UNIT III

IPv4 :

Address Space, Notations, Classful, Classless, Network Address Translation, Datagram, Fragmentation and Checksum IPv6 Addresses: Structure, Address Space, Packet Format and Extension Headers, ICMP, IGMP, ARP, RARP, Congestion Control and Resource Allocation: Problem, Issues, Queuing, TCP

3.1 IPv4 Address Space

Q1. What is address space? Explain about it.

(Imp.)

Ans :

Address space is the amount of memory allocated for all possible addresses for a computational entity for example, a device, a file, a server or a networked computer. The system provides each device and process address space that holds a specific portion of the processor's address space. This can include either physical or virtual addresses accessible to a processor or reserved for a particular process.

The width of its address bus and registers often restricts the processor's address space. However, a memory management technique called virtual memory can increase the size of the address space to be higher than that of the physical memory.

Address space is classified as either flat or segmented. Flat address spaces are represented by incrementally increasing integers starting at zero. Independent segments augmented by offsets or values added to create secondary addresses represent segmented addresses.

In some systems, address space can be converted from one format to another through a thinking process — low-level, machine-generated code used to deploy details of a software system. Thanking is often used to delay calculations until the system requires a result.

Some types of address spaces

Here are a few examples of address spaces.

1. Virtual address space

A binary number in the virtual memory that allows processes to use a location in primary storage is a virtual address. This accommodates use of the main memory, independent of other processes, and supports the use of more space than what actually exists. It works by relegating some content to a hard disk or internal flash drive.

2. Logical address space

A logical address space is a set of logical addresses a computer generates for a specific program. A group of physical addresses mapped to corresponding logical addresses is called a physical address space.

3. IPv4 to IPv6

In terms of IP address space, concern emerged that the 32-bit address space of IP version 4 (IPv4) would be inadequate to support the enormous growth of the internet. So, IPv6 was developed with its 128-bit address space.

4. Subnetting IPv6 address space

The primary purpose of subnetting IPv6 address space is to improve address allocation efficiency by subnetting a segment of a network address space. Splitting an extensive network into smaller groups of interconnected networks reduces traffic, which helps increase network speeds because traffic does not have to flow through unnecessary routes. The subnet mask shares the network portion of the IP address and the host address range with the computer. The host address range comprises addresses assigned to host computers on the network.

Representation of 8 Bit Octet

128	64	32	16	8	4	2	1

The above representation shows the structure of 8- bit octet.

Now, we will see how to obtain the binary representation of the above IP address, i.e., 66.94.29.13

Step 1: First, we find the binary number of 66

128	64	32	16	8	4	2	1
0	1	0	0	0	0	1	0

To obtain 66, we put 1 under 64 and 2 as the sum of 64 and 2 is equal to 66 (64+2=66), and the remaining bits will be zero, as shown above. Therefore, the binary bit version of 66 is 01000010.

Step 2: Now, we calculate the binary number of 94

128	64	32	16	8	4	2	1
0	1	0	1	1	1	1	0

To obtain 94, we put 1 under 64, 16, 8, 4, and 2 as the sum of these numbers is equal to 94, and the remaining bits will be zero. Therefore, the binary bit version of 94 is 01011110.

Step 3: The next number is 29

128	64	32	16	8	4	2	1
0	0	0	1	1	1	0	0

To obtain 29, we put 1 under 16, 8, 4, and 1 as the sum of these numbers is equal to 29, and the remaining bits will be zero. Therefore, the binary bit version of 29 is 00011101.

Step 4: The last number is 13

128	64	32	16	8	4	2	1
0	0	0	0	1	1	0	1

To obtain 13, we put 1 under 8, 4, and 1 as the sum of these numbers is equal to 13, and the remaining bits will be zero. Therefore, the binary bit version of 13 is 00001101.

3.1.1 Notations

Q2. Explain about notations used in IPv4 and IPv6.

(Imp.)

Ans :

Dotted Decimal Notation

We have seen that the IPv4 address is expressed as a 32-bit number in dotted decimal notation. IP addresses may have a fixed part and variable part depending upon the allocation of total addresses to you or your organization.

The fixed part of the address may be from one octet to three octets, and the remaining octets will then be available for the variable part.

For example, you can take an IP address like 192.168.10.25. Now set all constant bits to 1 and set all variable bits to 0. This gives 11111111 11111111 00000000 On converting it in dotted-decimal notation, the outcome is 255.255.0.0.

This dotted-decimal notation with constant and variable methods can address prefixes to 192.168.10.25 and is represented as 192.168.10. 25, 255.255.0.0. This method of expressing the prefix length as a dotted-decimal number is known as a network mask or subnet mask notation.

Slash Notation

It is also known as CIDR (Classless Inter-Domain Routing) notation.

IPv4

Slash notation is a compact way to show or write an IPv4 subnet mask. When you use slash notation, you write the IP address, a forward slash (/), and the subnet mask number.

To find the subnet mask number:

1. Convert the decimal representation of the subnet mask to a binary representation.
2. Count each "1" in the subnet mask. The total is the subnet mask number.

For example, to write the IPv4 address 192.168.42. 23 with a subnet mask of 255.255.255.0 in slash notation:

1. **Convert the subnet mask to binary**

In this example, the binary representation of 255.255.255.0 is:

11111111.11111111.11111111.00000000.

2. **Count each 1 in the subnet mask**

In this example, there are twenty-four (24).

3. **Write the original IP address, a forward slash (/), and then the number from Step 2.**

The result is 192.168.42.23/24.

This table shows common network masks and their equivalents in slash notation.

Network Mask	Slash Equivalent
255.0.0.0	/8
255.255.0.0	/16
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
255.255.255.255	/32

IPv6

In IPv6, slash notation is used to represent the network identifier prefix for an IPv6 network. The prefix is expressed as a slash (/) followed by the prefix size, which is a decimal number between 1 and 128. The CIDR notation works exactly the same as with IPv4, which means if you have a /48, that means the first 48 bits of the address are the prefix.

This table shows common IPv6 network prefixes and the number of IPv6 subnets and IPv6 addresses they support.

Prefix	Number of Subnets
/64	1 IPv6 subnet with up to 18,446,744, 073,709,551,616 IPv6 host addresses
/56	256 /64 subnets
/48	65,536 /64 subnets

A network site that is assigned a /48 prefix can use prefixes in the range /49 to /64 to define valid subnets.

3.1.2 CLASS FULL

Q3. What is class full addressing? Explain

Ans :

Classfull Addressing

The 32 bit IP address is divided into five sub-classes. These are:

1. Class A
2. Class B
3. Class C
4. Class D
5. Class E

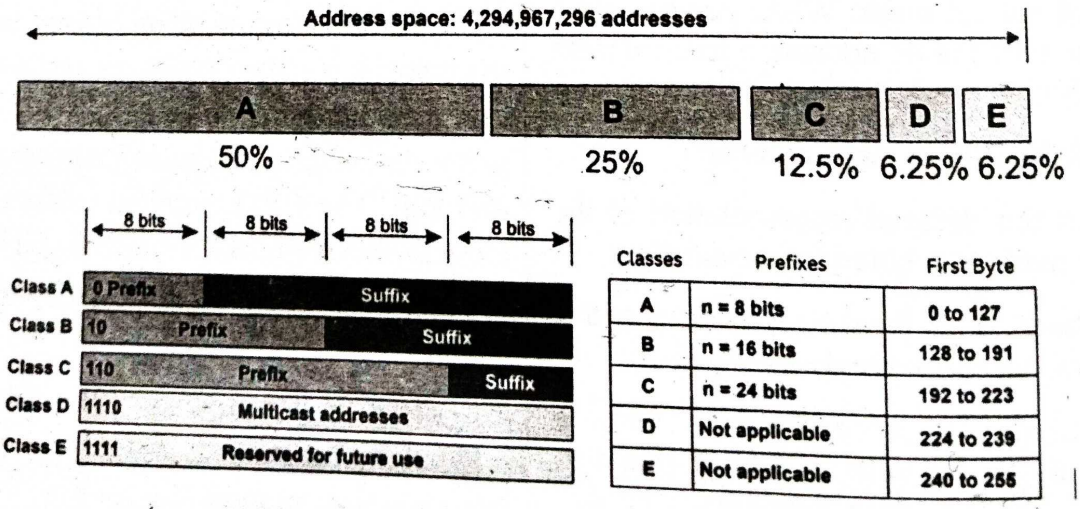
Each of these classes has a valid range of IP addresses. Classes D and E are reserved for multicast and experimental purposes respectively. The order of bits in the first octet determine the classes of IP address.

IPv4 address is divided into two parts:

- **Network ID**
- **Host ID**

The class of IP address is used to determine the bits used for network ID and host ID and the number of total networks and hosts possible in that particular class. Each ISP or network administrator assigns IP address to each device that is connected to its network.

Occupation of the address space in classful addressing



Classes	Prefixes	First Byte
A	n = 8 bits	0 to 127
B	n = 16 bits	128 to 191
C	n = 24 bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

CLASS A

1. Despite the fact that the network length is 8 bits, we can only use seven bits for the network identifier since the first bit, which is 0 and determines the class, is part of the length. This indicates that only $2^7 = 128$ networks can have a class A address globally.
   - Net ID = 8bits long and Host ID = 24 bits long
   - Method to identify class A addresses:
     - The first bit is reserved to 0 in binary
     - Range of the first octet is [0, 127] in dotted decimal
   - Total number of connections in Class A = 2^{31} (2, 14, 74, 83, 648)
   - There are $2^7 - 2 = 126$ networks in the Class A network.
     - There are 2 fewer networks available overall since IP Address 0.0.0.0 is set aside for broadcasting needs. For usage as a loopback address while testing software, the IP address 127.0.0.1 is set aside.
     - Hence, the range of the first octet becomes [1, 126]
   - Total number of Host IDs in Class A = 2^{24} - 2 [1, 67, 77, 214]
     - There are 2 fewer hosts that can be established across all classes due to the two reserved IP addresses, where all of the host ID bits are either zero or one.
     - The Network ID for the network is represented when all of the Host ID bits are set to 0.
   - The Broadcast Address is represented when all of the Host ID bits are set to 1.
   - Organizations needing very large networks, like Indian Railways, employ class A.

2. **CLASS B**

Despite the fact that the first two bits of class B's network, which are 10 in binary or we can write it as $(10)_2$, determine the class, we can only use 14 bits as the network identification, as class B's network length is 16 bits. As a result, only $2^{14} = 16{,}384$ networks in the entire world are capable of using a class B address.

- Length of Net Id = 16 bits and length of Host ID 16 bits.
- Method to identify Class B networks:
- First two bits are reserved to 10 in binary notation
- The Range of the first octet is [128, 191] in dotted decimal notation
- Total number of connections in the class B network is $2^{30} = 1, 07, 37, 41, 824$
- Total number of networks available in class B is $2^{14} = 16, 384$
- Total number of hosts that can be configured in Class B = $2^{16} - 2 = 165, 534$
- Organizations needing medium-sized networks typically utilize class B.

3. **CLASS C**

All addresses that begin with the number $(110)_2$ fall under class C. Class C networks are 24 bits long, but since the class is defined by three bits, the network identifier can only be 21 bits long. As a result, $2^{21} = 2, 097, 152$ networks worldwide are capable of using a class C address.

- The length of the Net Id and the Host Id = 24 bits and 16 bits respectively.
- Method to identify Class C networks:

- First three bits are reserved for 110 in binary notation or $(110)_2$.
- The range of the first octet is [192, 223] in dotted decimal notation.
- Total number of connections in Class C = 2^{29} = 53, 68, 70, 912.
- Total number of networks available in Class C = 2^{24} = 20, 97, 152.
- Total number of hosts that can be configured in every network in Class C = 2^8 - 2 = 254.
- Organizations needing small to medium-sized networks typically choose class C.

4. **CLASS D**

Prefix and suffix categories do not exist for Class D. It is employed for multicast addresses.

- There is no concept of Host ID and Net ID
- Method to identify Class D network:
- The first four bits are reserved to 1110 in binary notation or $(1110)_2$
- The range of the first octet is [224, 239] in dotted decimal notation
- Total number of IP addresses available is 2^{28} = 26, 84, 35, 456
- Because data is not intended for a specific host, Class D is set aside for multicasting, which eliminates the requirement to extract the host address from the IP address.

5. **CLASS E**

All binary addresses with the prefix 1111 fall under class E. Class E, like Class D, does not have a prefix or a suffix and is used as a reserve.

- Like in Class D, there is also no concept of Host ID and Net ID.
- Method to identify Class E networks:
- The first four bits are reserved to 1111 in binary notation or (1111)
- The range of the first octet is [240, 255] in dotted decimal notation.
- Total number of IP addresses available is 2^{28} = 26,84,35,456.
- Class E is set aside for hypothetical or experimental uses.

Rules for assigning Network ID

Hosts that are located on the same physical network are identified by the network ID, as all host on the same physical network is assigned the same network ID. The network ID is assigned based on the following rules:

- The network ID cannot start with 127 because 127 belongs to class A address and is reserved for internal loop-back functions.
- All bits of network ID set to 1 are reserved for use as an IP broadcast address and therefore, cannot be used.
- All bits of network ID set to 0 are used to denote a specific host on the local network and are not routed and therefore, aren't used.

3.1.3 CLASSLESS

Q4. Write about classless addressing.

Ans :

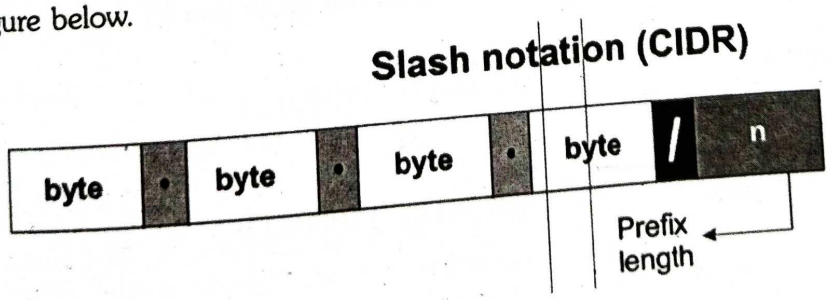
The address depletion issue was not fully resolved by classful addressing's subnetting and supernetting techniques. As the Internet expanded, it became obvious that a bigger address space was required as a long-term fix. However, the expanded address space necessitates that IP addresses should be longer as well, necessitating a change in IP packet syntax. The short-term solution, which uses the same address space but modifies the distribution of addresses to deliver a fair amount to each business, was developed despite the fact that the long-term solution, known as IPv6, has already been developed. Classless addressing is the temporary fix, which nevertheless makes use of IPv4 addresses. In order to make up for address depletion, the class privilege was taken out of the distribution.

The entire address space is partitioned into blocks of varying lengths with classless addressing. An address's prefix designates the block (network); its suffix designates the node (device). We are capable of having a block of 20, 21, 22 ,..., 232 addresses, theoretically. One of the limitations is that a block of addresses must have a power of two addresses. One address block may be given to an organization. The given figure demonstrates the non-overlapping block segmentation of the entire address space.

The address depletion issue was not fully resolved by classful addressing's subnetting and supernetting techniques. As the Internet expanded, it became obvious that a bigger address space was required as a long-term fix. However, the expanded address space necessitates that IP addresses should be longer as well, necessitating a change in IP packet syntax. The short-term solution, which uses the same address space but modifies the distribution of addresses to deliver a fair amount to each business, was developed despite the fact that the long-term solution, known as IPv6, has already been developed. Classless addressing is the temporary fix, which nevertheless makes use of IPv4 addresses. In order to make up for address depletion, the class privilege was taken out of the distribution.

The entire address space is partitioned into blocks of varying lengths with classless addressing. An address's prefix designates the block (network); its suffix designates the node (device). We are capable of having a block of 20, 21, 22 ,..., 232 addresses, theoretically. One of the limitations is that a block of addresses must have a power of two addresses. One address block may be given to an organization. The given figure demonstrates the non-overlapping block segmentation of the entire address space.

In classless addressing, the first issue that needs to be resolved is how to determine the prefix length if an address is provided. We must individually provide the prefix length because it is not a property of the address. The address is inserted in this scenario, followed by a slash, and the prefix length, n. Slash notation is the colloquial name for the notation, while classless interdomain routing, or CIDR (pronounced cider) method, is the official name. An address in classless addressing can thus be expressed as illustrated in the figure below.



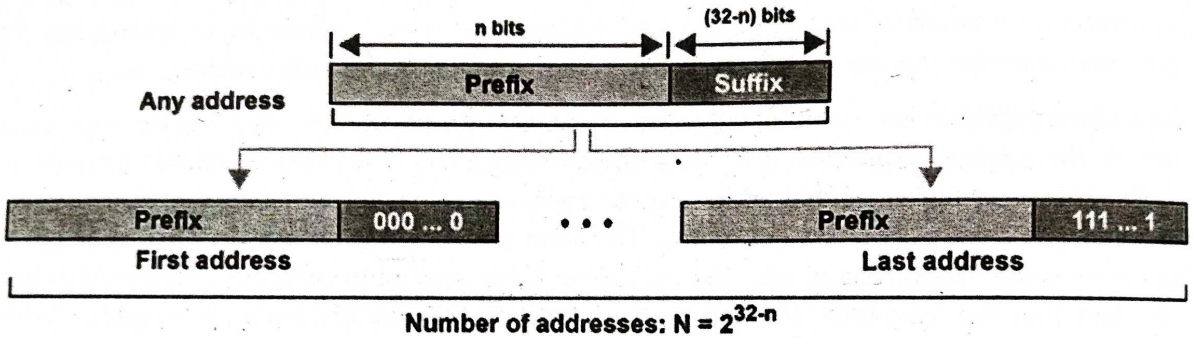
To put it another way, we must also provide the prefix length in classless addressing because an address does not automatically define the block or network to which it belongs.

Extracting Information from an Address

With respect to any given address in the block, we typically like to know three things: the number of addresses in the block, the start address in the block, and the last address. These three pieces of information, which are depicted in the picture below, are simple to locate because the prefix length, n, is known.

- The block has N = 232n addresses, according to the calculation.
- The n leftmost bits are kept, and the (32 - n) rightmost bits are all set to zeroes to determine the first address.
- The n leftmost bits are kept, while the (32 - n) rightmost bits are all set to 1s to determine the last address.

Information extraction in classless addressing



For Example

The address 167.199.170.82/27 is a classless address. The following is where we can find the aforementioned three pieces of data. In the network, there are $2^{32-n} = 2^5 = 32$ addresses in all.

The first 27 bits are kept while the remaining bits are converted to 0s to determine the first address.

Address: 167.199.170.82/27 10100111 11000111

10101010 01010010

Last address: 167.199.170.64/27 10100111 11000111

10101010 01000000

Keeping the first 27 bits and turning the remaining bits to 1s will allow you to determine the last address.

Address: 167.199.170.82/27 10100111 11000111

1010101001011111

Last address: 167.199.170.95/27 10100111 11000111

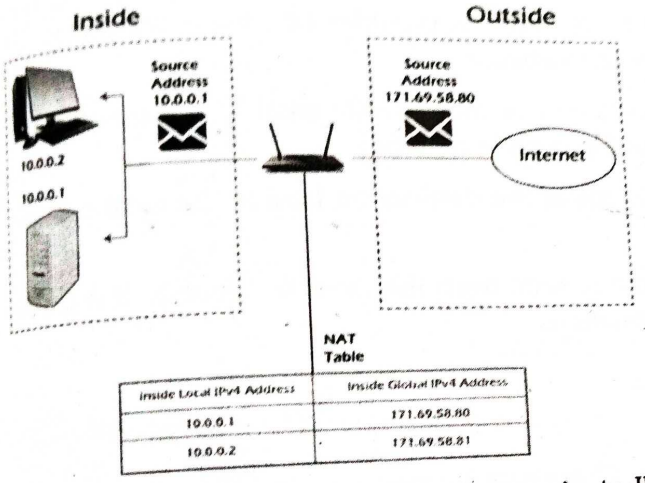
1010101001011111

3.1.4 Network Address Translation

Q5. What is Network Address Translation? Explain

Ans : **(Imp.)**

NAT (Network Address Translation) connects two networks and maps the private (inside local) addresses into public addresses (inside global). Inside local denotes that the best address belonged to an internal network and was not assigned by a Network Information Centre or service power. The inside global signifies that the address is a valid address assigned by the NIC or service provider, and one or more inside local addresses to the outside world.



NAT is a method of converting a private IP address or a local address into a public IP address. NAT is a technique for reducing the rate at which available IP addresses are depleted by translating a local IP or private IP address into a global or public IP address. The NAT relation might be one-to-one or many-to-one.

Furthermore, NAT can only configure one address in order to represent the entire network to the outside world. As a result, the translation process is transparent. NAT can be used to migrate and merge networks, share server loads, and create virtual servers, etc.

Types of NAT

There are three types of NAT:

1. Static NAT

In static NAT, a local address is mapped to a global address. In this type of NAT, the relationship is one-to-one. Static NAT is used if a host needs a consistent address that must be acceded from the Internet. For example, networking devices or enterprise servers.

2. Dynamic NAT

Unregistered private IP addresses can be converted to registered public IP numbers from a pool of public IP addresses using dynamic NAT.

3. PAT/NAT Overloading/IP masquerading

Among the three varieties, PAT is the most famous. It's a form of Dynamic NAT that's comparable to it, but it uses ports to translate many private IP addresses to a single public IP address.

4. Network Address Translation (NAT) working

Generally, the border router is configured for NAT i.e the router which has one interface in the local (inside) network and one interface in the global (outside) network. When a packet traverse outside the local (inside) network, then NAT converts that local (private) IP address to a global (public) IP address. When a packet enters the local network, the global (public) IP address is converted to a local (private) IP address.

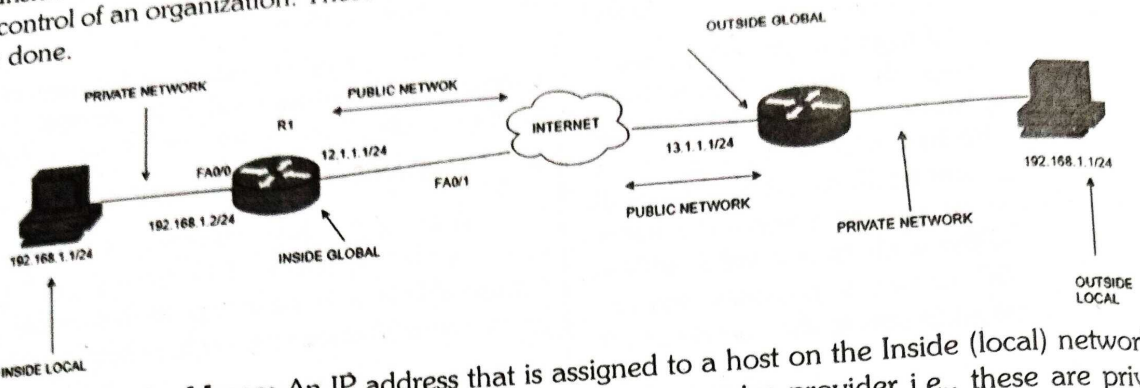
- If NAT runs out of addresses, i.e., no address is left in the pool configured then the packets will be dropped and an Internet Control Message Protocol (ICMP) host unreachable packet to the destination is sent.

Mask port numbers

Suppose, in a network, two hosts A and B are connected. Now, both of them request for the same destination, on the same port number, say 1000, on the host side, at the same time. If NAT does only translation of IP addresses, then when their packets will arrive at the NAT, both of their IP addresses would be masked by the public IP address of the network and sent to the destination. Destination will send replies to the public IP address of the router. Thus, on receiving a reply, it will be unclear to NAT as to which reply belongs to which host (because source port numbers for both A and B are the same). Hence, to avoid such a problem, NAT masks the source port number as well and makes an entry in the NAT table.

NAT inside and outside addresses

Inside refers to the addresses which must be translated. Outside refers to the addresses which are not in control of an organization. These are the network Addresses in which the translation of the addresses will be done.



- **Inside local address:** An IP address that is assigned to a host on the Inside (local) network. The address is probably not an IP address assigned by the service provider i.e., these are private IP addresses. This is the inside host seen from the inside network.
- **Inside global address:** IP address that represents one or more inside local IP addresses to the outside world. This is the inside host as seen from the outside network.
- **Outside local address:** This is the actual IP address of the destination host in the local network after translation.
- **Outside global address:** This is the outside host as seen from the outside network. It is the IP address of the outside destination host before translation.

Network Address Translation (NAT) Types

There are 3 ways to configure NAT:

1. **Static NAT:** In this, a single unregistered (Private) IP address is mapped with a legally registered (Public) IP address i.e one-to-one mapping between local and global addresses. This is generally used for Web hosting. These are not used in organizations as there are many devices that will need Internet access and to provide Internet access, a public IP address is needed.

   Suppose, if there are 3000 devices that need access to the Internet, the organization has to buy 3000 public addresses that will be very costly.

2. **Dynamic NAT:** In this type of NAT, an unregistered IP address is translated into a registered (Public) IP address from a pool of public IP addresses. If the IP address of the pool is not free, then the packet will be dropped as only a fixed number of private IP addresses can be translated to public addresses.

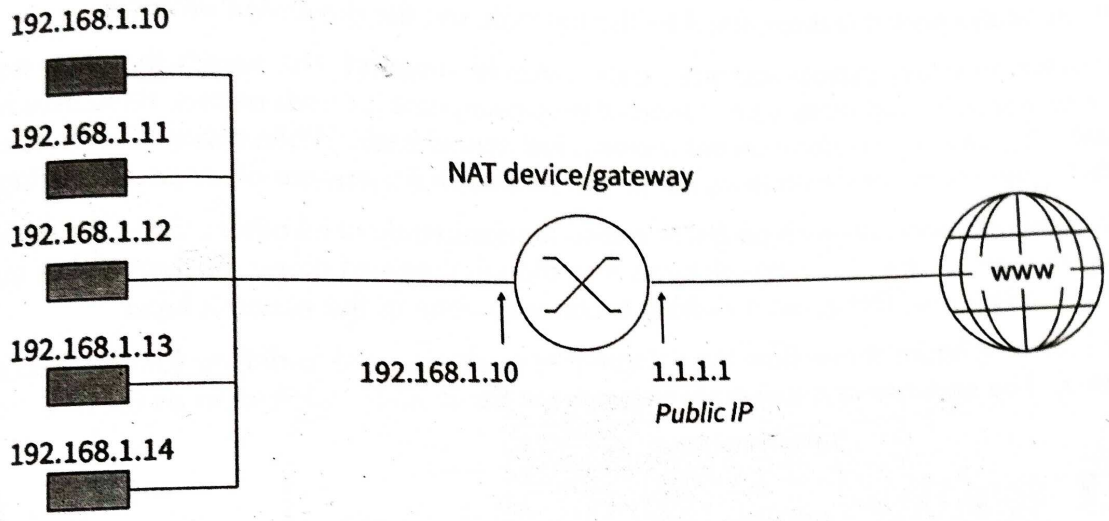
   Suppose, if there is a pool of 2 public IP addresses then only 2 private IP addresses can be translated at a given time. If 3rd private IP address wants to access the Internet then the packet will be dropped therefore many private IP addresses are mapped to a pool of public IP addresses. NAT is used when the number of users who want to access the Internet is fixed. This is also very costly as the organization has to buy many global IP addresses to make a pool.

3. **Port Address Translation (PAT):** This is also known as NAT overload. In this, many local (private) IP addresses can be translated to a single registered IP address. Port numbers are used to distinguish the traffic i.e., which traffic belongs to which IP address. This is most frequently used as it is cost-effective as thousands of users can be connected to the Internet by using only one real global (public) IP address.

NAT(Network Address Translation) Examples

When a host on the internal or private network with an internal IP address needs to communicate with a device outside of the private network, it will use the public IP address on the network's gateway to identify itself to the outside world, and NAT would translate the private IP address into the public address. If, for instance, a computer with the internal address 192.168.1.10 192.168.1.10 wished to communicate with a web server on the internet, NAT would translate that address to the company's public address, which we'll name in this case 1.1.1.11.1.1.1.

So that when communicating with the outside world, the internal address is recognized as the public address. This is necessary because, for the webserver to respond to this internal computer, it would need to transmit the response to the public address, which is a distinct and routable address on the internet. The private address is secret, non-routable, and concealed from the outside world, the original address of 192.168.1.10192.168.1.10 cannot be used. The public address for that company would be this one at 1.1.1.11.1.1.1, which is visible to everyone.



Network Address Translation

The web server would now respond to that 1.1.1.11.1.1.1 public address. NAT would use its records to convert the packets received from the web server intended for 1.1.1.11.1.1.1 back to the internal network address of 192.168.1.10192.168.1.10 so that the computer that requested the original information would get the requested packets.

The two advantages of NAT are now readily apparent. First, it would reduce the number of IP addresses we need because not every computer needs a public address. Second, it would shield these private computers from prying eyes. Only the public address is visible to everyone, everything else is concealed behind it. Therefore, nothing past the public address on the firewall's or router's external interface may be seen from the internet.

Advantages of NAT

The following are the advantages of NAT:

- NAT protects the public addresses that have been registered and slow down the IP address space exhaustion.
- Removes the address renumbering process that occurs when switching networks
- The occurrence of address overlap was significantly reduced.
- Increases flexibility of the connection establishment.

Disadvantages of NAT

The following are the disadvantages of NAT:

- Lack of end-to-end traceability
- Certain applications are not compatible with NAT
- Switching path delays are the outcome of the translation

3.1.5 DATAGRAM

Q6. Explain about datagram network **(Imp.)**

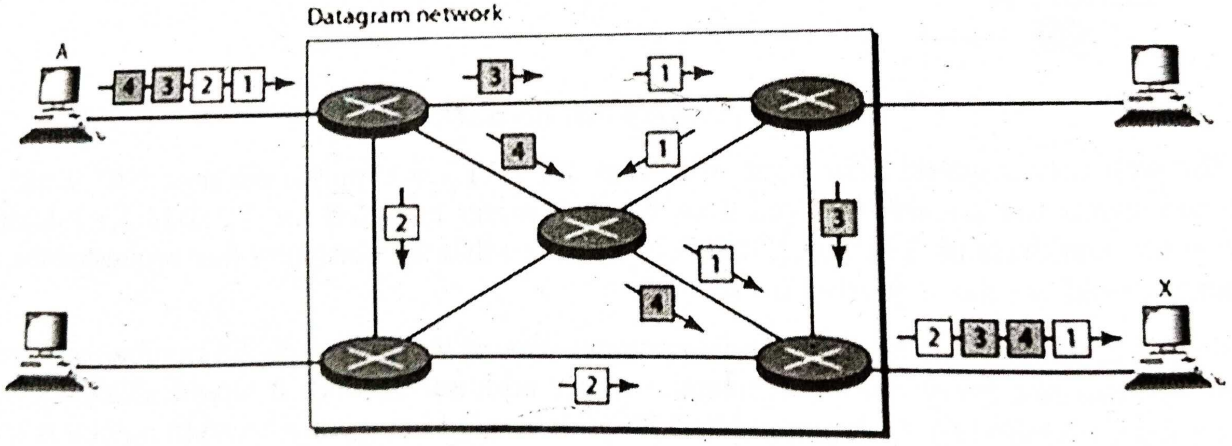
Ans :

In data communications, we need to send messages from one end system to another. If the message is going to pass through a packet-switched network, it needs to be divided into packets of fixed or variable size. The size of the packet is determined by the network and the governing protocol.

In packet switching, there is no resource allocation for a packet. This means that there is no reserved bandwidth on the links, and there is no scheduled processing time for each packet. Resources are allocated on demand. The allocation is done on a first come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed.

In a datagram network, each packet is treated independently of all others. Even if a packet is part of a multi packet transmission, the network treats it as though it existed alone. Packets in this approach are referred to as datagrams. Datagram switching is normally done at the network layer.

The following figure shows how the datagram approach is used to deliver four packets from station A to station X. The switches in a datagram network are traditionally referred to as routers.



In this example, all four packets (or datagrams) belong to the same message, but may travel different paths to reach their destination. This is so because the links may be involved in carrying packets from other sources and do not have the necessary bandwidth available to carry all the packets from A to X. This approach can cause the datagrams of a transmission to arrive at their destination out of order with different delays between the packets. Packets may also be lost or dropped because of a lack of resources. In most protocols, it is the responsibility of an upper-layer protocol to reorder the datagrams or ask for lost datagrams before passing them on to the application. The datagram networks are sometimes referred to as connectionless networks.

Routing Table

In this type of network, each switch (or packet switch) has a routing table which is based on the destination address. The routing tables are dynamic and are updated periodically. The destination addresses

and the corresponding forwarding output ports are recorded in the tables. This is different from the table of a circuit switched network in which each entry is created when the setup phase is completed and deleted when the teardown phase is over.

Destination Address

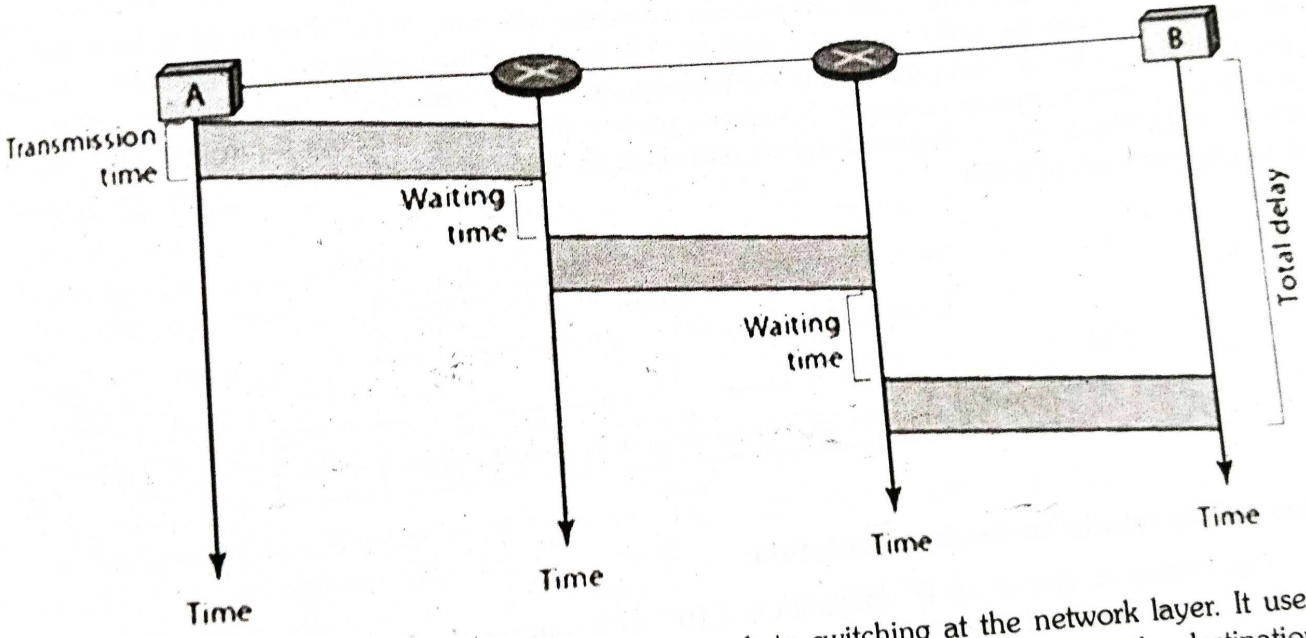
Every packet in a datagram network carries a header that contains, among other information, the destination address of the packet. When the switch receives the packet, this destination address is examined; the routing table is consulted to find the corresponding port through which the packet should be forwarded.

Efficiency

The efficiency of a datagram network is better than that of a circuit-switched network. Resources are allocated only when there are packets to be transferred. If a source sends a packet and there is a delay of a few minutes before another packet can be sent, the resources can be reallocated during these minutes for other packets from other sources.

Delay

There may be greater delay in a datagram network than in a virtual-circuit network. Although there are no setup and teardown phases, each packet may experience a wait at a switch before it is forwarded. In addition, since not all packets in a message necessarily travel through the same switches, the delay is not uniform for the packets of a message. The following figure gives an example of delay in a datagram network for one single packet.



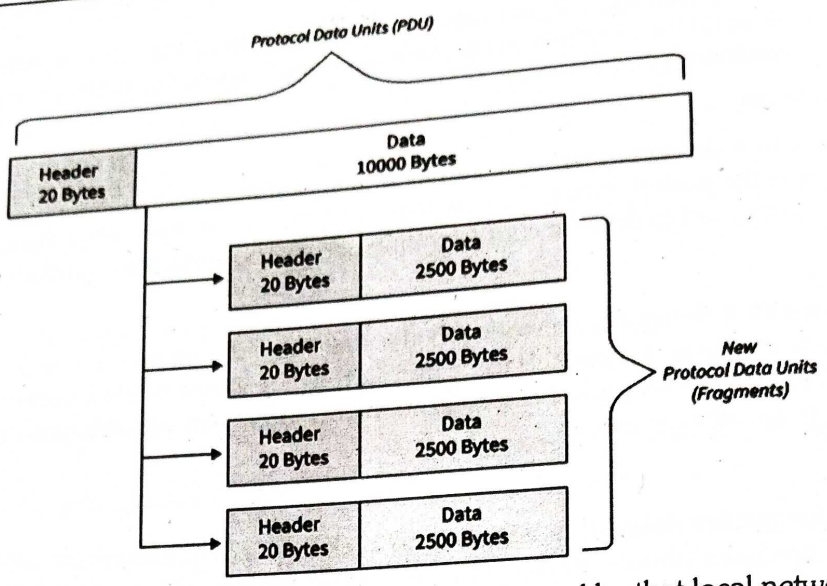
The Internet has chosen the datagram approach to switching at the network layer. It uses the universal addresses defined in the network layer to route packets from the source to the destination.

3.1.6 Fragmentation and Checksum

Q7. What is Fragmentation in Networking? Explain. **(Imp.)**

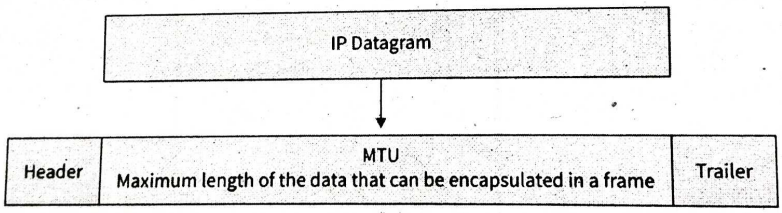
Ans :

IP fragmentation is a process that divides packets into smaller pieces (fragments) so that the resulting pieces can travel across a link with a smaller maximum transmission unit (MTU) than the original packet size. The receiving host reassembles the fragments.



An IP packet cannot be larger than the maximum size allowed by that local network when sent over the network by a host. The network's data link and IP Maximum Transmission Units (MTUs), which are typically the same, determine its size. 1500 byte MTUs are standard for modern Ethernet-based office, campus, or data center networks.

However, packets initially delivered across a network with one MTU may need to be routed over networks with a smaller MTU (such as a WAN or VPN tunnel). If the packet size in these circumstances is greater than the lower MTU, the data in the packet must be fragmented (if possible). This indicates that it is divided into fragments carried in brand-new packets (fragments) that are equal to or less than the lower MTU. This is known as fragmentation, and when the fragments arrive at their destination, the data is usually put back together.



Frame

Some points related to the fragmentation:

- The maximum size of an IP datagram is 216"1=65,535216"1=65,535 bytes, as the IP header has a total length of 16 bits.
- It is performed by the network layer at the destination side, typically at routers.
- Due to intelligent (*excellent*) segmentation by the transport layer, the source side does not require fragmentation. Specifically, the transport layer looks at the datagram and frame data limits and segments the data so that it can easily fit in a frame without the need for fragmentation.
- The receiver uses the identification (16 bits) field in the IP header to identify the packet. The identification number for each frame fragment is the same.
- The receiver uses the fragment offset(13 bits) field in the IP header to identify the series of frames.

The extra header created by fragmentation results in overhead at the network layer.

Process of Fragmentation

RFC 791 specifies IP packet fragmentation, transmission, and reassembly mechanism.

RFC 815 specifies a streamlined reassembly algorithm. The Identification field in the IP header, along with the foreign and local internet addresses and the protocol ID, and the Fragment offset field in the IP header, coupled with the Don't Fragment and More Fragments flags, are used for fragmentation and reassembly of IP packets.

If a receiving host receives a fragmented IP packet, it must put the packet back together and send it to the higher protocol layer. Reassembling is supposed to occur in the receiving host, but in reality, it might be carried out by an intermediate router. For instance, *network address translation (NAT)* can need to reassemble fragments to translate data streams.

Fields in IP Header for Fragmentation

Fields in IP header for fragmentation

- **Identification (16 bits):** use to identify fragments of the same frame.
- **Fragment offset (13 bits):** use to identify the sequence of fragments in the frame. It generally indicates a number of data bytes preceding or ahead of the fragment.

Maximum fragment offset possible = (65535 - 20) = 65515 {where 65535 is the maximum size of datagram and 20 is the minimum size of IP header}

So, we need ceil($\log_2 65515$) = 16 bits for a fragment offset but the fragment offset field has only 13 bits. So, to represent efficiently we need to scale down the fragment offset field by $2^{16}/2^{13} = 8$ which acts as a scaling factor. Hence, all fragments except the last fragment should have data in multiples of 8 so that fragment offset " N.

- **More fragments (MF = 1 bit): –** tells if more fragments are ahead of this fragment i.e. if MF = 1, more fragments are ahead of this fragment and if MF = 0, it is the last fragment.
- **Don't fragment (DF = 1 bit) –** if we don't want the packet to be fragmented then DF is set i.e. DF = 1.

IP Fragmentation Examples

Now let's understand the concept of IP fragmentation with the help of an example.

- In network X, a host named A has an MTU of 520 bytes.
- In network Y, a host named B has an MTU of 200 bytes.
- Host A of network X wants to send a message to host B in network Y.

Assume a router gets a datagram from host A that contains:

- The length of the header is 20 bytes.
- The length of the payload is 500 bytes.
- The whole length is 520 bytes.
- The *DF* bit is set to 0. The router now operates in the following steps:

Step 1

The router looks through the datagram and discovers:

- The datagram has a size of 520 bytes.
- Network Y is the destination, and its MTU is 200 bytes.
- The DF bit is set to 0.

The router concludes:

- The datagram's size exceeds the *MTU*.
- It must therefore break the datagram into fragments.
- *DF* bit has been set to 0.
- Therefore, it is acceptable to create datagram fragments.

Step 2

The router determines the amount of data that should be transmitted in each fragment.

The router is aware of:

- The destination network's MTU is 200 bytes.
- Therefore, any fragment can only have a maximum total length of 200 bytes.

- The header will take up 20 bytes out of the total 200 bytes.
- Thus, 180 bytes is the maximum amount of data that can be delivered in any fragment.

The router uses the following rule to determine how much data will be delivered in a single fragment:

Rule

The quantity of data delivered in a single fragment is chosen in such a way that-

1. It is as large as feasible but less than or equal to *MTU*.
2. It is a multiple of 8, so a pure decimal value for the fragment offset field can be obtained.
   - The final fragment is not required to contain data in multiples of 8, though.
   - This is because it need not determine the fragment offset value for any other fragment.

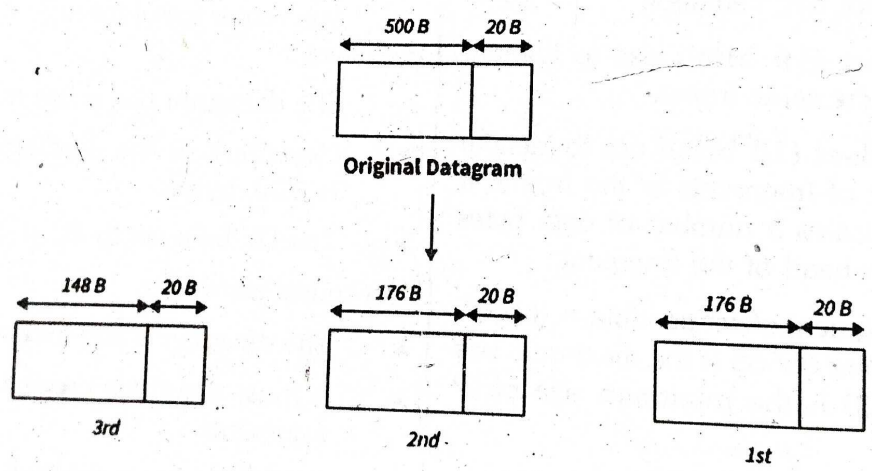
Following the above rule,

- The router determines a maximum of 176 bytes of data that can be sent in one fragment.
- This is because it is the highest figure that is less than *MTU* and a multiple of 8.

Step 3

The router splits the original datagram into three parts where:

- The first fragment contains data = 176 bytes.
- The second segment has data = 176 bytes.
- The third fragment contains data = 148 bytes.



Fragments of Original Datagram

Each fragment's IP header contains information:

Header information of 1st fragment

- Field value for header length = 20/4=520/4=5.
- Total length field value = 176+20=196176+20=196.
- MF bit = 1.
- The value of the fragment offset field is 0.
- The header checksum is updated.
- The identification number is the same as the original datagram.

Header information of 2nd fragment

- Field value for header length = 20/4=520/4=5.
- Total length field value = 176+20 =196176 + 20 = 196.
- MF bit = 1.
- The value of the fragment offset field is 176/8=22176/8=22.
- The header checksum is updated.
- The identification number is the same as the original datagram.

Header information of 3rd fragment

- Field value for header length = 20/4=520/4=5.
- Total length field value = 148+20 = 168148 + 20=168.
- MF bit = 0.
- The value of the fragment offset field is (176+176)/8=44(176+176)/8=44.
- The header checksum is updated.
- The identification number is the same as the original datagram.

The router retransmits all the fragments.

Step 4

On the destination side,

- The receiver receives three datagram fragments.
- To get the original datagram, the reassembly algorithm is used to combine all of the fragments.

Q8. Why is Fragmentation Needed?

Ans :

The datagram generated by the network layer at the source computer must traverse many networks before arriving at the destination computer. Typically, the source computer favors sending large datagrams. This is because if the datagram is broken up into smaller pieces, the header will be repeated for each datagram unit. The header is repeated for every fragmented datagram, wasting network bandwidth.

However, each network has a cap on the largest packet size it can send during this occurrence. Even worse, the source computer is unaware of the packet's route to get to its destination. It cannot, therefore, determine how small each fragmented datagram must be. The reasons for fragmenting a large datagram into a small fragmented datagram are listed below:

1. The capacity of data is limited by the hardware and operating system employed.
2. Conformity with national and international norms.
3. Each network's protocols allow for different packet sizes.
4. Large packets occupy the network for a longer time than small packets.
5. Reduce the mistake caused by retransmission.

Q9. What is Checksum? How to apply checksum for error detection? Explain.

(Imp.)

Ans :

The checksum is a network method to check for any error or damage to the data transmitted to the sender side from the sender side. The checksum method applies the bit addition and bit complement method to perform the checksum implementation.

The need to apply checksum or any other error-detection method is done simply to identify the damage to the data when it's being transmitted over the network channel.

The checksum uses the Checksum Generator on the sender side and the Checksum Checker on the receiver side.

Working Steps for Checksum

Steps involved in the checksum error-detection method:

Step 1: At the Sender Side

- Divide the original data into the 'm' number of blocks with 'n' data bits in each block.
- Adding all the 'k' data blocks.
- The addition result is complemented using 1's complement.
- The obtained data is known as the Checksum.

Step 2: Data Transmission

- Integrate the checksum value to the original data bit.
- Begin the transmission of data to the receiver side.

Step 3: At the Receiver Side

- Divide the received data into the 'k' number of blocks.
- Adding all the 'k' data blocks along with the checksum value.
- The addition result is complemented using 1's complement.
- Two possible cases after 1's complement:
- Case 1: If the result is 0.
- No errors in the received data from the sender side.
- The receiver accepts the data.
- Case 2: If the result is not 0.
- Errors in the received data from the sender side.
- The receiver discards the data and requests for retransmission of data.

Solved Example

Let's use an example to implement the checksum method and consolidate our understanding of the network principle.

For the given data value 11001100 10101010 11110000 11000011, perform the checksum method.

1. The first step is to perform the bit addition of the given data bits at the sender side.

Sender Side:

1	0	0	1	1	0	0	1
1	1	1	0	0	0	1	0
0	0	1	0	0	1	0	0
1	0	0	0	0	1	0	0
0	0	1	0	0	0	1	1
						1	0
0	0	1	0	0	1	0	1

Note: The extra carry bits are added to the summation result.

2. Perform the 1's Complement for the bit addition result, thus obtaining the checksum value.

Sender Side:

0 0 1 0 0 1 0 1

1's Complement

Checksum

0 0 1 0 0 1 0 1

3. Integrate the checksum value and the original data bit and begin the data transmission to the receiver.

11011010	10011001	11100010	00100100	10000100

4. The receiver side will begin the Checksum Checker method, repeat the bit addition, and perform the 1's complement.

Receiver Side:

```
  1  0  0  1  1  0  0  1
  1  1  1  0  0  0  1  0
  0  0  1  0  0  1  0  0
  1  0  0  0  0  1  0  0
  1  1  0  1  1  0  1  0
-------------------------
  1  1  1  1  1  1  0  1
                   1  0
-------------------------
  1  1  1  1  1  1  1  1
```

5. If the complement result is 0, the data received is correct and without any error.

Receiver Side:

1 1 1 1 1 1 1 1

1's Complement

Checksum 0 0 0 0 0 0 0 . 0

Result: No error in the data received from the sender side.

3.2 IPV6 ADDRESSES

3.2.1 Structure

Q10. Explain the structure of IPV6 **(Imp.)**

Ans :

Before introducing IPv6 Address format, we shall look into Hexadecimal Number System. Hexadecimal is a positional number system that uses radix (base) of 16. To represent the values in readable format, this system uses 0-9 symbols to represent values from zero to nine and A-F to represent values from ten to fifteen. Every digit in Hexadecimal can represent values from 0 to 15.

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Address Structure

An IPv6 address is made of 128 bits divided into eight 16-bits blocks. Each block is then converted into 4-digit Hexadecimal numbers separated by colon symbols.

For example, given below is a 128 bit IPv6 address represented in binary format and divided into eight 16-bits blocks:

0010000000000001 0000000000000000 0011001000111000 1101111111100001
0000000001100011 0000000000000000 0000000000000000 1111111011111011

Each block is then converted into Hexadecimal and separated by ':' symbol:

2001:0000:3238:DFE1:0063:0000:0000:FEFB

Even after converting into Hexadecimal format, IPv6 address remains long. IPv6 provides some rules to shorten the address. The rules are as follows:

Rule.1: Discard leading Zero(es)

In Block 5, 0063, the leading two 0s can be omitted, such as (5th block):

2001:0000:3238:DFE1:63:0000:0000:FEFB

Rule.2:

If two of more blocks contain consecutive zeroes, omit them all and replace with double colon sign ::, such as (6th and 7th block):

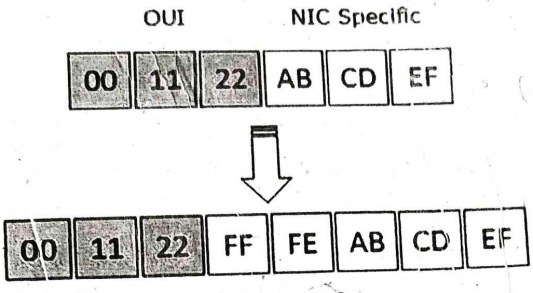
2001:0000:3238:DFE1:63::FEFB

Consecutive blocks of zeroes can be replaced only once by :: so if there are still blocks of zeroes in the address, they can be shrunk down to a single zero, such as (2nd block):

2001:0:3238:DFE1:63::FEFB

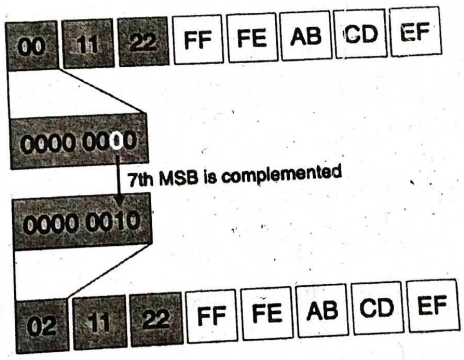
Interface ID

IPv6 has three different types of Unicast Address scheme. The second half of the address (last 64 bits) is always used for Interface ID. The MAC address of a system is composed of 48-bits and represented in Hexadecimal. MAC addresses are considered to be uniquely assigned worldwide. Interface ID takes advantage of this uniqueness of MAC addresses. A host can auto-configure its Interface ID by using IEEE's Extended Unique Identifier (EUI-64) format. First, a host divides its own MAC address into two 24-bits halves. Then 16-bit Hex value 0xFFFE is sandwiched into those two halves of MAC address, resulting in EUI-64 Interface ID.



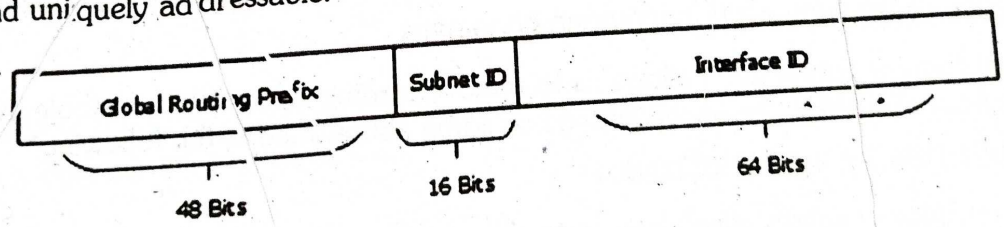
Conversion of EUI-64 ID into IPv6 Interface Identifier

To convert EUI-64 ID into IPv6 Interface Identifier, the most significant 7th bit of EUI-64 ID is complemented. For example:



Global Unicast Address

This address type is equivalent to IPv4's public address. Global Unicast addresses in IPv6 are globally identifiable and uniquely addressable.

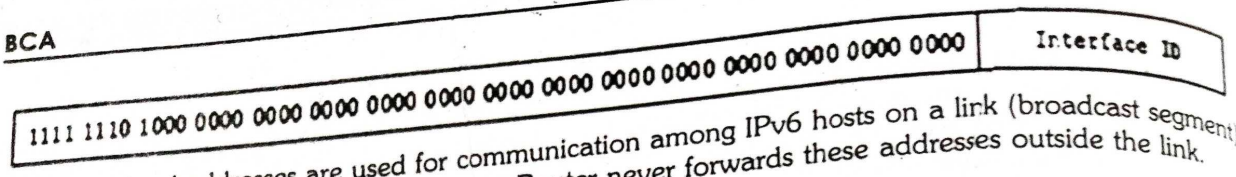


Global Routing Prefix

The most significant 48-bits are designated as Global Routing Prefix which is assigned to specific autonomous system. The three most significant bits of Global Routing Prefix is always set to 001.

Link-Local Address

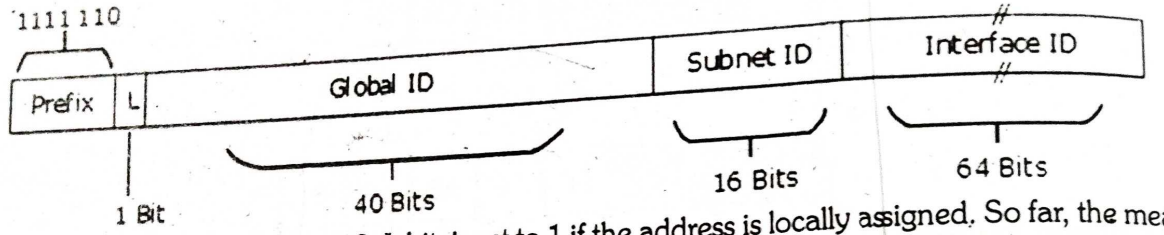
Auto-configured IPv6 address is known as Link-Local address. This address always starts with FE80. The first 16 bits of link-local address is always set to 1111 1110 1000 0000 (FE80). The next 48-bits are set to 0, thus:



Link-local addresses are used for communication among IPv6 hosts on a link (broadcast segment) only. These addresses are not routable, so a Router never forwards these addresses outside the link.

Unique-Local Address

This type of IPv6 address is globally unique, but it should be used in local communication. The second half of this address contain Interface ID and the first half is divided among Prefix, Local Bit, Global ID and Subnet ID.



Prefix is always set to 1111 110. L bit, is set to 1 if the address is locally assigned. So far, the meaning of L bit to 0 is not defined. Therefore, Unique Local IPv6 address always starts with 'FD'.

Scope of IPv6 Unicast Addresses

The scope of Link-local address is limited to the segment. Unique Local Address are locally global, but are not routed over the Internet, limiting their scope to an organizatior's boundary. Global Unicast addresses are globally unique and recognizable. They shall make the essence of Internet v2 addressing.

3.2.2 Address Space

Q11. Explain about address space of IPv6.

Ans :

To simplify the address representation, IPv6 supports two types of abbeviations. Both abbreviations work with zeros. The first abbreviation allows us to skip leading zeros within a nibble while the second abbreviation allows us to drop the nibbles that contain only zeros. Since most IPv6 addresses contain long sequences of zeros, these two abbreviations can make writing IPv6 addresses a lot easier.

To understand how abbreviations work, let's take an example. The following is the IPv6 address.

2001:0DB8:5002:AB41:0000:0000:0000:0801

The first form of abbreviation allows us to remove leading zeros within a nibble. After removing leading zeros from all nibbles, the above address could be abbreviated as the following.

2001:DB8:5002:AB41:0:0:0:801

The second form of abbreviation allows us to use a double colon torepresent one or more consecutive sets of zero nibbles. Using this form of abbreviation, the above address can be further abbreviated as the following.

2001:DB8:5002:AB41::801

Two important rules must be followed when abbreviating an IPv6address. First, you can't abbreviate a zero that is not leading in the nibble. For example, you can't abbreiate the address 2001:0DB8:5000: AB00:2300:0034:00A4:0801 as the 2001:0DB8:5:AB:23:34:A4:801 But, you can abbreviate this address as the 2001:DB8:5000:AB00:2300:34:A4:801.

Second, you can use only one double colon within an address representation. For example, you can abbreviate the address 2001:AB3:0:45CA:0:0:0:F5 as either 2001:AB3::45CA:0:0:0:F5 or 2001:AB3:0:45CA::F5. But you can't abbreviate this address as the 2001:AB3::45CA::F5. If you will use two double colons in an address, the address will be considered as an ambiguous address.

Calculating the full address from an abbreviated address

To calculate the full address from an abbreviate address, first, check whether the address has a double colon. If it has a double colon, determine how many 0 blocks are represented by the double colon. For this, count the number of blocks in the abbreviated address and subtract this number from 8. For example, in the address FF01::1, there are two blocks: FF01 and 1. The number of blocks expressed by the double colon (::) is 6 (8 blocks - 2 two blocks).

Once all 8 blocks are determined, count the number of hexadecimal digits in each block. Each block must contain 4 Hexadecimal digits. If any block contains less than 4 hexadecimal digits, add an equal number of zeros on the left side or in the leading position of the block.

Let's calculate the full address from the abbreviated example address.

Abbreviated example address

FF01::1

The address after removing abbreviated double-colon

FF01:0:0:0:0:0:0:1

The address after adding leading zeros in each block

FF01:0000:0000:0000:0000:0000:0000:0001

So the full address of the abbreviated address FF01::1 is the FF01:0000:0000:0000: 0000:0000: 0000:0001.

3.2.3 Packet Format and Extension Headers

Q12. Explain about IPv6 packet format and extension headers. **(Imp.)**

Ans :

The size of the IPv6 address is four times greater than the size of the IPv4 address but the IPv6 header size is only two times greater than the IPv4 header size.

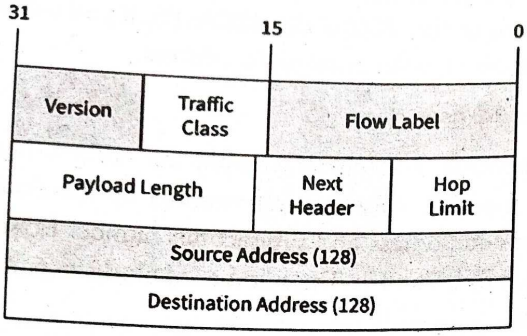
There is one fixed size header and zero or more than zero optional or extension headers in the IPv6 header. The fixed header keeps all the information that is important for the router. Optional information is kept in the extension header.

List of IPv6 Header Format Components

There are two main parts to the IPv6 data packet that is header and payload. The header of IPv6 is of fix length of 40 bytes which has the following fields:

Refer to the below image for the components of the IPv6 header

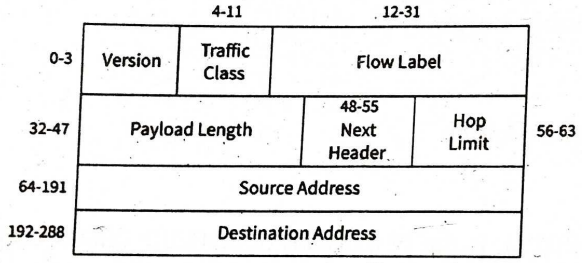
IPv6 Header Format



IPv6 Fixed Header

The size of the IPv6 fixed header is 40 bytes long and IPv6 header format consists of the following information:

Refer to the below image for the IPv6 fixed header



Version (4-bits) :

It shows the version of internet protocol we used, i.e. 0110

Traffic Class (8-bits) :

This is an 8-bit field in which 8 bits are divided into two parts. The most significant 6-bit is for the type of service so that the router will get to know about what services need to be provided to the given packet. And for Explicit Congestion Notification (ECN), the least significant 2-bit is used.

Flow Label (20-bits)

This 20-bit is required for maintaining the sequential flow of packets related to a particular communication. This field is also helpful in avoiding the reordering of packets. The source labels the sequence to help the router so that it can identify that a particular packet is related to a specific flow of data. It is generally used for real or streaming media.

Payload Length (16-bits)

This field is used to help the router in knowing how much information is stored in the payload of a particular packet.

Next Header (8-bits)

This field is used to represent the type of extension header or if the extension header is not present then it shows the Upper Layer PDU. The value for Upper Layer PDU is the same as that of values in IPv4.

Hop Limit (8-bits)

Hop limit is a field in a header that stops the header to go into an infinite loop in the network. It works the same as that of TTL in IPv4. When it passes a hop or router its value is decremented by 1. The packet is discarded when it reaches 0.

Source Address (128-bits)

This field provides the address from where the packet originates.

Destination Address (128-bits)

The destination address is the address of the packet's intended recipient.

IPv6 Extension Headers

The fixed headers in IPv6 store only the information which is necessary, instead of the information that is rarely used or not needed. All this rarely used or not required information is stored in the form of the extension header and placed between the fixed header and the upper header. A distinct value is used for the identification of the extension header.

In the IPv6 header format, the Fixed Header's next header points to the header that is the first extension header, when the extension header is used. After this, if one or more header is present in the extension header then, the next header field of the first extension header points to the second extension header and follows this process for the rest of the extension headers. The next header field of the last extension header points to the Upper Layer header.

If there is a 59 value in the next header field then, it shows that there is no header after this header. And also not even the Upper Layer header is here after this header.

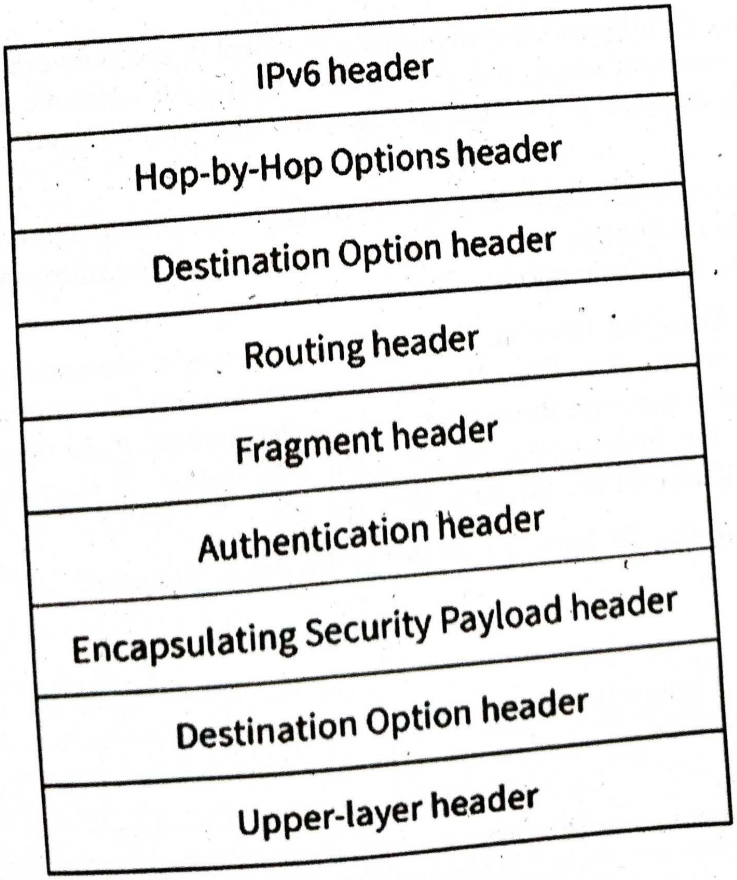
Following are some extension headers that must be supported according to RFC 2460:

Refer to the below image for the extensions header supported by RFC 2460.

Extension Header	Next Header Value	Description
Hop-by-hop Options Header	0	read by all devices in transit network
Routing Header	43	contains methods to support making routing decision
Fragment Header	44	contains parameters of datagram fragmentation
Destination Options Header	60	read by destination devices
Authentications Header	51	information regarding authenticity
Encapsulating Security Payload Header	50	encryption information

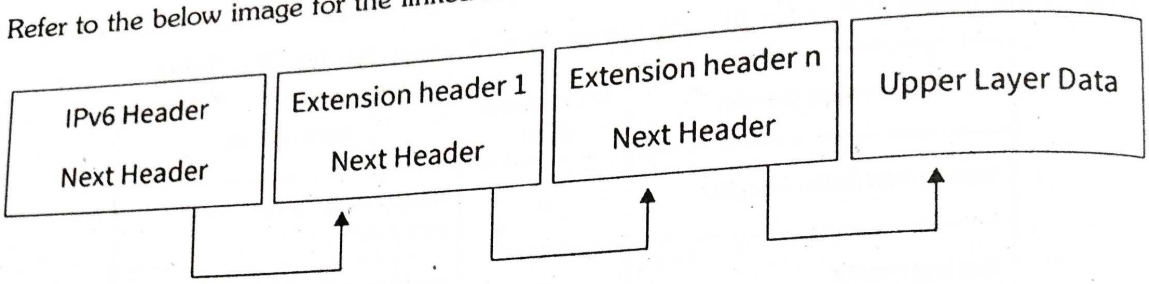
The sequence of the Extension headers is given below :

Refer to the below image for the sequence of the extension header.



The headers that are arranged one after another in a linked list format are known as extension headers. The extension header is shown in the given figure.

Refer to the below image for the linked list format of the extension header.



Rules of Headers

The order of the header is defined by some predefined rules which are given below:

- If there is a **hop-by-hop** option then it must be after the **base** header of the IPv6 header.
- All headers except the destination header must be present once in the list.
- If the destination header appears before the routing header, then all the intermediate nodes that are in the routing header examine the destination header.
- If the destination header is present before the upper layer, then only the destination nodes will examine the destination header.

3.2.4 ICMP

Q13. Explain about ICMP Protocol.

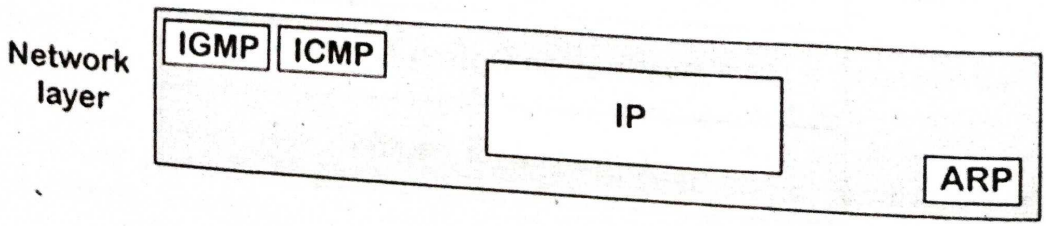
Ans :

The ICMP stands for Internet Control Message Protocol. It is a network layer protocol. It is used for error handling in the network layer, and it is primarily used on network devices such as routers. As different types of errors can exist in the network layer, so ICMP can be used to report these errors and to debug those errors.

For example, some sender wants to send the message to some destination, but the router couldn't send the message to the destination. In this case, the router sends the message to the sender that I could not send the message to that destination.

The IP protocol does not have any error-reporting or error-correcting mechanism, so it uses a message to convey the information. For example, if someone sends the message to the destination, the message is somehow stolen between the sender and the destination. If no one reports the error, then the sender might think that the message has reached the destination. If someone in-between reports the error, then the sender will resend the message very quickly.

The ICMP resides in the IP layer, as shown in the below diagram.



Messages

The ICMP messages are usually divided into two categories:

ICMP messages

Category	Type	Message
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

- **Error-reporting messages**

The error-reporting message means that the router encounters a problem when it processes an IP packet then it reports a message.

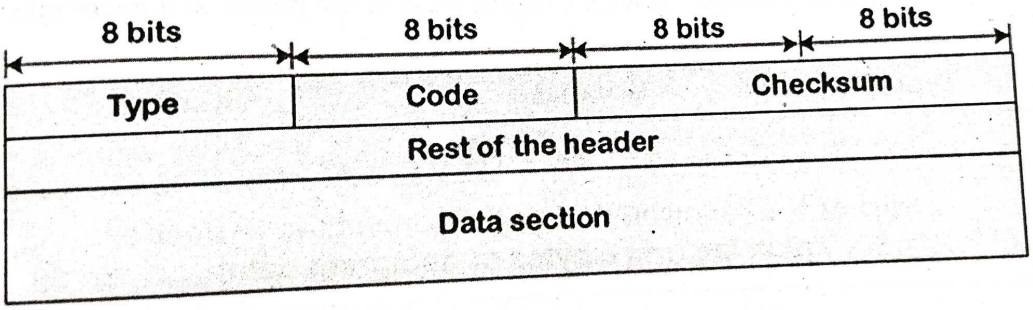
- **Query messages**

The query messages are those messages that help the host to get the specific information of another host. For example, suppose there are a client and a server, and the client wants to know whether the server is live or not, then it sends the ICMP message to the server.

ICMP Message Format

The message format has two things; one is a category that tells us which type of message it is. If the message is of error type, the error message contains the type and the code. The type defines the type of message while the code defines the subtype of the message.

The ICMP message contains the following fields



- **Type:** It is an 8-bit field. It defines the ICMP message type. The values range from 0 to 127 are defined for ICMPv6, and the values from 128 to 255 are the informational messages.
- **Code:** It is an 8-bit field that defines the subtype of the ICMP message
- **Checksum:** It is a 16-bit field to detect whether the error exists in the message or not.

Types of Error Reporting messages

The error reporting messages are broadly classified into the following categories:

- **Destination unreachable**

The destination unreachable error occurs when the packet does not reach the destination. Suppose the sender sends the message, but the message does not reach the destination, then the intermediate router reports to the sender that the destination is unreachable.

Type: 3	Code: 0 to 15	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

The above diagram shows the message format of the destination unreachable message. In the message format:

Type

It defines the type of message. The number 3 specifies that the destination is unreachable.

Code (0 to 15)

It is a 4-bit number which identifies whether the message comes from some intermediate router or the destination itself.

Sometimes the destination does not want to process the request, so it sends the destination unreachable message to the source. A router does not detect all the problems that prevent the delivery of a packet.

Source quench

There is no flow control or congestion control mechanism in the network layer or the IP protocol. The sender is concerned with only sending the packets, and the sender does not think whether the receiver is ready to receive those packets or is there any congestion occurs in the network layer so that the sender can send a lesser number of packets, so there is no flow control or congestion control mechanism. In this case, ICMP provides feedback, i.e., source quench. Suppose the sender resends the packet at a higher rate, and the router is not able to handle the high data rate. To overcome such a situation, the router sends a source quench message to tell the sender to send the packet at a lower rate.

Type: 4	Code: 0	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

The above diagram shows the message format of the source quench message. It is a type 4 message, and code is zero.

So, the sender must either stop or slow down the sending of datagrams until the congestion is reduced. The router sends one source-quench message for each datagram that is discarded due to the congestion in the network layer.

Time exceeded

Sometimes the situation arises when there are many routers that exist between the sender and the receiver. When the sender sends the packet, then it moves in a routing loop. The time exceeded is based on the time-to-live value. When the packet traverses through the router, then each router decreases the value of TTL by one. Whenever a router decreases a datagram with a time-to-live value to zero, then the router discards a datagram and sends the time exceeded message to the original source.

Each of the MAC layers has different data units. For example, some layers can handle upto 1500 data units, and some can handle upto 300 units. When the packet is sent from a layer having 1500 units to the layer having 300 units, then the packet is divided into fragments; this process is known as fragmentation. These 1500 units are divided into 5 fragments, i.e., f1, f2, f3, f4, f5, and these fragments reach the destination in a sequence. If all the fragments are not reached to the destination in a set time, they discard all the received fragments and send a time-exceeded message to the original source.

In the case of fragmentation, the code will be different as compared to TTL. Let's observe the message format of time exceeded.

Type: 11	Code: 0 or 1	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

The above message format shows that the type of time-exceeded is 11, and the code can be either 0 or 1. The code 0 represents TTL, while code 1 represents fragmentation. In a time-exceeded message, the code 0 is used by the routers to show that the time-to-live value is reached to zero.

The code 1 is used by the destination to show that all the fragments do not reach within a set time.

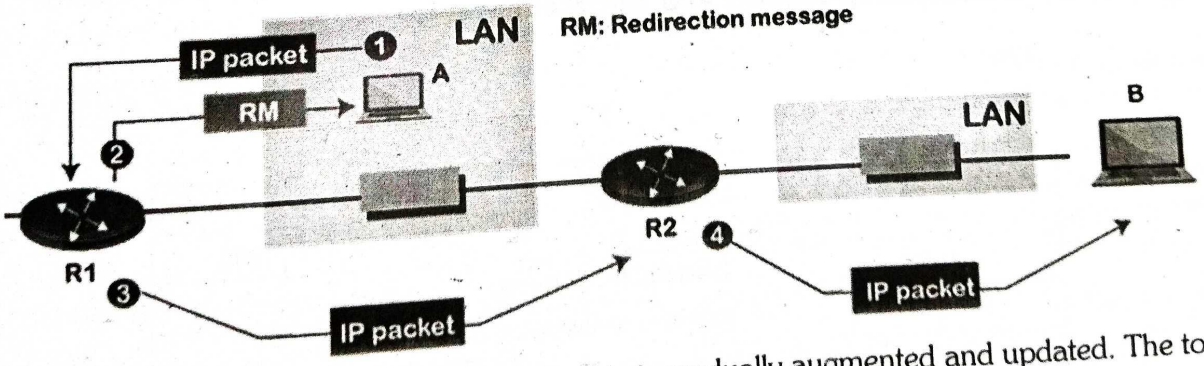
Parameter problems

The router and the destination host can send a parameter problem message. This message conveys that some parameters are not properly set.

Type: 12	Code: 0 or 1	Checksum
Pointer	Unused (All 0s)	
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

The above diagram shows the message format of the parameter problem. The type of message is 12, and the code can be 0 or 1.

Redirection



When the packet is sent, then the routing table is gradually augmented and updated. The tool used to achieve this is the redirection message. For example, A wants to send the packet to B, and there are two routers exist between A and B. First, A sends the data to the router 1. The router 1 sends the IP packet to router 2 and redirection message to A so that A can update its routing table.

ICMP Query Messages

The ICMP Query message is used for error handling or debugging the internet. This message is commonly used to ping a message.

Echo-request and echo-reply message

A router or a host can send an echo-request message. It is used to ping a message to another host that "Are you alive". If the other host is alive, then it sends the echo-reply message. An echo-reply message is sent by the router or the host that receives an echo-request message.

Key points of Query messages

1. The echo-request message and echo-reply message can be used by the network managers to check the operation of the IP protocol. Suppose two hosts, i.e., A and B, exist, and A wants to communicate with host B. The A host can communicate to host B if the link is not broken between A and B, and B is still alive.
2. The echo-request message and echo-reply message check the host's reachability, and it can be done by invoking the ping command.

The message format of echo-request and echo-reply message

Type 8: Echo request
Type 0: Echo reply

Type: 8 or 0	Code: 0	Checksum
Identifier		Sequence number
Optional data Sent by the request message; repeated by the reply message		

The above diagram shows the message format of the echo-request and echo-reply message. The type of echo-request is 8, and the request of echo-reply is 0. The code of this message is 0.

Timestamp-request and timestamp-reply message

The timestamp-request and timestamp-reply messages are also a type of query messages. Suppose the computer A wants to know the time on computer B, so it sends the timestamp-request message to computer B. The computer B responds with a timestamp-reply message.

Message format of timestamp-request and timestamp-reply

Type 13: request
Type 14: reply

Type: 13 or 14	Code: 0	Checksum
Identifier		Sequence number
Original timestamp		
Receive timestamp		
Transmit timestamp		

The type of timestamp-request is 13, and the type of timestamp-reply is 14. The code of this type of message is 0.

Key points related to timestamp-request and timestamp-reply message

- It can be used to calculate the round-trip time between the source and the destination, even if the clocks are not synchronized.
- It can also be used to synchronize the clocks in two different machines if the exact transit time is known.

If the sender knows the exact transit time, then it can synchronize the clock. The sender asks the time on the receiver's clock, and then it adds the time and propagation delay. Suppose the time is 1:00 clock and propagation delay is 100 ms, then time would be 1:00 clock plus 100 ms.

Debugging tools

There are several tools used for debugging. In this topic, we will learn two tools that use ICMP for debugging. The two tools are ping and traceroute. We have learned about ping in echo-request and echo-reply messages that check whether the host or a router is alive or running.

Now we will take a look at the traceroute

Traceroute is a tool that tracks the route taken by a packet on an IP network from source to destination. It records the time taken by the packet on each hop during its route from source to destination. Traceroute uses ICMP messages and TTL values. The TTL value is calculated; if the TTL value reaches zero, the packet gets discarded. Traceroute uses small TTL values as they get quickly expired. If the TTL value is 1 then the message is produced by router 1; if the TTL value is 2 then the message is produced by router 2, and so on.

Let's understand the traceroute through an example

Suppose A and B are two different hosts, and A wants to send the packet to the host B. Between A and B, 3 routers exist. To determine the location of the routers, we use the traceroute tool.

TTL value =1

First, host A sends the packet to router 1 with TTL value 1, and when the packet reaches to router 1 then router reduces the value of TTL by one and TTL values becomes 0. In this case, router 1 generates the time-exceeded message and host A gets to know that router 1 is the first router in a path.

TTL value=2

When host A sends the packet to router 1 with TTL value 2, and when the packet reaches to router 1 then the TTL value gets decremented by 1 and the TTL value becomes 1. Then router 1 sends the packet to router 2, and the TTL value becomes 0, so the router generates a time-exceeded message. The host A gets to know that router 2 is the second router on the path.

TTL value=3

When host A sends the packet to router 1 with TTL value 3, then the router decrements its value by one, and the TTL value becomes 2. Then, router 1 sends the packet to router 2, and the TTL value becomes 1. Then, router 2 sends the packet to router 3, and the TTL value becomes 0. As TTL value becomes 0, router 3 generates a time-exceeded message. In this way, host A is the third router on a path.

3.2.5 IGMP

Q14. Explain about IGMP protocol.

Ans :

IGMP is an abbreviated form of Internet Group Management Protocol(IGMP). Mainly the Internet Protocol can be involved in the two types of communication i.e, Unicasting and multicasting. IGMP is one of the necessary but not the efficient protocol that is involved in Multicasting.

IGMP is basically a companion of Internet Protocol (IP).

IGMP is not a multicasting routing protocol but it is a protocol that manages the group membership. This protocol mainly helps the multicast routers in order to create and update a list of loyal members that are related to each router interface.

This protocol is used in streaming videos, gaming, or web conferencing tools.

IGMP Messages

There are two versions of IGMP: IGMPv1 and IGMPv2.

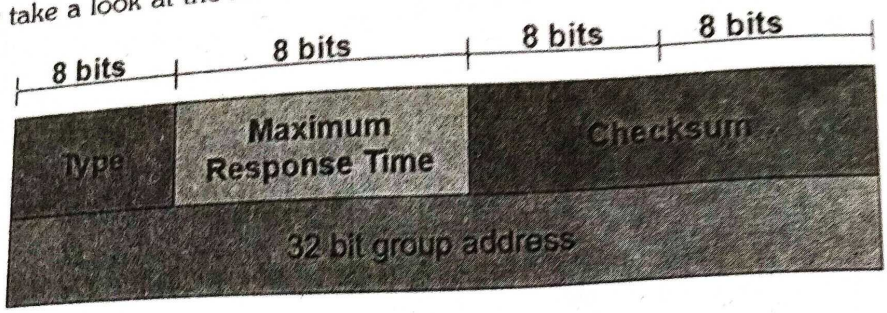
The version IGMPv2 has three types of messages:

- The Query
- The Membership report
- The Leave report.

There are two types of Query messages: General and Special

Message Format

Let us now take a look at the format of IGMP(Version 2):



Type

This is an 8-bit field and is mainly used to define the type of the message. The value of the type can be in both hexadecimal as well as binary notations.

Maximum Response Time

The size of this field is also 8 bit and it mainly defines the amount of time in which query must be answered. The value of this field is nonzero in the query message; while its value is zero in the other two types.

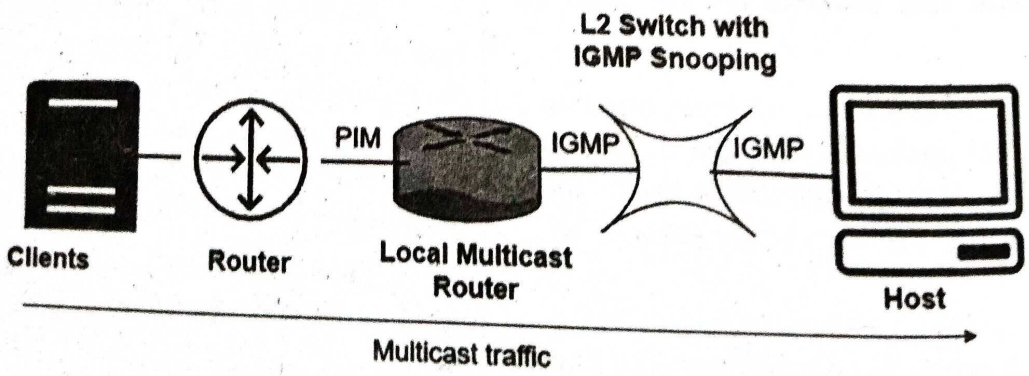
Working of IGMP

This protocol mainly works on the device that has the capability of handling multiple groups and used for dynamic multicasting; these devices mainly allow the host in order to join or leave the membership in the multicast group.

Also, these devices are allowed to add and remove the clients from the group.

IGMP protocol mainly operates in between the host and local multicast router

At the time when there is a creation of the multicast group, the multicast group address is in the range of class D (224-239) IP addresses and is forwarded as the destination IP address in the packet.



L2 means level-2 devices like switches; these are mainly used in between the host and multicast router for the snooping of IGMP.

IGMP snooping

It is a process that is used to listen to the IGMP network traffic in a controlled manner.

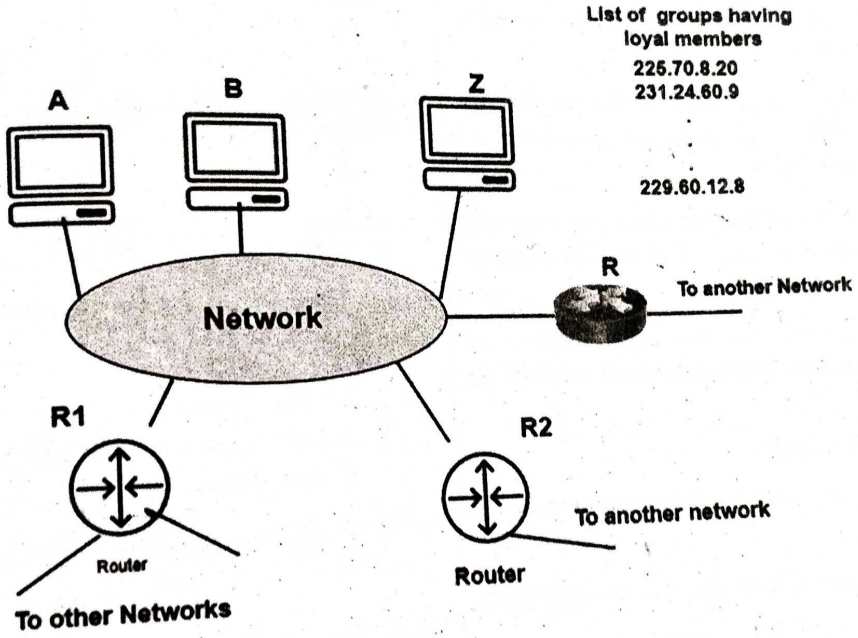
The switch mainly receives the message from the host and then forwards the membership report mainly to the local multicast router. After that, the multicast traffic is then further forwarded to remote routers from local multicast routers using PIM (Protocol Independent Multicast) protocol so that the clients can receive the message/data packets.

If the Clients wish to join the network then they can send a join message in the query and then the switch intercepts the message and then adds the ports of clients to its multicast routing table.

IGMP Operation

The Internet Group Management Protocol operates locally. The Multicast router that is connected to the network mainly has a list of multicast addresses of the group with at least one loyal member in that network. And for each group, there is mainly one router that has the duty of distributing the multicast packets destined for that group.

This simply indicates that in the case if there are three multicast routers connected to a network then their list of groupids are mutually exclusive.



Given below are the operations of IGMP:

Joining a Group

- In this operation, both the host and the router can join a group. Whenever a process on the host wants to join a group then it simply sends the request to the host. After that, the host then adds the name of the process and the name of the group to its list.
- In case, if this is the first entry of that particular group, then the host sends the membership report message to the multicast router of the group.
- And if the entry is not the first entry then there is no need of sending such a message.

Leaving a group

- Whenever the host finds that there is no process that is interested in the group then it mainly leaves a report message.

- The membership is not disinfected by the multicast router of the group, rather than it immediately transmits the query packets repeatedly to see if anyone is still interested or not.
- And in case if it gets the response in the form of a membership report message then the membership of the host or network is preserved.
- **Monitoring Membership:** Mainly the general query message does not define a particular group.
- **Delayed Response:** In order to prevent unnecessary traffic, the IGMP mainly makes use of a delayed response strategy.

Advantages of IGMP

The listed below are some of the benefits offered by the IGMP:

- With the help of this protocol, the bandwidth is utilized efficiently aa because all the shared links are connected.
- Using this protocol, the host can immediately leave a multicast group and then join another group.
- The performance of this protocol is optimized as there is no transmission of junk packets to the host.

Disadvantages of IGMP

Given below are some of the drawbacks of the IGMP :

- During filtering and security, it does not offers good efficiency.
- This protocol is vulnerable to Denial of Service (DoS) attacks.
- Network congestion can occur due to a lack of TCP.

3.2.6 ARP

Q15. What is ARP? Explain about it.

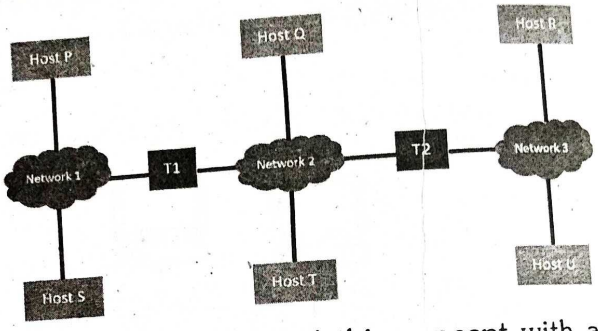
Ans :

Address Resolution Protocol (ARP) is an important protocol of the network layer in the OSI model, which helps find the MAC (Media Access Control) address given the system's IP address. The ARP's main task is to convert the 32-bit IP address (for IPv4) to a 48-bit MAC address.

This protocol is mostly used to determine the hardware (MAC) address of a device from an IP address. It is also used when one device wants to communicate with some other device on a local network. The full form of ARP is Address Resolution Protocol.

Working Mechanism

All OS in an IPv4 network keeps an ARP cache. When the host requests a MAC address to send a packet to another host in the LAN, it checks its ARP cache to check that the MAC address translation already presents.



Let us understand this concept with an example:

- Hosta P resolves protocol address for host U for protocol messages from an application on P sent to U.
- P does not resolve a protocol address for host U
- By using the internet layer, host P delivers to host U by routing through T1 and T2.
- Host P resolves the T1 hardware address.
- Network layer on host P passes packet containing destination protocol address for U for delivery to T1
- T1 delivers the packet to T2 which in turn forwards the packet to Host U.

Address Resolution Methods

Association between a protocol address and a hardware address is known as binding.

There are three techniques used for this purpose:

- **Table lookup:** Bindings stored in memory with protocol address as the key. It uses the data link layer to check the protocol address to find the hardware address.
- **Dynamic:** This type of network messaging method is used for "just-in-time" resolution. Data link layer sends message requests in a hardware address. destination responds.
- **Closed-form computation:** In this method, a protocol address is based on a hardware address. Data link layer derives the hardware address from the protocol address.

Types

Here are four types of Address Resolution Protocol, which is given below:

1. Proxy ARP
2. Gratuitous ARP
3. Reverse ARP
4. Inverse ARP

1. Proxy ARP

In the Proxy ARP method, Layer 3 devices can respond to ARP requests. This ARP type is configured router will respond to the target IP address and maps the router's MAC address with the target IP address and sender when it is reached to its destination.

2. Gratuitous ARP

Gratuitous is another type of ARP request of the host. This type of ARP request helps the network to identify the duplicate IP address. Therefore, when an ARP request is sent by a router or switch to get its IP address, no ARP responses are received so that no other nodes can use the IP address allocated to that switch or router.

3. Reverse ARP (RARP)

Reverse ARP, also now called RARP, is a type of ARP networking protocol which is used by the client system in a LAN to request its IPv4 address from the ARP router table. The network admin mostly creates a table in the gateway-router, which helps determine the MAC address to that specific IP address.

4. Inverse ARP (InARP)

Inverse ARP is also called InARP, is a type of ARP used to find the nodes' IP of addresses from the data link layer addresses. InARP is widely used for ATM networks frame relays where Layer 2 virtual circuit addressing acquired from Layer 2 signaling.

ARP Header

Hardware Type		Protocol Type
Hardware Length	Protocol Length	Operation Request 1, Reply 2
Sender hardware address (For example, 6 bytes for Ethernet)		
Sender protocol address (For example, 4 bytes for IP)		
Target hardware address (For example, 4 bytes for IP)		
Target Protocol address (For example, 4 bytes for IP)		

ARP header

- **Hardware Type:** It is 1 for Ethernet.
- **Protocol Type:** It is a protocol used in the network layer.
- **Hardware Address Length:** It is the length in bytes so that it would be 6 for Ethernet.
- **Protocol Address Length:** Its value is 4 bytes.
- **Operation Code:** indicates that the packet is an ARP Request (1) or an ARP Response (2).
- **Senders Hardware Address:** It is a hardware address of the source node.
- **Senders Protocol Address:** It is a layer 3 address of the source node.
- **Target Hardware Address:** It is used in a RARP request, which response impact both the destination's hardware and layer 3 addresses.
- **Target Protocol Address:** It is used in an ARP request when the response carries both layer 3 addresses and the destination's hardware.

Advantages of using ARP

Here are the pros/benefits of using ARP

- If you are using ARP, then MAC addresses can easily be known if you know the IP address of the same system.
- End nodes should not be configured to "know" MAC addresses. It can be found when needed.
- ARP's goal is to enable each host on a network that allows you to build up a mapping between IP addresses and physical addresses.
- The set of mappings or table stored in the host is called ARP table or ARP cache.

3.2.7 RARP

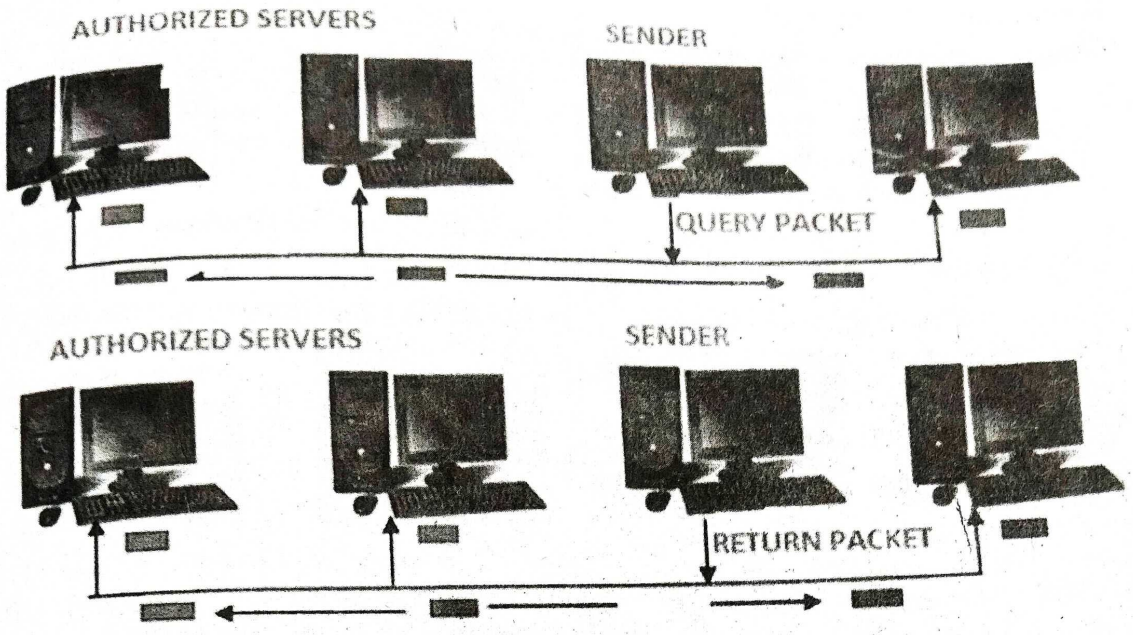
Q16. Explain about Reverse Address Resolution Protocol

Ans :

RARP is abbreviation of Reverse Address Resolution Protocol which is a protocol based on computer networking which is employed by a client computer to request its IP address from a gateway server's Address Resolution Protocol table or cache. The network administrator creates a table in gateway-router, which is used to map the MAC address to corresponding IP address.

This protocol is used to communicate data between two points in a server. The client doesn't necessarily need prior knowledge the server identities capable of serving its request. Media Access Control (MAC) addresses requires individual configuration on the servers done by an administrator. RARP limits to the serving of IP addresses only.

When a replacement machine is set up, the machine may or might not have an attached disk that may permanently store the IP Address so the RARP client program requests IP Address from the RARP server on the router. The RARP server will return the IP address to the machine under the belief that an entry has been setup within the router table.



REVERSE ADDRESS RESOLUTION PROTOCOL

History

RARP was proposed in 1984 by the university Network group. This protocol provided the IP Address to the workstation. These diskless workstations were also the platform for the primary workstations from Sun Microsystems.

Working

The RARP is on the Network Access Layer and is employed to send data between two points in a very network.

Each network participant has two unique addresses:- IP address (a logical address) and MAC address (the physical address).

The IP address gets assigned by software and after that the MAC address is constructed into the hardware.

The RARP server that responds to RARP requests, can even be any normal computer within the network. However, it must hold the data of all the MAC addresses with their assigned IP addresses. If a RARP request is received by the network, only these RARP servers can reply to it. The info packet needs to be sent on very cheap layers of the network. This implies that the packet is transferred to all the participants at the identical time.

The client broadcasts a RARP request with an Ethernet broadcast address and with its own physical address. The server responds by informing the client its IP address.

Q17. How is RARP different from ARP ?

Ans :

Sl.No.	RARP	ARP
1.	RARP stands for Reverse Address Resolution Protocol	ARP stands for Address Resolution Protocol
2.	In RARP, we find our own IP address	In ARP, we find the IP address of a remote machine
3.	The MAC address is known and the IP address is requested	The IP address is known, and the MAC address is being requested
4.	It uses the value 3 for requests and 4 for responses	It uses the value 1 for requests and 2 for responses

Q18. Sate the uses and disadvantage of RARP?

Ans :

Uses of RARP

RARP is used to convert the Ethernet address to an IP address. It is available for the LAN technologies like FDDI, token ring LANs, etc.

Disadvantages

The Reverse Address Resolution Protocol had few disadvantages which eventually led to its replacement by BOOTP and DHCP. Some of the disadvantages are listed below:

- The RARP server must be located within the same physical network.
- The computer sends the RARP request on very cheap layer of the network. Thus, it's unattainable for a router to forward the packet because the computer sends the RARP request on very cheap layer of the network.
- The RARP cannot handle the subnetting process because no subnet masks are sent. If the network is split into multiple subnets, a RARP server must be available with each of them.
- It isn't possible to configure the PC in a very modern network.
- It doesn't fully utilize the potential of a network like Ethernet

3.3 Congestion Control and Resource Allocation

3.3.1 Problem

Q19. What is congestion control problem for assigning resources

Ans :

The main problem here is , how to effectively and fairly allocate resources among a collection of competing users. The resources being shared include the bandwidth of the links and the buffers on the routers or switches where packets are queued awaiting transmission. Packets *contend* at a router for the use of a link, with each contending packet placed in a queue waiting its turn to be transmitted over the link. When too many packets are contending for the same link, the queue fills and two undesirable things happen: packets experience increased end-to-end delay, and in the worst case, the queue overflows and packets have to be dropped. When long queues persist and drops become common, the network is said to be *congested*. Most networks provide a *congestion-control* mechanism to deal with just such a situation.

Congestion control and resource allocation are two sides of the same coin. On the one hand, if the network takes an active role in allocating resources - for example, scheduling which virtual circuit gets to use a given physical link during a certain period of time - then congestion may be avoided, thereby making congestion control unnecessary. Allocating network resources with any precision is difficult, however, because the resources in question are distributed throughout the network; multiple links connecting a series of routers need to be scheduled. On the other hand, you can always let packet sources send as much data as they want and then recover from congestion should it occur. This is the easier approach, but it can be disruptive because many packets may be discarded by the network before congestion can be controlled.

Congestion control and resource allocation involve both hosts and network elements such as routers. In network elements, various queuing disciplines can be used to control the order in which packets get transmitted and which packets get dropped. The queuing discipline can also segregate traffic to keep one user's packets from unduly affecting another user's packets. At the end hosts, the congestion-control mechanism paces how fast sources are allowed to send packets. This is done in an effort to keep congestion from occurring in the first place and, should it occur, to help eliminate the congestion.

3.3.2 ISSUES

Q20. What are the issues in allocating resources ? Write about it. **(Imp.)**

Ans :

Resource allocation is partially implemented in the routers, switches, and links inside the network and partially in the transport protocol running on the end hosts. End systems may use signalling protocols to convey their resource requirements to network nodes, which respond with information about resource availability.

It is also important to understand the difference between flow control and congestion control. Flow control involves keeping a fast sender from overrunning a slow receiver

Network Model

We begin by defining three salient features of the network architecture.

Packet-Switched Network

We consider resource allocation in a packet-switched network (or internet) consisting of multiple links and switches (or routers). Since most of the mechanisms described in this chapter were designed for use on the Internet, and therefore were originally defined in terms of routers rather than switches, we use the term *router* throughout our discussion. The problem is essentially the same, whether on a network or an internetwork.

In such an environment, a given source may have more than enough capacity on the immediate outgoing link to send a packet, but somewhere in the middle of a network its packets encounter a link that is being used by many different traffic sources. Figure 1 illustrates this situation—two high-speed links are feeding a low-speed link. This is in contrast to shared-access networks like Ethernet and wireless networks, where the source can directly observe the traffic on the network and decide accordingly whether or not to send a packet.

Connectionless Flows

We assume that the network is essentially connectionless, with any connection-oriented service implemented in the transport protocol that is running on the end hosts

This is precisely the model of the Internet, where IP provides a connectionless datagram delivery service and TCP implements an end-to-end connection abstraction.

The major shortcoming of this approach is that it leads to an underutilization of resources—buffers reserved for a particular circuit are not available for use by other traffic even if they were not currently being used by that circuit.

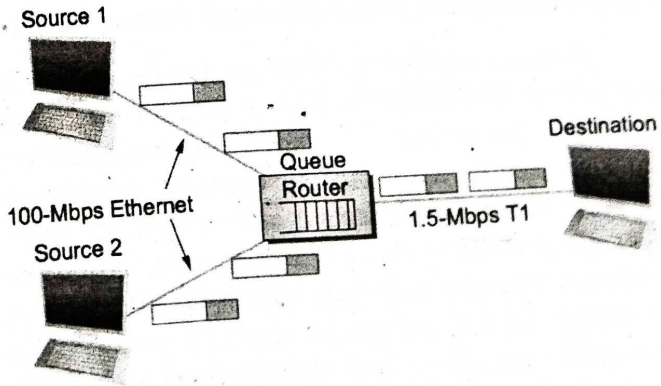


Fig.1: A potential bottleneck router

We need to qualify the term *connectionless* because our classification of networks as being either connectionless or connection oriented is a bit too restrictive; there is a gray area in between. In particular, the assumption that all datagrams are completely independent in a connectionless network is too strong. The datagrams are certainly switched independently, but it is usually the case that a stream of datagrams between a particular pair of hosts flows through a particular set of routers. This idea of a *flow*—a sequence of packets sent between a source/destination pair and following the same route through the network—is an important abstraction in the context of resource allocation.

One of the powers of the flow abstraction is that flows can be defined at different granularities. For example, a flow can be host-to-host (i.e., have the same source/destination host addresses) or process-to-process (i.e., have the same source/destination host/port pairs). In the latter case, a flow is essentially the same as a channel, as we have been using that term throughout this book. The reason we introduce a new term is that a flow is visible to the routers inside the network, whereas a channel is an end-to-end abstraction. Figure 153 illustrates several flows passing through a series of routers.

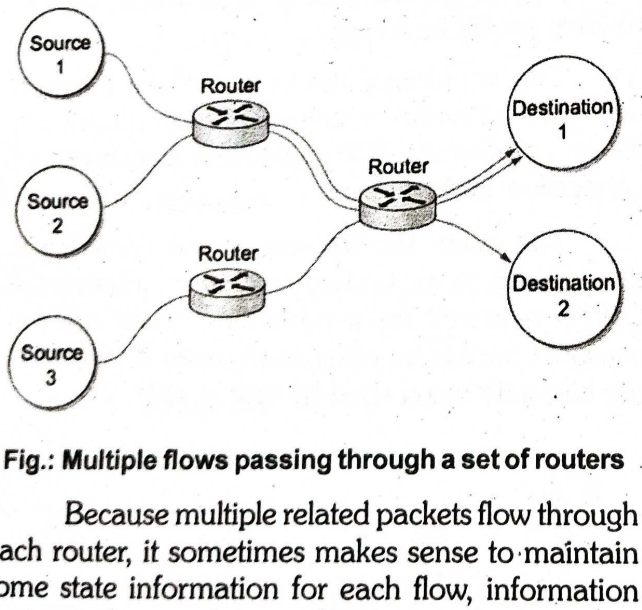


Fig.: Multiple flows passing through a set of routers

Because multiple related packets flow through each router, it sometimes makes sense to maintain some state information for each flow, information that can be used to make resource allocation decisions about the packets that belong to the flow. This state is sometimes called *soft state*. The main difference between soft state and hard state is that soft state need not always be explicitly created and removed by signalling. Soft state represents a middle ground between a purely connectionless network that maintains *no* state at the routers and a purely connection-oriented network that maintains hard state at the routers. In general, the correct operation of the network does not depend on soft state being present (each packet is still routed correctly without regard to this state), but when a packet happens to belong to a flow for which the router is currently maintaining soft state, then the router is better able to handle the packet.

Note that a flow can be either implicitly defined or explicitly established. In the former case, each router watches for packets that happen to be traveling between the same source/destination pair- the router does this by inspecting the addresses in the header - and treats these packets as belonging to the same flow for the purpose of congestion control. In the latter case, the source sends a flow setup message across the network, declaring that a flow of packets is about to start.

Taxonomy

There are countless ways in which resource allocation mechanisms differ, so creating a thorough taxonomy is a difficult proposition. We describe three dimensions along which resource allocation mechanisms can be characterized.

Router-Centric versus Host-Centric

Resource allocation mechanisms can be classified into two broad groups: those that address the problem from inside the network (i.e., at the routers or switches) and those that address it from the edges of the network (i.e., in the hosts, perhaps inside the transport protocol).

In a router-centric design, each router takes responsibility for deciding when packets are forwarded and selecting which packets are to be dropped, as well as for informing the hosts that are generating the network traffic how many packets they are allowed to send. In a host-centric design, the end hosts observe the network conditions (e.g., how many packets they are successfully getting through the network) and adjust their behavior accordingly.

Reservation-Based versus Feedback-Based

A second way that resource allocation mechanisms are sometimes classified is according to whether they use *reservations* or *feedback*. In a reservation-based system, some entity (e.g., the end host) asks the network for a certain amount of

capacity to be allocated for a flow. Each router then allocates enough resources (buffers and/or percentage of the link's bandwidth) to satisfy this request. If the request cannot be satisfied at some router, because doing so would overcommit its resources, then the router rejects the reservation. This is analogous to getting a busy signal when trying to make a phone call. In a feedback-based approach, the end hosts begin sending data without first reserving any capacity and then adjust their sending rate according to the feedback they receive. This feedback can be either explicit or implicit.

Window Based versus Rate Based

A third way to characterize resource allocation mechanisms is according to whether they are window based or rate based. This is one of the areas, noted above, where similar mechanisms and terminology are used for both flow control and congestion control. Both flow-control and resource allocation mechanisms need a way to express, to the sender, how much data it is allowed to transmit. There are two general ways of doing this: with a *window* or with a *rate*. We have already seen window-based transport protocols, such as TCP, in which the receiver advertises a window to the sender. This window corresponds to how much buffer space the receiver has, and it limits how much data the sender can transmit; that is, it supports flow control. A similar mechanism - window advertisement - can be used within the network to reserve buffer space (i.e., to support resource allocation). TCP's congestion-control mechanisms are window based.

Evaluation Criteria

The final issue is one of knowing whether a resource allocation mechanism is good or not. Recall that in the problem statement at the start of this chapter we posed the question of how a network effectively and fairly allocates its resources.

There re two ways by which a resource allocation scheme can be evaluated.

Effective Resource Allocation

A good starting point for evaluating the effectiveness of a resource allocation scheme is to consider the two principal metrics of networking: throughput and delay. Clearly, we want as much throughput and as little delay as possible. Unfortunately, these goals are often somewhat at odds with each other. One sure way for a resource allocation algorithm to increase throughput is to allow as many packets into the network as possible, so as to drive the utilization of all the links up to 100%. We would do this to avoid the possibility of a link becoming idle because an idle link necessarily hurts throughput. The problem with this strategy is that increasing the number of packets in the network also increases the length of the queues at each router. Longer queues, in turn, mean packets are delayed longer in the network.

This ratio is sometimes referred to as the *power* of the network:

Power = Throughput / Delay

Fair Resource Allocation

The effective utilization of network resources is not the only criterion for judging a resource allocation scheme. We must also consider the issue of fairness. However, we quickly get into murky waters when we try to define what exactly constitutes fair resource allocation. For example, a reservation-based resource allocation scheme provides an explicit way to create controlled unfairness. With such a scheme, we might use reservations to enable a video stream to receive 1 Mbps across some link while a file transfer receives only 10 kbps over the same link.

In the absence of explicit information to the contrary, when several flows share a particular link, we would like for each flow to receive an equal share of the bandwidth. This definition presumes that a *fair* share of bandwidth means an *equal* share of bandwidth. But, even in the absence of reservations, equal shares may not equate to fair shares.

3.3.3 Queuing

Q21. Explain about the queuing algorithms in congestion control. **(Imp.)**

Ans :

The queuing algorithm can be thought of as allocating both bandwidth and buffer space .It also directly affects the latency experienced by a packet by determining how long a packet waits to be transmitted.

There are two common queuing algorithms - first-in, first-out (FIFO) and fair queuing (FQ)/

FIFO

The idea of FIFO queuing, also called first-come, first-served (FCFS) queuing, is simple: The first packet that arrives at a router is the first packet to be transmitted

This is illustrated in Figure 1(a), which shows a FIFO with "slots" to hold up to eight packets. Given that the amount of buffer space at each router is finite, if a packet arrives and the queue (buffer space) is full, then the router discards that packet, as shown in Figure 1(b). This is done without regard to which flow the packet belongs to or how important the packet is. This is sometimes called *tail drop*, since packets that arrive at the tail end of the FIFO are dropped.

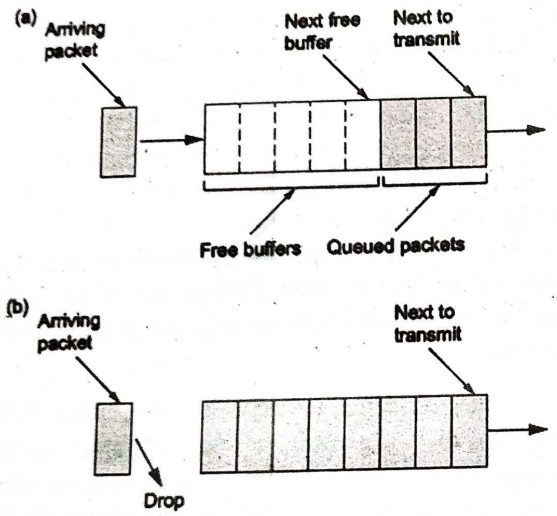


Fig.: FIFO queuing (a), and tail drop at a FIFO queue (b)

Note that tail drop and FIFO are two separable ideas. FIFO is a scheduling discipline - it determines the order in which packets are transmitted. Tail drop is a drop policy - it determines which packets get dropped. Because FIFO and tail drop are the simplest instances of scheduling discipline and drop policy, respectively, they are sometimes viewed as a bundle.

FIFO with tail drop, as the simplest of all queuing algorithms, is the most widely used in Internet routers at the time of writing. This simple approach to queuing pushes all responsibility for congestion control and resource allocation out to the edges of the network. Thus, the prevalent form of congestion control in the Internet currently assumes no help from the routers: TCP takes responsibility for detecting and responding to congestion.

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

Fair Queuing

This algorithm finds its role incontrolling congestion in datagram. Itwas found out that fair queuing provides several important advantages over the usual firstcome-first-serve queuing algorithm .

Congestion in the datagram networks can be controlled in two ways:

1. At the source, where flow control algorithms vary the rate at which source sendspackets.
2. At the gateway, where routing and queuing algorithms control the congestion. We are going to assume that we are using the best flow control algorithms and hence in this paper we are going to discuss only about queuing algorithms.

Queuing algorithms can be thought of as allocating three nearly independent quantities: bandwidth (which packets get transmitted), promptness (when do those packets get transmitted), and buffer space (which packets are discarded by the gateway).

Before this first-come-first-serve (FCFS) was the most common algorithm. But this algorithm was fraught with malfunctions such as allocation of higher bandwidth to a source, sending packets at sufficiently high speed. So in a fair queuing algorithm in which gateways maintain separate queues for packets from each individual source and serviced them in a round-robin manner.

In circuit switched networks where there is explicit buffer reservation and uniform packet size,

it has been established that round robin service disciplines allocate bandwidth fairly. But in case of other networks, a source using long packets gets more bandwidth than one using short bandwidth, so bandwidth is not allocated fairly.

fair allocation

Let's consider allocation of a single resource among N users as an example. Assume there is an amount total of this resource and each user requests an amount 0i and, under a particular allocation, receives an amount ii. The max-min fairness criterion states that an allocation is fairif (1) no user receives more than it requests, (2) no other allocation scheme satisfying condition 1 has a higher minimum allocation, and condition 2 remains recursively true as we remove the minimal user and reduce the total resource accordingly,

μtotal ← μtotal - μmin.

This condition reduces to μi = MIN(μfair, μi) in the simple example, with μfair, the fair share, being set so that $\mu total = \sum_1^N = 1\mu_{1\,i}$

Allocation on the basis of source-destination pairs, or conversations, will constitute a user.

Definition of Algorithm

Owing to varying packet-sizes, fair allocation of bandwidths is not an easy task. To see how this unfairness can be avoided, a hypothetical service discipline where transmission occurs in a bit by bit round robin fashion.

In this the bandwidth is allocated fairly since at every instant in time, each conversation is receiving its fair share. Let R(t) denote the number of rounds made in the round-robin service discipline up to time t (R(t) is a continuous function, with the fractional part indicating partially completed rounds). Let Nac(t) denote the number of active conversations, i.e. those that have bits in their queue at time t. Then, $\frac{\partial R}{\partial t} = \frac{\mu}{N_{ac}(t)}$, where μ is the

Line speed of the gateway's outgoing line (we will, for convenience, work in units such that μ= 1). A packet of size P whose first bit ge ts serviced at time to will have its last bit serviced P rounds later, at time t such that R(t) = R(to) + P. Let tt_ be the time that packet i belonging to conversation _ arrives at the gateway, and define the numbers Si_ and Fi_ as the values of R(t) when the packet started and finished service. With Pi_ denoting the size of the packet, the following relations hold: $F_i^\alpha = S_i^\alpha + P_i^\alpha$ and $S_i = MAX(Fi\text{-}1, R(t_i^\alpha))$.

Sending packets in a bit-by-bit round robin fashion while satisfying our requirements for an adequate queuing algorithm, is obviously unrealistic. This impractical algorithm is tried to be emulated in a practical packet-by-packet transmission scheme. A natural way to emulate the bit-by-bit round-robin algorithm is to let the quantities F_i^α define the sending order of the packets. Our packet-by-packet transmission algorithm is simply defined by the rule that, whenever a packet finishes transmission, the next packet sent is the one with the smallest value of F_i^α. In a preemptive version of this algorithm, newly arriving packets whose finishing number F_i^α is smaller than that of the packet currently in transmission preempt the transmitting packet.

Promptness allocation must be based solely on data already available at the gateway. One such allocation strategy is to give more promptness (less delay) to users who utilize less than their fair share of bandwidth. Separating the promptness allocation from the bandwidth allocation can be accomplished by introducing a nonnegative parameter :, and defining a new quantity, the bid B_i, via $B_i^a = P_i^a + MAX(Fi\text{-}1^a, R(t_i^a)\text{-}d)$. The quantities R(t), Nac(t), S_i^a and F_i remain as before, but now the sending order is determined by the B's, not the F's. The asymptotic band width allocation is independent odd. since the F's control the bandwidth allocation, but the algorithm gives slightly faster service to packets that arrive at an inactive conversation.

3.3.4 TCP

Q22. Explain about TCP algorithms for congestion control. **(Imp.)**

Ans :

TCP reacts to congestion by reducing the sender window size.

The size of the sender window is determined by the following two factors-

Receiver Window Size-

Receiver window size is an advertisement of-

"How much data (in bytes) the receiver can receive without acknowledgment?"

- Sender should not send data greater than receiver window size.
- Otherwise, it leads to dropping the TCP segments which causes TCP Retransmission.
- So, sender should always send data less than or equal to receiver window size.
- Receiver dictates its window size to the sender through TCP Header.

2. Congestion Window

- Sender should not send data greater than congestion window size.
- Otherwise, it leads to dropping the TCP segments which causes TCP Retransmission.
- So, sender should always send data less than or equal to congestion window size.
- Different variants of TCP use different approaches to calculate the size of congestion window.
- Congestion window is known only to the sender and is not sent over the links.

So, always-

Sender window size = Minimum (Receiver window size, Congestion window size)

TCP Congestion Policy

TCP's general policy for handling congestion consists of following three phases-

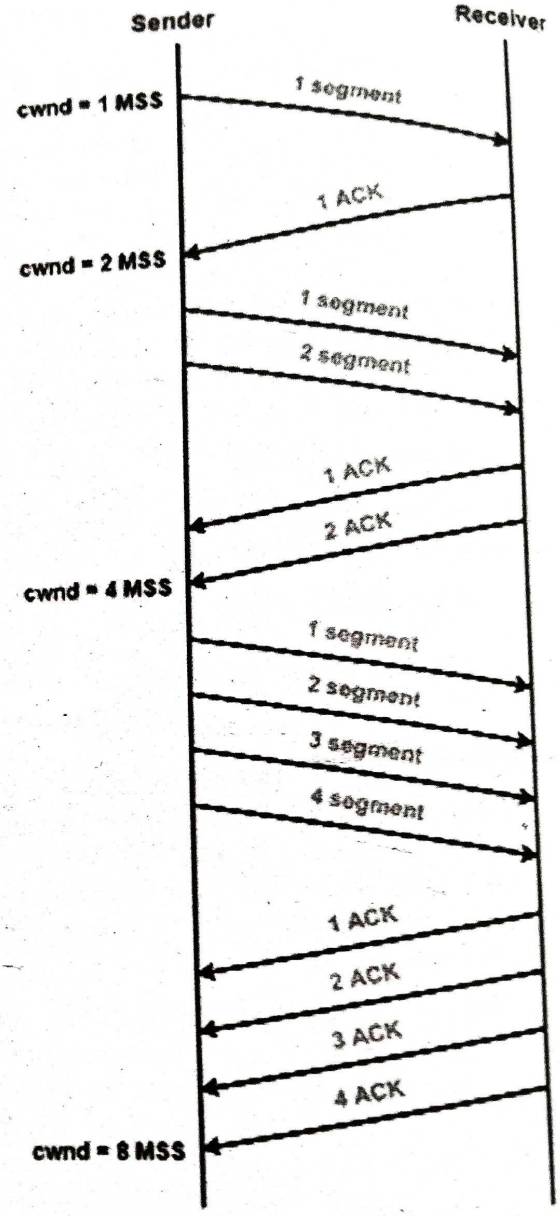
1. Slow Start Phase

- Initially, sender sets congestion window size = Maximum Segment Size (1 MSS).
- After receiving each acknowledgment, sender increases the congestion window size by 1 MSS.
- In this phase, the size of congestion window increases exponentially.

The followed formula is-

Congestion window size = Congestion window size + Maximum segment size

This is shown below-



(cwnd = congestion window size)

- After 1 round trip time, congestion window size = $(2)^1$ = 2 MSS
- After 2 round trip time, congestion window size = $(2)^2$ = 4 MSS
- After 3 round trip time, congestion window size = $(2)^3$ = 8 MSS and so on.

This phase continues until the congestion window size reaches the slow start threshold.

Threshold

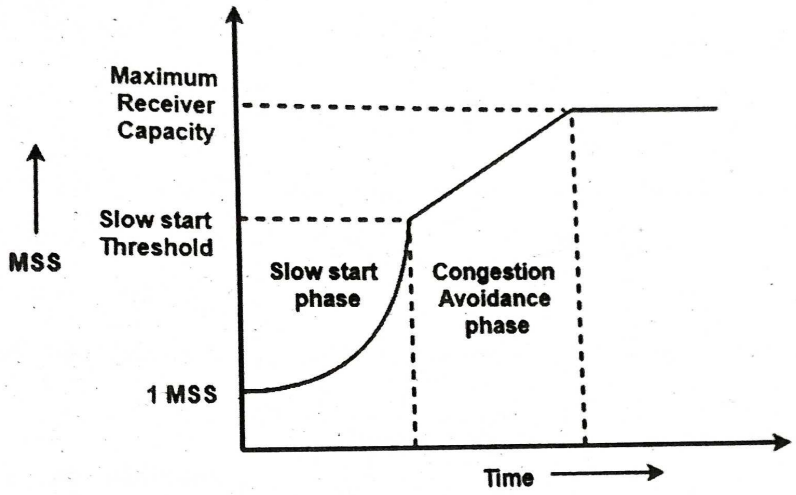
= Maximum number of TCP segments that receiver window can accommodate / 2

= (Receiver window size / Maximum Segment Size) / 2

2. **Congestion Avoidance Phase-**

After reaching the threshold,

- Sender increases the congestion window size linearly to avoid the congestion.
- On receiving each acknowledgments, sender increments the congestion window size by 1.
- The followed formula is- Congestion window size = Congestion window size + 1
- This phase continues until the congestion window size becomes equal to the receiver window size.



3. **Congestion Detection Phase**

When sender detects the loss of segments, it reacts in different ways depending on how the loss is detected-

Case-01: Detection On Time Out

- Time Out Timer expires before receiving the acknowledgment for a segment.
- This case suggests the stronger possibility of congestion in the network.
- There are chances that a segment has been dropped in the network.

Reaction

In this case, sender reacts by-

- Setting the slow start threshold to half of the current congestion window size.
- Decreasing the congestion window size to 1 MSS.
- Resuming the slow start phase.

Case-02: Detection On Receiving 3 Duplicate Acknowledgments-

- Sender receives 3 duplicate acknowledge-ments for a segment.
- This case suggests the weaker possibility of congestion in the network.
- There are chances that a segment has been dropped but few segments sent later may have reached.

Reaction

In this case, sender reacts by-

- Setting the slow start threshold to half of the current congestion window size.
- Decreasing the congestion window size to slow start threshold.
- Resuming the congestion avoidance phase.

Short Question and Answers

1. Address space.

Ans :

Address space is the amount of memory allocated for all possible addresses for a computa-tional entity for example, a device, a file, a server or a networked computer. The system provides each device and process address space that holds a specific portion of the processor's address space. This can include either physical or virtual addresses accessible to a processor or reserved for a particular process.

The width of its address bus and registers often restricts the processor's address space. However, a memory management technique called virtual memory can increase the size of the address space to be higher than that of the physical memory.

2. IPv4.

Ans :

Slash notation is a compact way to show or write an IPv4 subnet mask. When you use slash notation, you write the IP address, a forward slash (/), and the subnet mask number.

To find the subnet mask number:

1. Convert the decimal representation of the subnet mask to a binary representation.
2. Count each "1" in the subnet mask. The total is the subnet mask number.

3. Write about classless addressing.

Ans :

The address depletion issue was not fully resolved by classful addressing's subnetting and supernetting techniques. As the Internet expanded, it became obvious that a bigger address space was required as a long-term fix. However, the expanded address space necessitates that IP addresses should be longer as well, necessitating a change in IP packet syntax. The short-term solution, which uses the same address space but modifies the distribution of addresses to deliver a fair amount to each business, was developed despite the fact that the long-term solution, known as IPv6, has already been developed. Classless addressing is the temporary fix, which nevertheless makes use of IPv4 addresses. In order to make up for address depletion, the class privilege was taken out of the distribution.

The entire address space is partitioned into blocks of varying lengths with classless addressing. An address's prefix designates the block (network); its suffix designates the node (device). We are capable of having a block of 20, 21, 22 ,..., 232 addresses, theoretically. One of the limitations is that a block of addresses must have a power of two addresses. One address block may be given to an organization. The given figure demonstrates the non-overlapping block segmentation of the entire address space.

4. What is Network Address Translation? Explain.

Ans :

NAT (Network Address Translation) connects two networks and maps the private (inside local) addresses into public addresses (inside global). Inside local denotes that the best address belonged to an internal network and was not assigned by a Network Information Centre or service power. The inside global signifies that the address is a valid address assigned by the NIC or service provider, and one or more inside local addresses to the outside world.

5. Advantages of NAT.

Ans :

The following are the advantages of NAT:

- NAT protects the public addresses that have been registered and slow down the IP address space exhaustion.
- Removes the address renumbering process that occurs when switching networks
- The occurrence of address overlap was significantly reduced.
- Increases flexibility of the connection establishment.

6. Datagram network.

Ans :

In data communications, we need to send messages from one end system to another. If the message is going to pass through a packet-switched network, it needs to be divided into packets of fixed or variable size. The size of the packet is determined by the network and the governing protocol.

In packet switching, there is no resource allocation for a packet. This means that there is no reserved bandwidth on the links, and there is no scheduled processing time for each packet. Resources are allocated on demand. The allocation is done on a first come, first-served basis. When a switch receives a packet, no matter what is the source or destination, the packet must wait if there are other packets being processed.

In a datagram network, each packet is treated independently of all others. Even if a packet is part of a multi packet transmission, the network treats it as though it existed alone. Packets in this approach are referred to as datagrams. Datagram switching is normally done at the network layer.

7. What is Fragmentation in Networking? Explain.

Ans :

IP fragmentation is a process that divides packets into smaller pieces (fragments) so that the resulting pieces can travel across a link with a smaller maximum transmission unit (MTU) than the original packet size. The receiving host reassembles the fragments.

8. What is Checksum? How to apply check-sum for error detection? Explain.

Ans :

The checksum is a network method to check for any error or damage to the data transmitted to the sender side from the sender side. The checksum method applies the bit addition and bit complement method to perform the checksum implementation.

The need to apply checksum or any other error-detection method is done simply to identify the damage to the data when it's being transmitted over the network channel.

The checksum uses the Checksum Generator on the sender side and the Checksum Checker on the receiver side.

9. IPV6.

Ans :

Before introducing IPv6 Address format, we shall look into Hexadecimal Number System. Hexadecimal is a positional number system that uses radix (base) of 16. To represent the values in readable format, this system uses 0-9 symbols to represent values from zero to nine and A-F to represent values from ten to fifteen. Every digit in Hexadecimal can represent values from 0 to 15.

10. ICMP Protocol.

Ans :

The ICMP stands for Internet Control Message Protocol. It is a network layer protocol. It is used for error handling in the network layer, and it is primarily used on network devices such as routers. As different types of errors can exist in the network layer, so ICMP can be used to report these errors and to debug those errors.

For example, some sender wants to send the message to some destination, but the router couldn't send the message to the destination. In this case, the router sends the message to the sender that I could not send the message to that destination.

The IP protocol does not have any error-reporting or error-correcting mechanism, so it uses a message to convey the information. For example, if someone sends the message to the destination, the message is somehow stolen between the sender and the destination. If no one reports the error, then the sender might think that the message has reached the destination. If someone in-between reports the error, then the sender will resend the message very quickly.

11. IGMP.

Ans :

IGMP is an abbreviated form of Internet Group Management Protocol(IGMP). Mainly the Internet Protocol can be involved in the two types of communication i.e, Unicasting and multicasting. IGMP is one of the necessary but not the efficient protocol that is involved in Multicasting.

IGMP is basically a companion of Internet Protocol (IP).

IGMP is not a multicasting routing protocol but it is a protocol that manages the group membership. This protocol mainly helps the multicast routers in order to create and update a list of loyal members that are related to each router interface.

This protocol is used in streaming videos, gaming, or web conferencing tools.

12. What is ARP.

Ans :

Address Resolution Protocol (ARP) is an important protocol of the network layer in the OSI model, which helps find the MAC (Media Access Control) address given the system's IP address. The ARP's main task is to convert the 32-bit IP address (for IPv4) to a 48-bit MAC address.

This protocol is mostly used to determine the hardware (MAC) address of a device from an IP address. It is also used when one device wants to communicate with some other device on a local network. The full form of ARP is Address Resolution Protocol.

13. Reverse Address Resolution Protocol.

Ans :

RARP is abbreviation of Reverse Address Resolution Protocol which is a protocol based on computer networking which is employed by a client computer to request its IP address from a gateway server's Address Resolution Protocol table or cache. The network administrator creates a table in gateway-router, which is used to map the MAC address to corresponding IP address.

This protocol is used to communicate data between two points in a server. The client doesn't necessarily need prior knowledge the server identities capable of serving its request. Media Access Control (MAC) addresses requires individual configuration on the servers done by an administrator. RARP limits to the serving of IP addresses only.

When a replacement machine is set up, the machine may or might not have an attached disk that may permanently store the IP Address so the RARP client program requests IP Address from the RARP server on the router.

Choose the Correct Answers

1. Which of the following does not have a Net ID and Host ID? [d]
   (a) Class A (b) Class B
   (c) Class C (d) Class D

2. The slash notation in classless addressing is referred to as – [c]
   (a) NIFT (b) PITF
   (c) CIDR (d) TRS

3. Which Class is reserved for future use? [d]
   (a) A (b) B
   (c) D (d) E

4. The maximum number of networks that can use Class C addresses in the IPv4 addressing format is _____. [c]
   (a) 2^{14} (b) 2^{7}
   (c) 2^{21} (d) 2^{24}

5. In IPv4, what is needed to determine the number of the last byte of a fragment? [d]
   (a) Identification number (b) Offset number
   (c) Total length (d) ((b) and ((c) D

6. Which of the following is a necessary part of IPv6 diagram [a]
   (a) Base header (b) Extension header
   (c) Data packet from the upper layer (d) ((a) and ((c)

7. Among the following which protocol can be used to report these errors and to debug those errors. [d]
   (a) ARP (b) RARP ICMP
   (c) IGMP (d) ICMP

8. Which type of ARP request helps the network to identify the duplicate IP address [b]
   (a) Proxy ARP (b) Gratuitous ARP
   (c) Reverse ARP (d) Inverse ARP

9. Among the following which protocol is used for dynamic multicasting [c]
   (a) ARP (b) RARP ICMP
   (c) IGMP (d) ICMP

10. _____ is an important protocol of the network layer in the OSI model, which helps find the MAC (Media Access Control) address given the system's IP address. [c]
    (a) ARP (b) RARP ICMP
    (c) IGMP (d) ICMP

Fill in the Blanks

1. _____ connects two networks and maps the private (inside local) addresses into public addresses (inside global).
2. _____ is a process that divides packets into smaller pieces so that the resulting pieces can travel across a link with a smaller maximum transmission unit (MTU) than the original packet size
3. In a _____ network, each packet is treated independently of all others. Even if a packet is part of a multi packet transmission.
4. The header of IPv6 is of fix length of _____ bytes
5. _____ is a tool that tracks the route taken by a packet on an IP network from source to destination.
6. Association between a protocol address and a hardware address is known as _____
7. RARP full form _____
8. In _____ queuing method, the first packet that arrives at a router is the first packet to be transmitted
9. So in a _____ algorithm in which gateways maintain separate queues for packets from each individual source and serviced them in a round-robin manner.
10. TCP reacts to _____ by reducing the sender window size.

Answers

1. NAT (Network Address Translation)
2. Fragmentation
3. Datagram
4. 40
5. Traceroute
6. Binding.
7. Reverse Address Resolution Protocol
8. FIFO
9. Fair queuing
10. Ccongestion