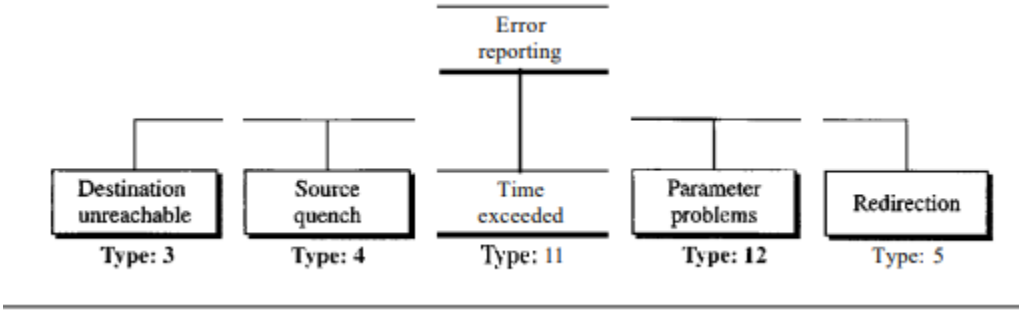
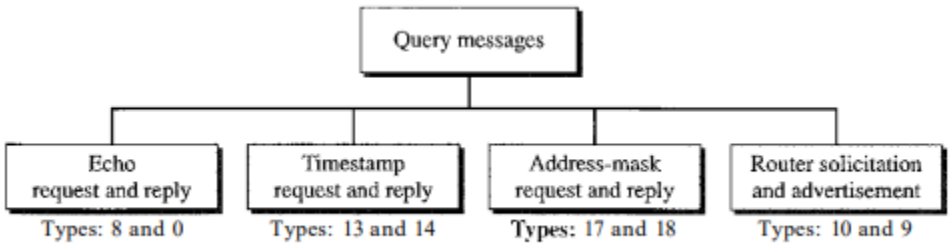


Figure: Error-reporting messages

Query

1. A query message is encapsulated in an IP packet, which in tum is encapsulated in a data link layer frame.
2. Figure show types of query Message,



5.3 Transport Layer : Connection Oriented and Connection less service,TCP and UDP protocol.

Compare Connection oriented and connectionless services

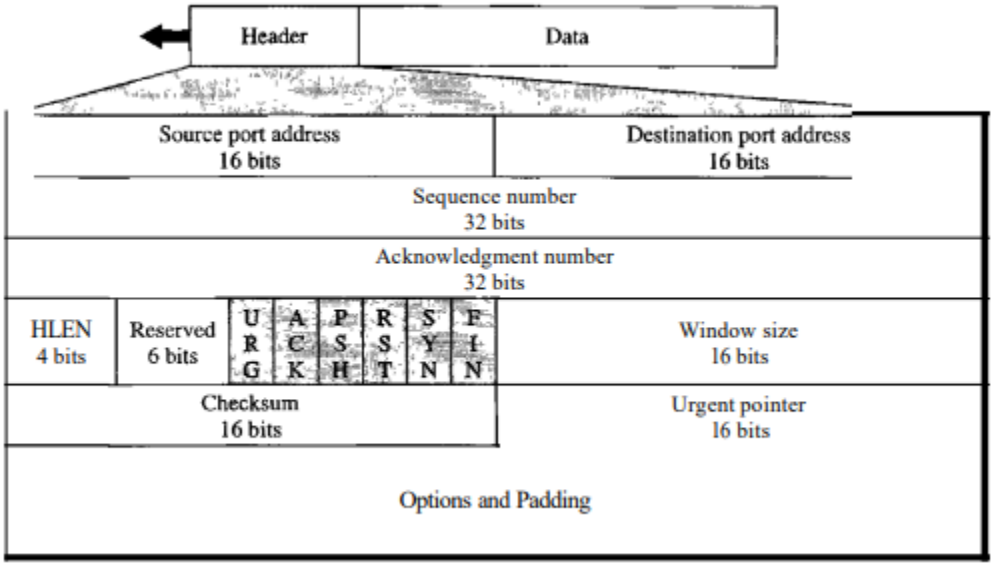
Sr No.	Protocol/Criteria	Connection-Oriented	Connectionless
1	Resource Allocation:	Resources need to be allocated.	No prior allocation of resource is required.
2	Utilization:	It often leads to low or underutilization of resources.	It ensures optimal usage of resources.
3	State info:	Lot of state related information needs to be stored.	Not much of information is required as packets are sent randomly.
4	Reliability:	It ensures reliable transfer of data.	Reliability is not guaranteed as it is a best effort service.
5	Connection:	Prior connection needs to be established.	No prior connection is established.
6	Delay:	There is more delay in transfer of information, but once conn. Established faster delivery.	However there is no delay due absence of connection establishment phase.
7	Packet travel:	In this packets travel to their destination node in a sequential manner.	In this packets reach the destination in a random manner.

8	**Congestion:**	**Congestion is not at all possible.**	**It is likely that congestion occurs.**

TCP Protocol

1. The second transport layer protocol we discuss in this chapter is called Transmission Control Protocol (TCP). TCP, like UDP, is a process-to-process (program-to-program) protocol.
2. TCP, therefore, like UDP, uses port numbers. Unlike UDP, TCP is a connection oriented protocol; it creates a virtual connection between two TCPs to send data.
3. In addition, TCP uses flow and error control mechanisms at the transport level. In brief, TCP is called a connection-oriented, reliable transport protocol.
4. It adds connection-oriented and reliability features to the services of IP.

TCP format:



TCP Services Before we discuss TCP in detail, let us explain the services offered by TCP to the processes at the application layer.

Process-to-Process Communication Like UDP, TCP provides process-to-process communication using port numbers. Table lists some well-known port numbers used by TCP.

Table 23.2 *Well-known ports used by TCP*

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
20	FIP, Data	File Transfer Protocol (data connection)
21	FIP, Control	File Transfer Protocol (control connection)
23	TELNET	Tenninal Network
25	SMTP	Simple Mail Transfer Protocol
53	DNS	Domain Name Server
67	BOOTP	Bootstrap Protocol
79	Finger	Finger
80	HTTP	Hypertext Transfer Protocol
111	RPC	Remote Procedure Call

Stream Delivery Service

1. TCP, unlike UDP, is a stream-oriented protocol. In UDP, a process (an application program) sends messages, with predefined boundaries, to UDP for delivery. UDP adds its own header to each of these messages and delivers them to IP for transmission.
2. Each message from the process is called a user datagram and becomes, eventually, one IP datagram. Neither IP nor UDP recognizes any relationship between the datagrams.
3. TCP, on the other hand, allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.
4. TCP creates an environment in which the two processes seem to be connected by an imaginary "tube" that carries their data across the Internet.

Full-Duplex Communication TCP offers full-duplex service, in which data can flow in both directions at the same time. Each TCP then has a sending and receiving buffer, and segments move in both directions.

Connection-Oriented Service TCP, unlike UDP, is a connection-oriented protocol. When a process at site A wants to send and receive data from another process at site B, the following occurs:

1. The two TCPs establish a connection between them.
2. Data are exchanged in both directions.
3. The connection is terminated.

Reliable Service TCP is a reliable transport protocol. It uses an acknowledgment mechanism to check the safe and sound arrival of data. We will discuss this feature further in the section on error control.

TCP Features

1. **Numbering System:** Although the TCP software keeps track of the segments being transmitted or received, there is no field for a segment number value in the segment header. Instead, there are two fields called the sequence number and the acknowledgment number. These two fields refer to the byte number and not the segment number.
2. **Sequence Number:** After the bytes have been numbered, TCP assigns a sequence number to each segment that is being sent. The sequence number for each segment is the number of the first byte carried in that segment.
3. **Flow Control:** TCP, unlike UDP, provides flow control. The receiver of the data controls the amount of data that are to be sent by the sender. This is done to prevent the receiver from being overwhelmed with data. The numbering system allows TCP to use a byte-oriented flow control.
4. **Error Control:** To provide reliable service, TCP implements an error control mechanism. Although error control considers a segment as the unit of data for error detection (loss or corrupted segments), error control is byte-oriented, as we will see later.
5. **Congestion Control:** TCP, unlike UDP, takes into account congestion in the network. The amount of data sent by a sender is not only controlled by the receiver (flow control), but is also detennined by the level of congestion in the network.
6. **Segment Before:** A packet in TCP is called a segment.

USER DATAGRAM PROTOCOL (UDP)

1. The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. It does not add anything to the services of IP except to provide process-to process communication instead of host-to-host communication. Also, it performs very limited error checking.
2. If UDP is so powerless, why would a process want to use it? With the disadvantages come some advantages.
3. UDP is a very simple protocol using a minimum of overhead. If a process wants to send a small message and does not care much about reliability, it can use UDP. Sending a small message by using UDP takes much less interaction between the sender and receiver than using TCP or SCTP.

Well-Known Ports for UDP

Port	*Protocol*	*Description*
7	Echo	Echoes a received datagram back to the sender
9	Discard	Discards any datagram that is received
11	Users	Active users

Port	*Protocol*	*Description*
13	Daytime	Returns the date and the time
17	Quote	Returns a quote of the day
19	Chargen	Returns a string of characters
53	Nameserver	Domain Name Service
67	BOOTPs	Server port to download bootstrap information
68	BOOTPc	Client port to download bootstrap information
69	TFTP	Trivial File Transfer Protocol
III	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

User Datagram Format

UDP packets, called user datagrams, have a fixed-size header of 8 bytes. Figure shows the format of a user datagram -

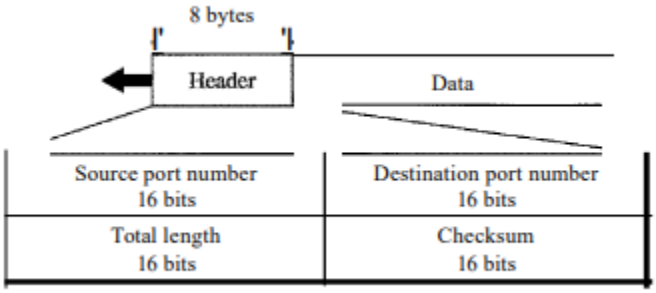


Figure: User datagram format

1. **Source port number.:**This is the port number used by the process running on the source host. It is 16 bits long, which means that the port number can range from 0 to 65,535. If The source host is the client (a client sending a request), the port number, in most cases, is an ephemeral port number requested by the process and chosen by the UDP software running on the source host. If the source host is the server (a server sending a response), the port number, in most cases, is a well-known port number.
2. **Destination port number:** This is the port number used by the process running on the destination host. It is also 16 bits long. If the destination host is the server (a client sending a request), the port number, in most cases, is a well-known port number. If the destination host is the client (a server sending a response), the port number, in most cases, is an ephemeral port number. In this case, the server copies the ephemeral port number it has received in the request packet.
3. **Length:** This is a 16-bit field that defines the total length of the user datagram, header plus data. The 16 bits can define a total length of 0 to 65,535 bytes. However, the total length needs to be much less because a UDP user datagram is stored in an IP datagram with a total length of 65,535 bytes.

4. **Checksum:** This field is used to detect errors over the entire user datagram (header plus data). The UDP checksum calculation is different from the one for IP and ICMP. Here the checksum includes three sections: a pseudo header, the UDP header, and the data coming from the application layer.

UDP Operation

UDP uses concepts common to the transport layer. These concepts will be discussed here briefly, and then expanded in the next section on the TCP protocol.

1. **Connectionless Services:** As mentioned previously, UDP provides a connectionless service. This means that each user datagram sent by UDP is an independent datagram. There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program. The user datagrams are not numbered. Also, there is no connection establishment and no connection termination, as is the case for TCP. This means that each user datagram can travel on a different path. One of the ramifications of being connectionless is that the process that uses UDP cannot send a stream of data to UDP and expect UDP to chop them into different related user datagrams. Instead each request must be small enough to fit into one user datagram. Only those processes sending short messages should use UDP.
2. **Flow and Error Control:** UDP is a very simple, unreliable transport protocol. There is no flow control and hence no window mechanism. The receiver may overflow with incoming messages. There is no error control mechanism in UDP except for the checksum. This means that the sender does not know if a message has been lost or duplicated. When the receiver detects an error through the checksum, the user datagram is silently discarded. The lack of flow control and error control means that the process using UDP should provide these mechanisms.
3. **Encapsulation and Decapsulation:** To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages in an IP datagram.

Use ofUDP The following lists some uses of the UDP protocol:

1. UDP is suitable for a process that requires simple request-response communication with little concern for flow and error control. It is not usually used for a process such as FrP that needs to send bulk data.
2. UDP is suitable for a process with internal flow and error control mechanisms. For example, the Trivial File Transfer Protocol (TFTP) process includes flow and error control. It can easily use UDP.
3. UDP is a suitable transport protocol for multicasting. Multicasting capability is embedded in the UDP software but not in the TCP software.
4. UDP is used for management processes such as SNMP.
5. UDP is used for some route updating protocols such as Routing Information Protocol (RIP).

5.4 Application Layer Protocols : HTTP, HTTPS,SMTP , SNMP, TELNET, DNS and FTP protocol.

For 5.4, https://www.gatevidyalay.com/computer-networks/