UNIT III - NETWORK LAYER

Network Layer Services – Packet Switching – Performance – IPV4 Addresses – Forwarding of IP Packets – Network Layer Protocols: IP, ICMP v4 – Unicast Routing Algorithms – Protocols – Multicasting Basics – IPV6 Addressing – IPV6 Protocol

1. NETWORK LAYER SERVICES

- The network layer in the TCP/IP protocol suite is responsible for the host-to-host delivery of datagrams.
- It provides services to the transport layer and receives services from the data-link layer.
- The network layer translates the logical addresses into physical addresses
- It determines the route from the source to the destination and also manages the traffic problems such as switching, routing and controls the congestion of data packets.
- The main role of the network layer is to move the packets from sending host to the receiving host.

Services provided by network layer are

PACKETIZING

- The first duty of the network layer is definitely packetizing.
- This means encapsulating the payload (data received from upper layer) in a network-layer packet at the source and decapsulating the payload from the network-layer packet at the destination.
- The network layer is responsible for delivery of packets from a sender to a receiver without changing or using the contents.

ROUTING AND FORWARDING

Routing

- The network layer is responsible for routing the packet from its source to the destination.
- The network layer is responsible for finding the best one among these possible routes.
- The network layer needs to have some specific strategies for defining the best route.
- Routing is the concept of applying strategies and running routing protocols to create the decision-making tables for each router.
- These tables are called as routing tables.

Forwarding

- Forwarding can be defined as the action applied by each router when a packet arrives at one of its interfaces.
- The decision-making table, a router normally uses for applying this action is called the forwarding table.
- When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network.

ERROR CONTROL

- The network layer in the Internet does not directly provide error control.
- It adds a checksum field to the datagram to control any corruption in the header, but not in the whole datagram.
- This checksum prevents any changes or corruptions in the header of the datagram.
- The Internet uses an auxiliary protocol called ICMP, that provides some kind of error control if the datagram is discarded or has some unknown information in the header.

FLOW CONTROL

- Flow control regulates the amount of data a source can send without overwhelming the receiver.
- The network layer in the Internet, however, does not directly provide any flow control.
- The datagrams are sent by the sender when they are ready, without any attention to the readiness of the receiver.
- Flow control is provided for most of the upper-layer protocols that use the services of the network layer, so another level of flow control makes the network layer more complicated and the whole system less efficient.

CONGESTION CONTROL

- Another issue in a network-layer protocol is congestion control.
- Congestion in the network layer is a situation in which too many datagrams are present in an area of the Internet.
- Congestion may occur if the number of datagrams sent by source computers is beyond the capacity of the network or routers.
- In this situation, some routers may drop some of the datagrams.

SECURITY

- Another issue related to communication at the network layer is security.
- To provide security for a connectionless network layer, we need to have another virtual level that changes the connectionless service to a connection-oriented service. This virtual layer is called as called IPSec (IP Security).

2. PACKET SWITCHING

(REFER THE TOPIC PACKET SWITCHING FROM UNIT – I)

3. NETWORK-LAYER PERFORMANCE

- The performance of a network can be measured in terms of
  Delay, Throughput and Packet loss.
- Congestion control is an issue that can improve the performance.

DELAY

- A packet from its source to its destination, encounters delays.
- The delays in a network can be divided into four types:
  Transmission delay, Propagation delay, Processing delay and Queuing delay.

Transmission Delay

- A source host or a router cannot send a packet instantaneously.
- A sender needs to put the bits in a packet on the line one by one.
- If the first bit of the packet is put on the line at time t_1 and the last bit is put on the line at time t_2, transmission delay of the packet is $(t_2 - t_1)$.
- The transmission delay is longer for a longer packet and shorter if the sender can transmit faster.
- The Transmission delay is calculated using the formula

$$\mathbf{Delay_{tr} = (Packet\ length) / (Transmission\ rate)}$$

- ***Example*** :
  In a Fast Ethernet LAN with the transmission rate of 100 million bits per second and a packet of 10,000 bits, it takes (10,000)/(100,000,000) or 100 microseconds for all bits of the packet to be put on the line.

Propagation Delay

- Propagation delay is the time it takes for a bit to travel from point A to point B in the transmission media.
- The propagation delay for a packet-switched network depends on the propagation delay of each network (LAN or WAN).
- The propagation delay depends on the propagation speed of the media, which is $3X10^8$ meters/second in a vacuum and normally much less in a wired medium.
- It also depends on the distance of the link.
- The Propagation delay is calculated using the formula

$$\mathbf{Delay_{pg} = (Distance) / (Propagation\ speed)}$$

- ***Example***
  If the distance of a cable link in a point-to-point WAN is 2000 meters and the propagation speed of the bits in the cable is $2\ X\square 10^8$ meters/second, then the propagation delay is 10 microseconds.

Processing Delay

- The processing delay is the time required for a router or a destination host to receive a packet from its input port, remove the header, perform an error detection procedure, and deliver the packet to the output port (in the case of a router) or deliver the packet to the upper-layer protocol (in the case of the destination host).

- The processing delay may be different for each packet, but normally is calculated as an average.

Delay_{pr} = Time required to process a packet in a router or a destination host

Queuing Delay

- Queuing delay can normally happen in a router.
- A router has an input queue connected to each of its input ports to store packets waiting to be processed.
- The router also has an output queue connected to each of its output ports to store packets waiting to be transmitted.
- The queuing delay for a packet in a router is measured as the time a packet waits in the input queue and output queue of a router.

Delay_{qu} = The time a packet waits in input and output queues in a router

Total Delay

- Assuming equal delays for the sender, routers and receiver, the total delay (source-to-destination delay) of a packet can be calculated if we know the number of routers, n, in the whole path.

$$\textbf{Total delay} = (n + 1) (\text{Delay}_{tr} + \text{Delay}_{pg} + \text{Delay}_{pr}) + (n) (\text{Delay}_{qu})$$

- If we have n routers, we have (n +1) links.
- Therefore, we have (n +1) transmission delays related to n routers and the source, (n +1) propagation delays related to (n +1) links, (n +1) processing delays related to n routers and the destination, and only n queuing delays related to n routers.

THROUGHPUT

- Throughput at any point in a network is defined as the number of bits passing through the point in a second, which is actually the transmission rate of data at that point.
- In a path from source to destination, a packet may pass through several links (networks), each with a different transmission rate.
- Throughput is calculated using the formula

$$\textbf{Throughput} = \textbf{minimum}\{TR_1 , TR_2, \ldots TR_n\}$$

- ***Example:***

  Let us assume that we have three links, each with a different transmission rate.

  The data can flow at the rate of 200 kbps in Link1, 100 kbps in Link2 and 150kbps in Link3.

  Throughput = minimum{200,100,150} = 100.

PACKET LOSS

- Another issue that severely affects the performance of communication is the number of packets lost during transmission.

- When a router receives a packet while processing another packet, the received packet needs to be stored in the input buffer waiting for its turn.
- A router has an input buffer with a limited size.
- A time may come when the buffer is full and the next packet needs to be dropped.
- The effect of packet loss on the Internet network layer is that the packet needs to be resent, which in turn may create overflow and cause more packet loss.

CONGESTION CONTROL

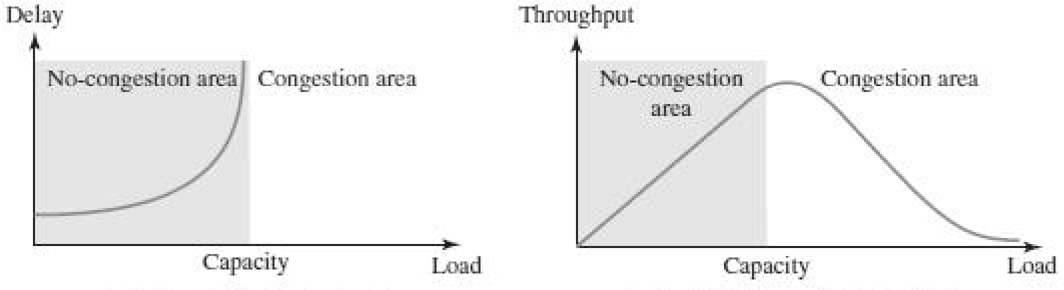
- Congestion at the network layer is related to two issues, throughput and delay.

Based on Delay

- When the load is much less than the capacity of the network, the delay is at a minimum.
- This minimum delay is composed of propagation delay and processing delay, both of which are negligible.
- However, when the load reaches the network capacity, the delay increases sharply because we now need to add the queuing delay to the total delay.
- The delay becomes infinite when the load is greater than the capacity.

Based on Throughout

- When the load is below the capacity of the network, the throughput increases proportionally with the load.
- We expect the throughput to remain constant after the load reaches the capacity, but instead the throughput declines sharply.
- The reason is the discarding of packets by the routers.
- When the load exceeds the capacity, the queues become full and the routers have to discard some packets.
- Discarding packets does not reduce the number of packets in the network because the sources retransmit the packets, using time-out mechanisms, when the packets do not reach the destinations.



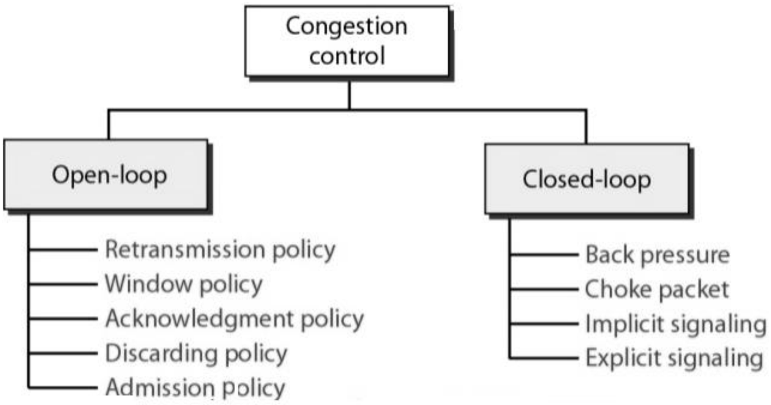
a. Delay as a function of load

b. Throughput as a function of load

Congestion Control Mechanisms

- Congestion control is a mechanism for improving performance.
- It refers to techniques and mechanisms that can either prevent congestion before it happens or remove congestion after it has happened.
- In general, we can divide congestion control mechanisms into two broad categories:

- **Open-loop Congestion control** (prevention)
- **Closed-loop Congestion control** (removal)



OPEN-LOOP CONGESTION CONTROL

- In open-loop congestion control, policies are applied to prevent congestion before it happens.
- In these mechanisms, congestion control is handled by either the source or the destination.

Retransmission Policy

- Retransmission is sometimes unavoidable.
- If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted.
- Retransmission in general may increase congestion in the network.
- However, a good retransmission policy can prevent congestion.
- The retransmission policy and the retransmission timers must be designed to optimize efficiency and at the same time prevent congestion.

Window Policy

- The type of window at the sender may also affect congestion.
- The Selective Repeat window is better than the Go-Back-N window for congestion control.
- In the Go-Back-N window, when the timer for a packet times out, several packets may be resent, although some may have arrived safe and sound at the receiver.
- This duplication may make the congestion worse.
- The Selective Repeat window, on the other hand, tries to send the specific packets that have been lost or corrupted.

Acknowledgment Policy

- The acknowledgment policy imposed by the receiver may also affect congestion.
- If the receiver does not acknowledge every packet it receives, it may slow down the sender and help prevent congestion.
- Several approaches are used in this case.
- A receiver may send an acknowledgment only if it has a packet to be sent or a special timer expires.

- A receiver may decide to acknowledge only N packets at a time.
- Sending fewer acknowledgments means imposing less load on the network.

Discarding Policy

- A good discarding policy by the routers may prevent congestion and at the same time may not harm the integrity of the transmission.
- For example, in audio transmission, if the policy is to discard less sensitive packets when congestion is likely to happen, the quality of sound is still preserved and congestion is prevented or alleviated.

Admission Policy

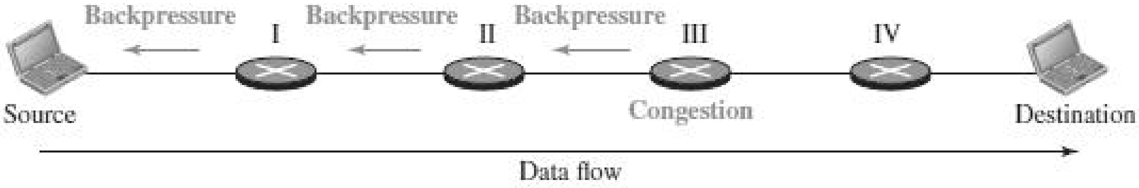
- An admission policy, which is a quality-of-service mechanism can also prevent congestion in virtual-circuit networks.
- Switches in a flow first check the resource requirement of a flow before admitting it to the network.
- A router can deny establishing a virtual-circuit connection if there is congestion in the network or if there is a possibility of future congestion.

CLOSED-LOOP CONGESTION CONTROL

- Closed-loop congestion control mechanisms try to alleviate congestion after it happens.
- Several mechanisms have been used by different protocols.

Backpressure

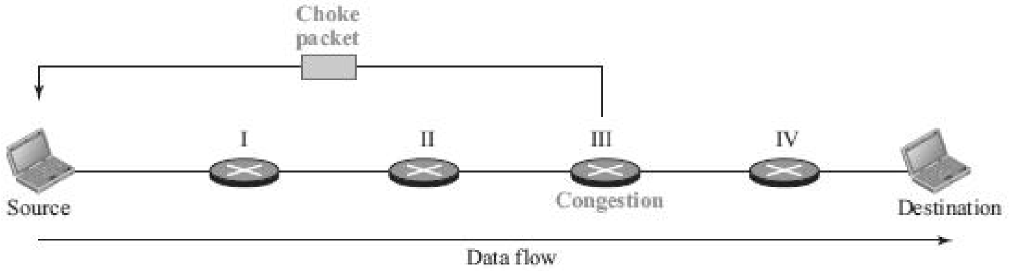
- The technique of backpressure refers to a congestion control mechanism in which a congested node stops receiving data from the immediate upstream node or nodes.
- This may cause the upstream node or nodes to become congested, and they, in turn, reject data from their upstream node or nodes, and so on.
- Backpressure is a node-to- node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source.
- The backpressure technique can be applied only to virtual circuit networks, in which each node knows the upstream node from which a flow of data is coming.



Choke Packet

- A choke packet is a packet sent by a node to the source to inform it of congestion.
- In backpressure, the warning is from one node to its upstream node, although the warning may eventually reach the source station.

- In the choke-packet method, the warning is from the router, which has encountered congestion, directly to the source station.
- The intermediate nodes through which the packet has traveled are not warned.
- The warning message goes directly to the source station; the intermediate routers do not take any action.



Implicit Signaling

- In implicit signaling, there is no communication between the congested node or nodes and the source.
- The source guesses that there is congestion somewhere in the network from other symptoms.
- For example, when a source sends several packets and there is no acknowledgment for a while, one assumption is that the network is congested.
- The delay in receiving an acknowledgment is interpreted as congestion in the network; the source should slow down.

Explicit Signaling

- The node that experiences congestion can explicitly send a signal to the source or destination.
- The explicit-signaling method is different from the choke-packet method.
- In the choke-packet method, a separate packet is used for this purpose; in the explicit-signaling method, the signal is included in the packets that carry data.
- Explicit signaling can occur in either the forward or the backward direction.

4. IPV4 ADDRESSES

- The identifier used in the IP layer of the TCP/IP protocol suite to identify the connection of each device to the Internet is called the Internet address or IP address.
- Internet Protocol version 4 (IPv4) is the fourth version in the development of the Internet Protocol (IP) and the first version of the protocol to be widely deployed.
- IPv4 is described in IETF publication in September 1981.

- The IP address is the address of the connection, not the host or the router. An IPv4 address is a 32-bit address that uniquely and universally defines the connection .
- If the device is moved to another network, the IP address may be changed.
- IPv4 addresses are unique in the sense that each address defines one, and only one, connection to the Internet.
- If a device has two connections to the Internet, via two networks, it has two IPv4 addresses.
- Pv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.

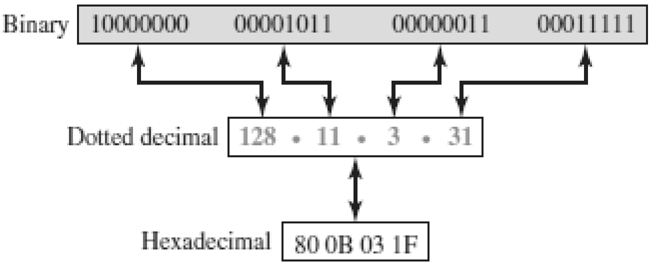
IPV4 ADDRESS SPACE

- IPv4 defines addresses has an address space.
- An address space is the total number of addresses used by the protocol.
- If a protocol uses *b* bits to define an address, the address space is 2^b because each bit can have two different values (0 or 1).
- IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than four billion).
- 4 billion devices could be connected to the Internet.

IPV4 ADDRESS NOTATION

There are three common notations to show an IPv4 address:

(i) binary notation (base 2), (ii) dotted-decimal notation (base 256), and
(ii) hexadecimal notation (base 16).



In *binary notation,* an IPv4 address is displayed as 32 bits. To make the address more readable, one or more spaces are usually inserted between bytes (8 bits).

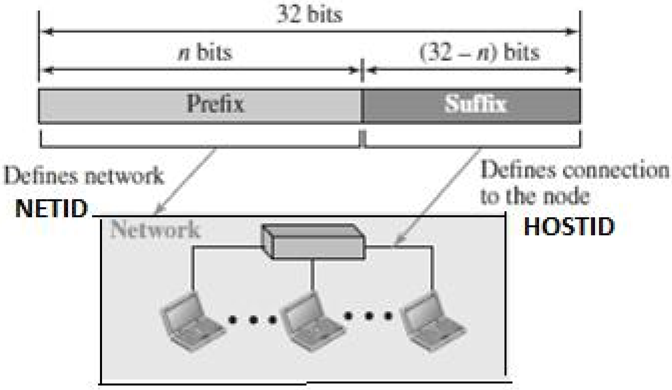
In *dotted-decimal notation,IPv4 addresses are* usually written in decimal form with a decimal point (dot) separating the bytes. Each number in the dotted-decimal notation is between 0 and 255.

In hexadecimal notation, each hexadecimal digit is equivalent to four bits. This means that a 32-bit address has 8 hexadecimal digits. This notation is often used in network programming.

HIERARCHY IN IPV4 ADDRESSING

- In any communication network that involves delivery, the addressing system is hierarchical.
- A 32-bit IPv4 address is also hierarchical, but divided only into two parts.

- The first part of the address, called the *prefix*, defines the network(Net ID); the second part of the address, called the *suffix*, defines the node (Host ID).
- The prefix length is *n* bits and the suffix length is (32-□*n*) bits.



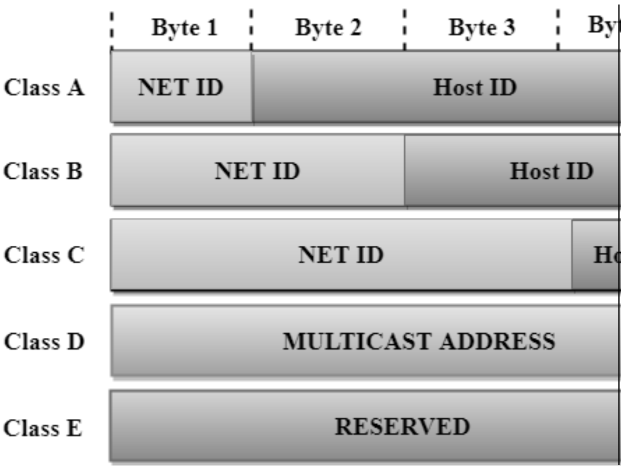
- A prefix can be fixed length or variable length.
- The network identifier in the IPv4 was first designed as a fixed-length prefix.
- This scheme is referred to as classful addressing.
- The new scheme, which is referred to as classless addressing, uses a variable-length network prefix.

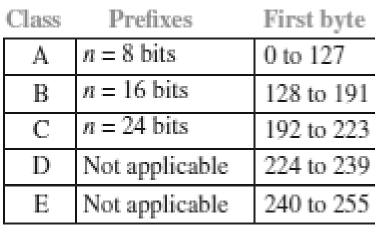
CATEGORIES OF IPV4 ADDRESSING

- There are two broad categories of IPv4 Addressing techniques.
- They are
  - Classful Addressing
  - Classless Addressing

CLASSFUL ADDRESSING

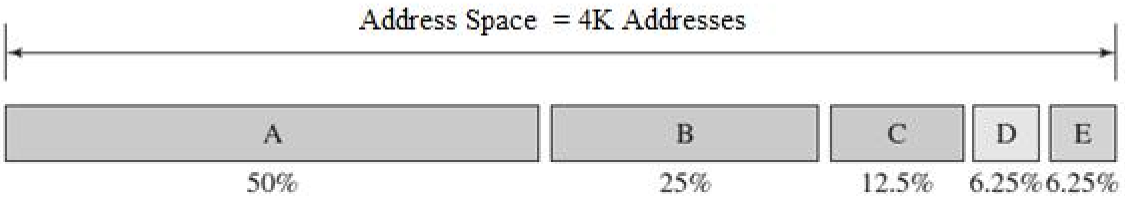
- An IPv4 address is 32-bit long(4 bytes).
- An IPv4 address is divided into sub-classes:

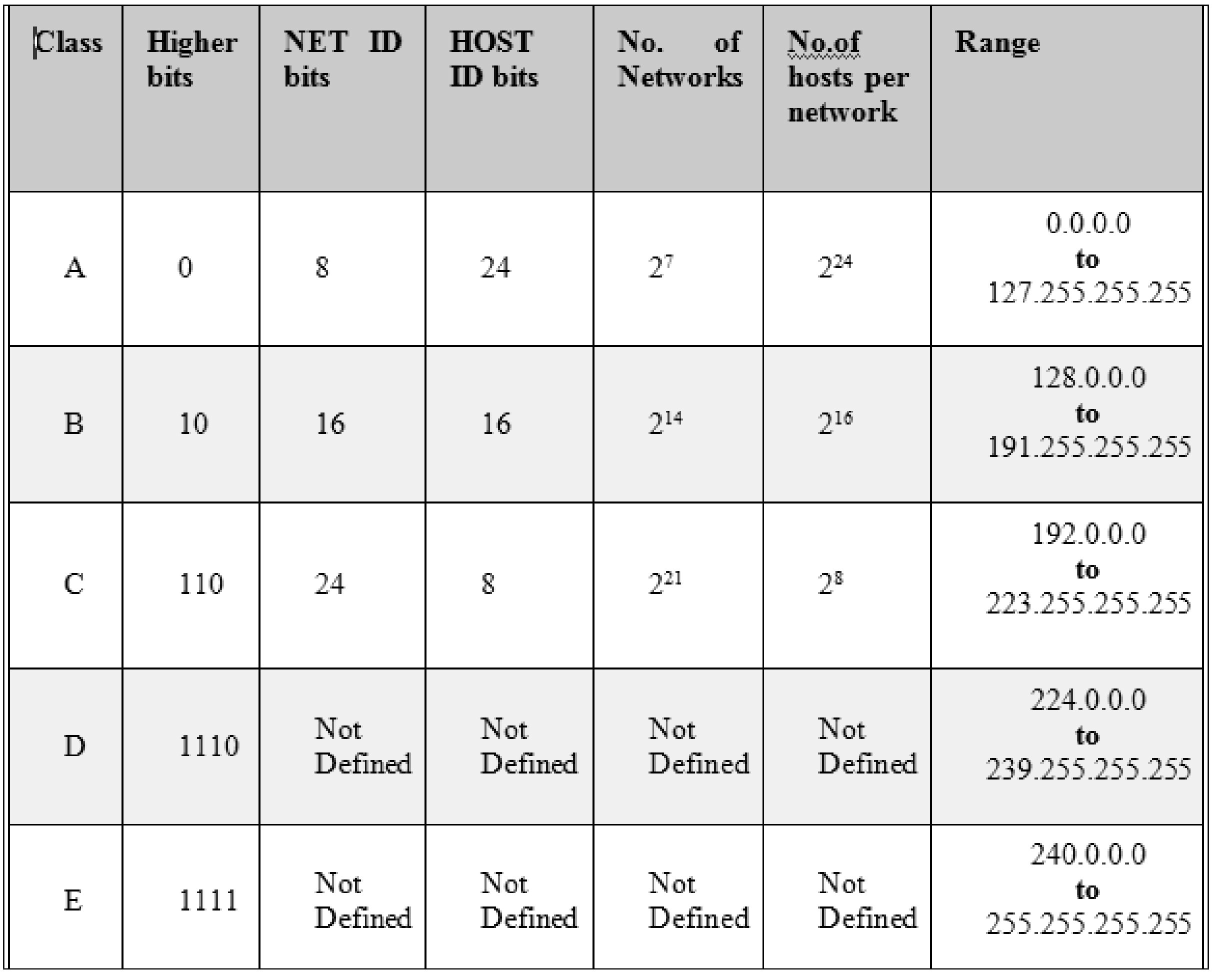




Class	Prefixes	First byte
A	n = 8 bits	0 to 127
B	n = 16 bits	128 to 191
C	n = 24 bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

Classful Network Architecture

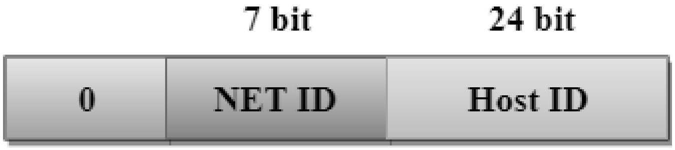




Class	Higher bits	NET ID bits	HOST ID bits	No. of Networks	No.of hosts per network	Range
A	0	8	24	2^7	2^{24}	0.0.0.0 **to** 127.255.255.255
B	10	16	16	2^{14}	2^{16}	128.0.0.0 **to** 191.255.255.255
C	110	24	8	2^{21}	2^8	192.0.0.0 **to** 223.255.255.255
D	1110	Not Defined	Not Defined	Not Defined	Not Defined	224.0.0.0 **to** 239.255.255.255
E	1111	Not Defined	Not Defined	Not Defined	Not Defined	240.0.0.0 **to** 255.255.255.255

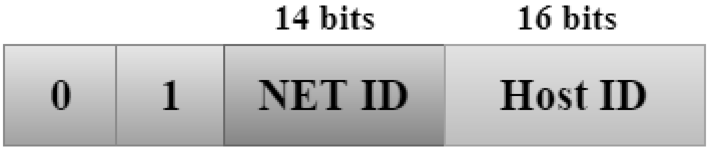
Class A

- In Class A, an IP address is assigned to those networks that contain a large number of hosts.
- The network ID is 8 bits long.
- The host ID is 24 bits long.
- In Class A, the first bit in higher order bits of the first octet is always set to 0 and the remaining 7 bits determine the network ID.
- The 24 bits determine the host ID in any network.
- The total number of networks in Class A = 2^7 = 128 network address
- The total number of hosts in Class A = 2^{24} - 2 = 16,777,214 host address



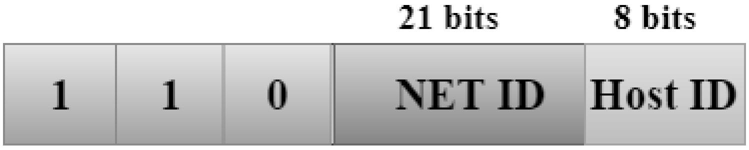
Class B

- In Class B, an IP address is assigned to those networks that range from small-sized to large-sized networks.
- The Network ID is 16 bits long.
- The Host ID is 16 bits long.
- In Class B, the higher order bits of the first octet is always set to 10, and the remaining14 bits determine the network ID.
- The other 16 bits determine the Host ID.
- The total number of networks in Class B = 2^{14} = 16384 network address
- The total number of hosts in Class B = 2^{16} - 2 = 65534 host address



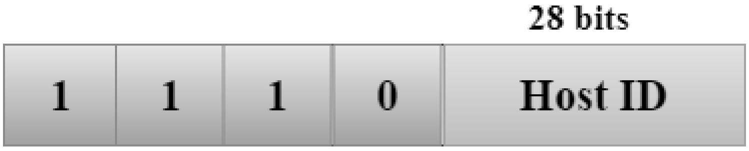
Class C

- In Class C, an IP address is assigned to only small-sized networks.
- The Network ID is 24 bits long.
- The host ID is 8 bits long.
- In Class C, the higher order bits of the first octet is always set to 110, and the remaining 21 bits determine the network ID.
- The 8 bits of the host ID determine the host in a network.
- The total number of networks = 2^{21} = 2097152 network address
- The total number of hosts = 2^{8} - 2 = 254 host address



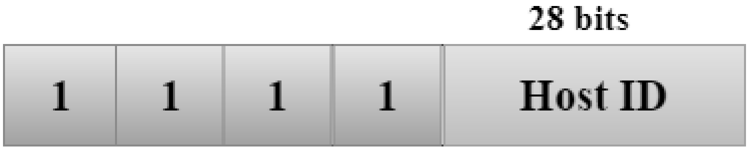
Class D

- In Class D, an IP address is reserved for multicast addresses.
- It does not possess subnetting.
- The higher order bits of the first octet is always set to 1110, and the remaining bits determines the host ID in any network.



Class E

- In Class E, an IP address is used for the future use or for the research and development purposes.
- It does not possess any subnetting.
- The higher order bits of the first octet is always set to 1111, and the remaining bits determines the host ID in any network.



Address Depletion in Classful Addressing

- The reason that classful addressing has become obsolete is address depletion.
- Since the addresses were not distributed properly, the Internet was faced with the problem of the addresses being rapidly used up.
- This results in no more addresses available for organizations and individuals that needed to be connected to the Internet.
- To understand the problem, let us think about class A.
- This class can be assigned to only 128 organizations in the world, but each organization needs to have a single network with 16,777,216 nodes .
- Since there may be only a few organizations that are this large, most of the addresses in this class were wasted (unused).
- Class B addresses were designed for midsize organizations, but many of the addresses in this class also remained unused.
- Class C addresses have a completely different flaw in design. The number of addresses that can be used in each network (256) was so small that most companies were not comfortable using a block in this address class.
- Class E addresses were almost never used, wasting the whole class.

Advantage of Classful Addressing

- Although classful addressing had several problems and became obsolete, it had one advantage.
- Given an address, we can easily find the class of the address and, since the prefix length for each class is fixed, we can find the prefix length immediately.
- In other words, the prefix length in classful addressing is inherent in the address; no extra information is needed to extract the prefix and the suffix.

Subnetting and Supernetting

- To alleviate address depletion, two strategies were proposed and implemented:
  (i) Subnetting and (ii) Supernetting.

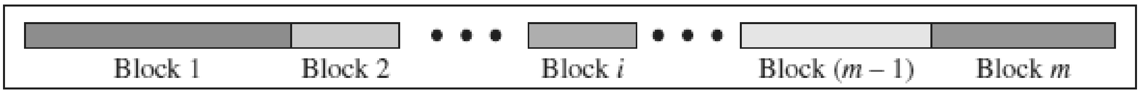
Subnetting

- In subnetting, a class A or class B block is divided into several subnets.
- Each subnet has a larger prefix length than the original network.
- For example, if a network in class A is divided into four subnets, each subnet has a prefix of $n_{sub} = 10$.
- At the same time, if all of the addresses in a network are not used, subnetting allows the addresses to be divided among several organizations.

CLASSLESS ADDRESSING

- In 1996, the Internet authorities announced a new architecture called **classless addressing.**
- In classless addressing, variable-length blocks are used that belong to no classes.
- We can have a block of 1 address, 2 addresses, 4 addresses, 128 addresses, and so on.

- In classless addressing, the whole address space is divided into variable length blocks.
- The prefix in an address defines the block (network); the suffix defines the node (device).
- Theoretically, we can have a block of 2^0, 2^1, 2^2, □□□□□□□□2^{32} addresses.
- The number of addresses in a block needs to be a power of 2. An organization can be granted one block of addresses.

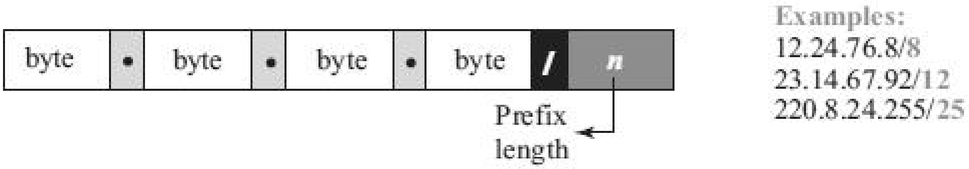


Address space

- The prefix length in classless addressing is variable.
- We can have a prefix length that ranges from 0 to 32.
- The size of the network is inversely proportional to the length of the prefix.
- A small prefix means a larger network; a large prefix means a smaller network.
- The idea of classless addressing can be easily applied to classful addressing.
- An address in class A can be thought of as a classless address in which the prefix length is 8.
- An address in class B can be thought of as a classless address in which the prefix is 16, and so on. In other words, classful addressing is a special case of classless addressing.

Notation used in Classless Addressing

- The notation used in classless addressing is informally referred to as *slash notation* and formally as ***classless interdomain routing*** or ***CIDR.***



- For example , 192.168.100.14 **/24** represents the IP address 192.168.100.14 and, its subnet mask 255.255.255.0, which has 24 leading 1-bits.

Address Aggregation

- One of the advantages of the CIDR strategy is **address aggregation** (sometimes called *address summarization* or *route summarization*).
- When blocks of addresses are combined to create a larger block, routing can be done based on the prefix of the larger block.
- ICANN assigns a large block of addresses to an ISP.
- Each ISP in turn divides its assigned block into smaller subblocks and grants the subblocks to its customers.

Special Addresses in IPv4

- There are five special addresses that are used for special purposes:
  this-host address, *limited-broadcast*address, *loopback* address,
  private addresses, and *multicast* addresses.

This-host Address

- ✓ The only address in the block **0.0.0.0**/**32** is called the *this-host* address.
- ✓ It is used whenever a host needs to send an IP datagram but it does not know its own address to use as the source address.

Limited-broadcast Address

- ✓ The only address in the block **255.255.255.255**/**32** is called the *limited-broadcast* address.
- ✓ It is used whenever a router or a host needs to send a datagram to all devices in a network.
- ✓ The routers in the network, however, block the packet having this address as the destination;the packet cannot travel outside the network.

Loopback Address

- ✓ The block **127.0.0.0**/**8** is called the *loopback* address.
- ✓ A packet with one of the addresses in this block as the destination address never leaves the host; it will remain in the host.

Private Addresses

- ✓ Four blocks are assigned as private addresses: 10.0.0.0/**8**, 172.16.0.0/**12**, 192.168.0.0/**16**, and 169.254.0.0/**16**.

Multicast Addresses

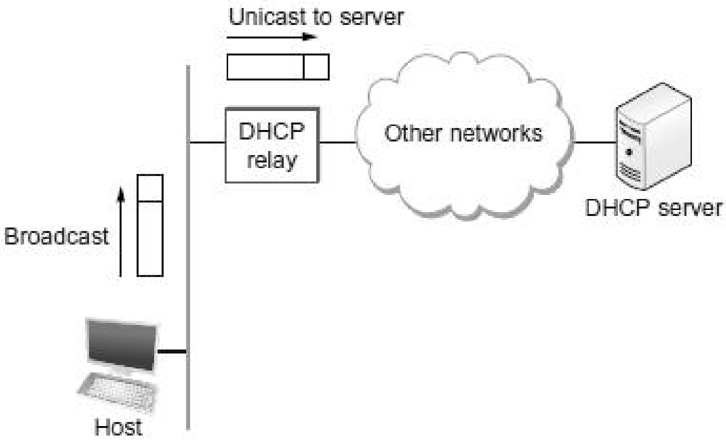
- ✓ The block 224.0.0.0/**4** is reserved for multicast addresses.

---

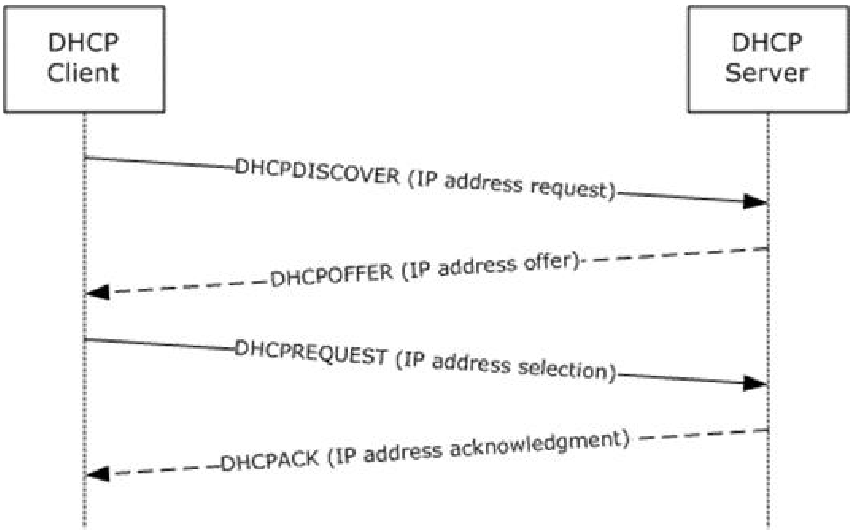
5. DHCP – DYNAMIC HOST CONFIGURATION PROTOCOL

- ➢ The dynamic host configuration protocol is used to simplify the installation and maintenance of networked computers.
- ➢ DHCP is derived from an earlier protocol called BOOTP.
- ➢ Ethernet addresses are configured into network by manufacturer and they are unique.
- ➢ IP addresses must be unique on a given internetwork but also must reflect the structure of the internetwork
- ➢ Most host Operating Systems provide a way to manually configure the IP information for the host
- ➢ **Drawbacks of manual configuration :**
    1. A lot of work to configure all the hosts in a large network
    2. Configuration process is error-prune
- ➢ It is necessary to ensure that every host gets the correct network number and that no two hosts receive the same IP address.
- ➢ For these reasons, automated configuration methods are required.
- ➢ The primary method uses a protocol known as the *Dynamic Host Configuration Protocol* (DHCP).
- ➢ The main goal of DHCP is to minimize the amount of manual configuration required for a host.

- If a new computer is connected to a network, DHCP can provide it with all the necessary information for full system integration into the network.
- DHCP is based on a client/server model.
- DHCP clients send a request to a DHCP server to which the server responds with an IP address
- DHCP server is responsible for providing configuration information to hosts.
- There is at least one DHCP server for an administrative domain.
- The DHCP server can function just as a centralized repository for host configuration information.
- The DHCP server maintains a pool of available addresses that it hands out to hosts on demand.

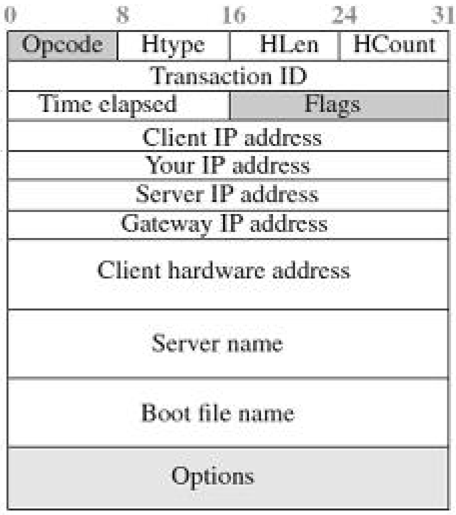


- A newly booted or attached host sends a DHCPDISCOVER message to a special IP address (255.255.255.255., which is an IP broadcast address.
- This means it will be received by all hosts and routers on that network.
- DHCP uses the concept of a *relay agent.* There is at least one relay agent on each network.
- DHCP relay agent is configured with the IP address of the DHCP server.
- When a relay agent receives a DHCPDISCOVER message, it unicasts it to the DHCP server and awaits the response, which it will then send back to the requesting client.


DHCP Client
DHCP Server
DHCPDISCOVER (IP address request)
DHCPOFFER (IP address offer)
DHCPREQUEST (IP address selection)
DHCPACK (IP address acknowledgment)

DHCP Message Format

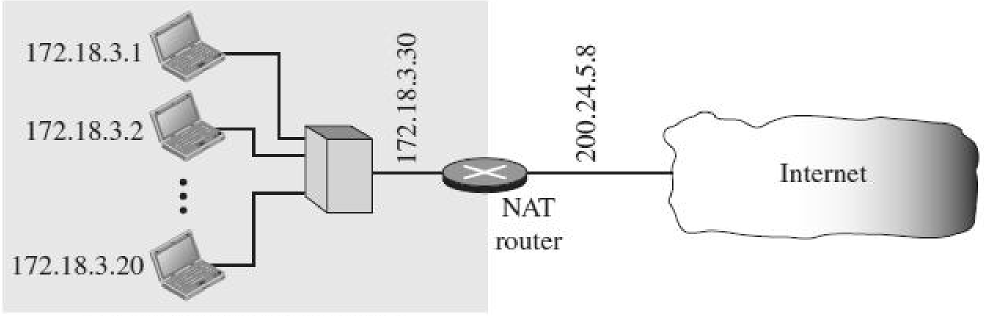
- A DHCP packet is actually sent using a protocol called the *User Datagram Protocol* (UDP).



Opcode: Operation code, request (1) or reply (2)
Htype: Hardware type (Ethernet, ...)
HLen: Length of hardware address
HCount: Maximum number of hops the packet can travel
Transaction ID: An integer set by the client and repeated by the server
Time elapsed: The number of seconds since the client started to boot
Flags: First bit defines unicast (0) or multicast (1); other 15 bits not used
Client IP address: Set to 0 if the client does not know it
Your IP address: The client IP address sent by the server
Server IP address: A broadcast IP address if client does not know it
Gateway IP address: The address of default router
Server name: A 64-byte domain name of the server
Boot file name: A 128-byte file name holding extra information
Options: A 64-byte field with dual purpose described in text

6. NETWORK ADDRESS TRANSLATION (NAT)

- A technology that can provide the mapping between the private and universal (external)addresses, and at the same time support virtual private networks is called as **Network Address Translation (NAT).**
- The technology allows a site to use a set of private addresses for internal communication and a set of global Internet addresses (at least one) for communication with the rest of the world.
- The site must have only one connection to the global Internet through a NAT-capable router that runs NAT software.



Site using private addresses

- The private network uses private addresses.
- The router that connects the network to the global address uses one private address and one global address.
- The private network is invisible to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8.

Types of NAT

- Two types of NAT exists .
  - (a) One-to-one translation of IP addresses
  - (b) One-to-many translation of IP addresses

Address Translation

- All of the outgoing packets go through the NAT router, which replaces the source address in the packet with the global NAT address.
- All incoming packets also pass through the NAT router, which replaces the destination address in the packet (the NAT router global address) with the appropriate private address.

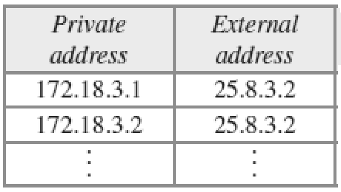
Translation Table

- There may be tens or hundreds of private IP addresses, each belonging to one specific host.
- The problem arises when we want to translate the source address to an external address. This is solved if the NAT router has a translation table.

Translation table with two columns

- ✓ A translation table has only two columns: the private address and the external address (destination address of the packet).
- ✓ When the router translates the source address of the outgoing packet, it also makes note of the destination address—where the packet is going.
- ✓ When the response comes back from the destination, the router uses the source address of the packet (as the external address) to find the private address of the packet.

Two-column translation table

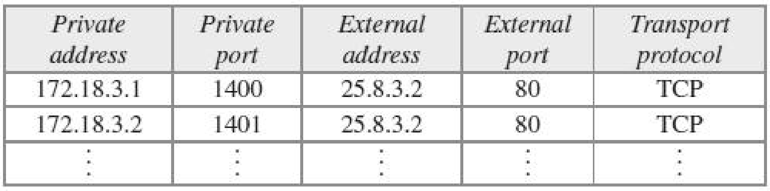


Private address	*External address*
172.18.3.1	25.8.3.2
172.18.3.2	25.8.3.2
⋮	⋮

Translation table with five columns

- ✓ To allow a many-to-many relationship between private-network hosts and external server programs, we need more information in the translation table.
- ✓ If the translation table has five columns, instead of two, that include the source and destination port addresses and the transport-layer protocol, the ambiguity is eliminated.

Five-column translation table



Private address	*Private port*	*External address*	*External port*	*Transport protocol*
172.18.3.1	1400	25.8.3.2	80	TCP
172.18.3.2	1401	25.8.3.2	80	TCP
⋮	⋮	⋮	⋮	⋮

7. FORWARDING OF IP PACKETS

- Forwarding means to deliver the packet to the next hop (which can be the final destination or the intermediate connecting device).
- Although IP protocol was originally designed as a connectionless protocol, today the tendency is to use IP as a connection-oriented protocol based on the label attached to an IP datagram .
- When ***IP is used as a connectionless protocol***, forwarding is based on the ***destination address*** of the IP datagram.
- When the ***IP is used as a connection-oriented protocol***, forwarding is based on the ***label*** attached to an IP datagram.

FORWARDING BASED ON DESTINATION ADDRESS

- This is a traditional approach.
- In this case, forwarding requires a host or a router to have a forwarding table.
- When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the next hop to deliver the packet to.

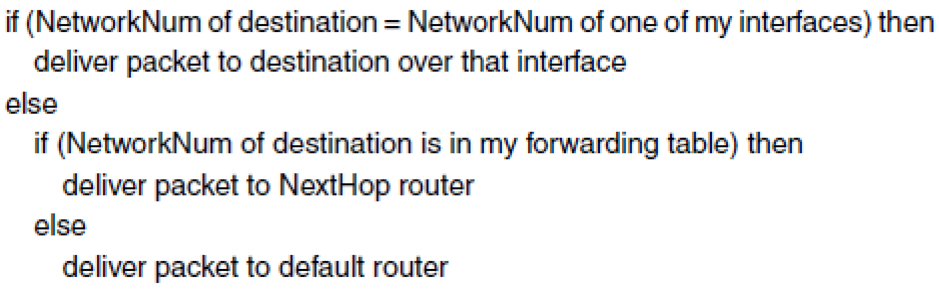
- The main points in forwarding of IP Packets(datagram) are the following:
  - Every IP Packets contains the IP address of the destination host.
  - The network part of an IP address uniquely identifies a single physical network that is part of the larger Internet.
  - All hosts and routers that share the same network part of their address are connected to the same physical network and can thus communicate with each other by sending frames over that network.
  - Every physical network that is part of the Internet has at least one router that, by definition, is also connected to at least one other physical network; this router can exchange packets with hosts or routers on either network.

- Forwarding IP Packets can therefore be handled in the following way.
  - A Packets is sent from a source host to a destination host, possibly passing through several routers along the way.
  - Any node, whether it is a host or a router, first tries to establish whether it is connected to the same physical network as the destination.

- To do this, it compares the network part of the destination address with the network part of the address of each of its network interfaces. (Hosts normally have only one interface, while routers normally have two or more, since they are typically connected to two or more networks.)
- If a match occurs, then that means that the destination lies on the same physical network as the interface, and the packet can be directly delivered over that network that has a reasonable chance of getting the packet closer to its destination.
- If there is no match, then the node is not connected to the same physical network as the destination node, then it needs to send the packet to a router.

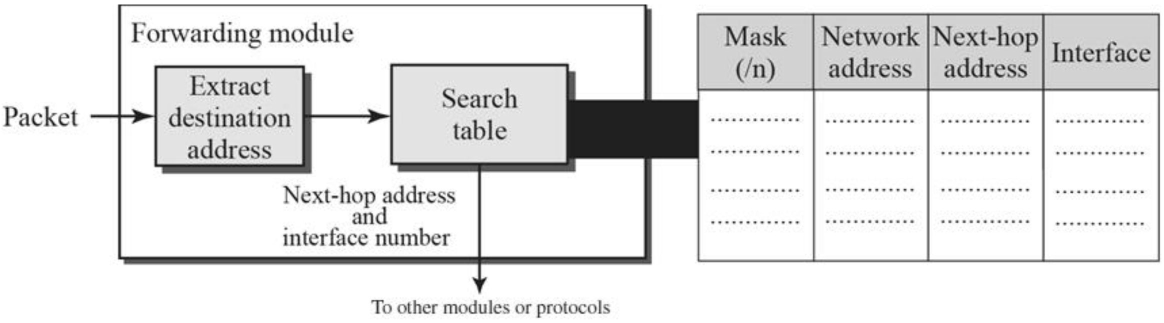
- In general, each node will have a choice of several routers, and so it needs to pick the best one, or at least one that has a reasonable chance of getting the datagram closer to its destination.
- The router that it chooses is known as the *next hop* router.
- The router finds the correct next hop by consulting its forwarding table. The forwarding table is conceptually just a list of (NetworkNum, NextHop) pairs.
- There is also a default router that is used if none of the entries in the table matches the destination's network number.
- All Packets destined for hosts not on the physical network to which the sending host is attached will be sent out through the default router.

Forwarding Algorithm



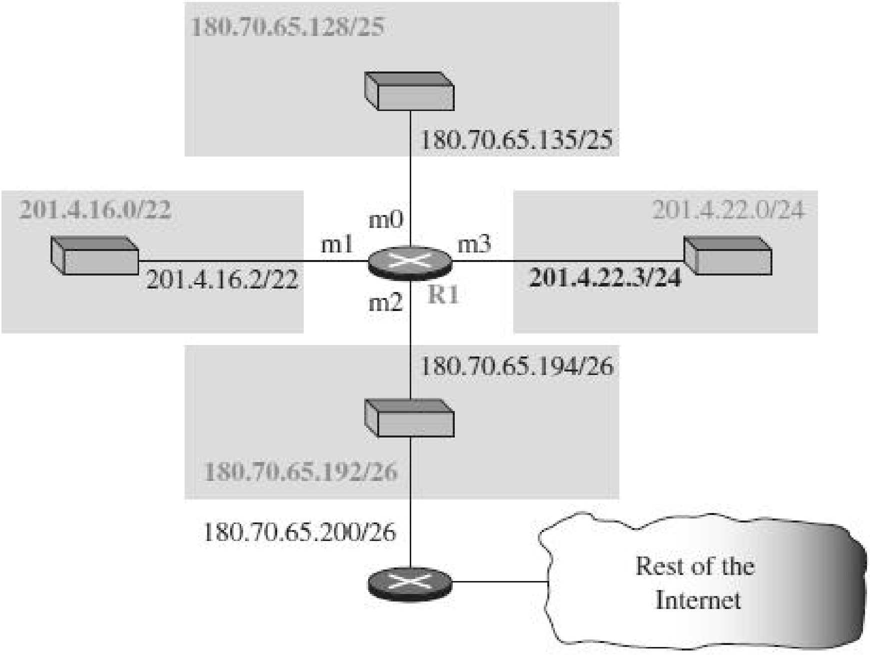
```
if (NetworkNum of destination = NetworkNum of one of my interfaces) then
  deliver packet to destination over that interface
else
  if (NetworkNum of destination is in my forwarding table) then
    deliver packet to NextHop router
  else
    deliver packet to default router
```

Simplified Forwarding Module


Forwarding module
Packet
Extract destination address
Search table
Next-hop address and interface number
To other modules or protocols
Mask (/n)
Network address
Next-hop address
Interface

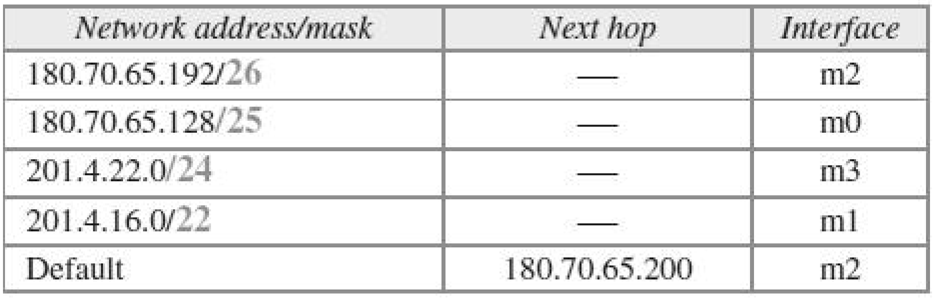
- The job of the forwarding module is to search the table, row by row.
- In each row, the *n* leftmost bits of the destination address (prefix) are kept and the rest of the bits (suffix) are set to 0s.
- If the resulting address (*network address*), matches with the address in the first column, the information in the next two columns is extracted; otherwise the search continues. Normally, the last row has a default value in the first column, which indicates all destination addresses that did not match the previous rows.
- Routing in classless addressing uses another principle, **longest mask matching.**
- This principle states that the forwarding table is sorted from the longest mask to the shortest mask.
- In other words, if there are three masks, /27, /26, and /24, the mask /27 must be the first entry and /24 must be the last.

Example



- Let us make a forwarding table for router R1 using the configuration as given in the figure above

Forwarding table for router R1



Network address/mask	*Next hop*	*Interface*
180.70.65.192/26	—	m2
180.70.65.128/25	—	m0
201.4.22.0/24	—	m3
201.4.16.0/22	—	m1
Default	180.70.65.200	m2

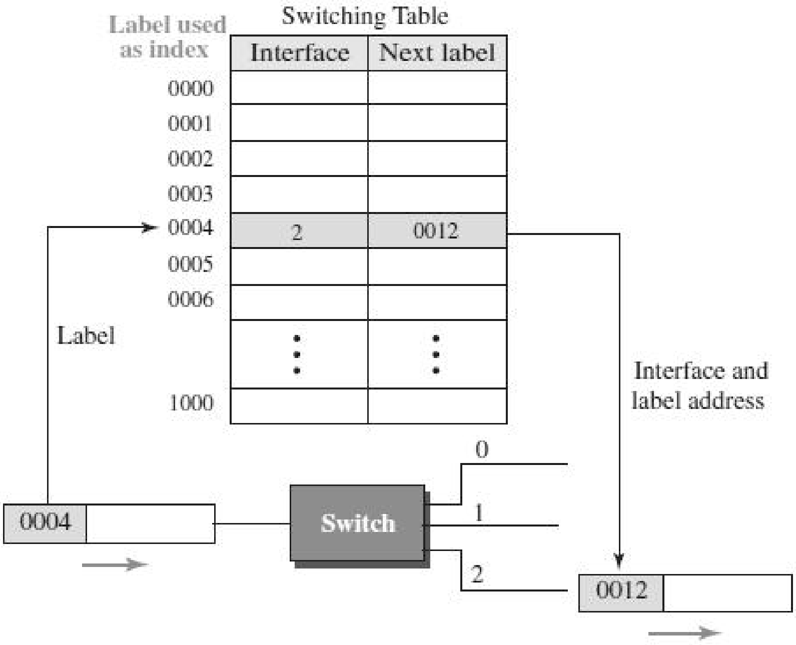
- When a packet arrives whose leftmost 26 bits in the destination address match the bits in the first row, the packet is sent out from interface m2.
- When a packet arrives whose leftmost 25 bits in the address match the bits in the second row, the packet is sent out from interface m0, and so on.
- The table clearly shows that the first row has the longest prefix and the fourth row has the shortest prefix.
- The longer prefix means a smaller range of addresses; the shorter prefix means a larger range of addresses.

FORWARDING BASED ON LABEL

- In a connection-oriented network (virtual-circuit approach), a switch forwards a packet based on the label attached to the packet.
- Routing is normally based on searching the contents of a table; switching can be done by accessing a table using an index.
- In other words, routing involves searching; switching involves accessing.

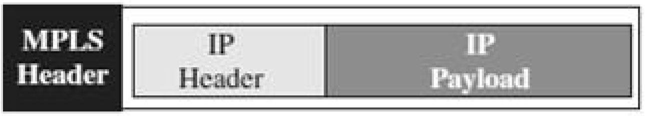
Example

- The Figure below shows a simple example of using a label to access a switching table.
- Since the labels are used as the index to the table, finding the information in the table is immediate.



Multi-Protocol Label Switching (MPLS)

- During the 1980s, several vendors created routers that implement switching technology.
- Later IETF approved a standard that is called Multi-Protocol Label Switching.
- In this standard, some conventional routers in the Internet can be replaced by MPLS routers, which can behave like a router and a switch.
- When behaving like a router, MPLS can forward the packet based on the destination address; when behaving like a switch, it can forward a packet based on the label.



8. NETWORK LAYER PROTOCOLS : IP, ICMPV4

- The main protocol Internet Protocol is responsible for packetizing, forwarding, and delivery of a packet at the network layer.
- The Internet Control Message Protocol version 4 (ICMPv4) helps IPv4 to handle some errors that may occur in the network-layer delivery.

IP - INTERNET PROTOCOL

- The Internet Protocol is the key tool used today to build scalable, heterogeneous internetworks.
- IP runs on all the nodes (both hosts and routers) in a collection of networks

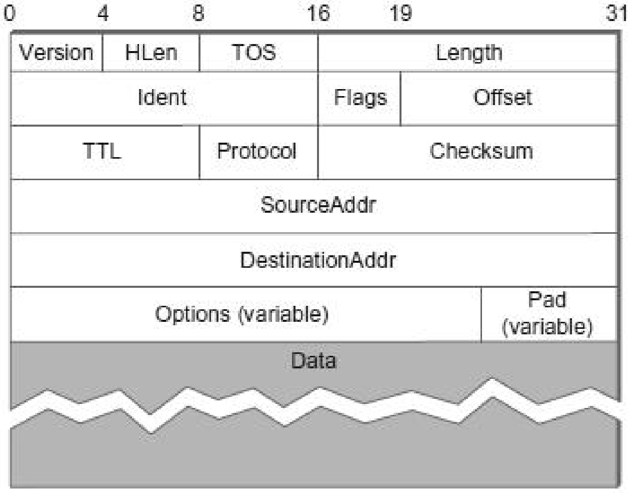
- IP defines the infrastructure that allows these nodes and networks to function as a single logical internetwork.

IP SERVICE MODEL

- Service Model defines the host-to-host services that we want to provide
- The main concern in defining a service model for an internetwork is that we can provide a host-to-host service only if this service can somehow be provided over each of the underlying physical networks.
- The Internet Protocol is the key tool used today to build scalable, heterogeneous internetworks.
- The **IP service model** can be thought of as having **two parts**:
  - A ***GLOBAL ADDRESSING SCHEME*** - which provides a way to identify all hosts in the internetwork
  - A ***DATAGRAM DELIVERY MODEL*** – A connectionless model of data delivery.

IP PACKET FORMAT / IP DATAGRAM FORMAT

- A key part of the IP service model is the type of packets that can be carried.
- The IP datagram consists of a header followed by a number of bytes of data.



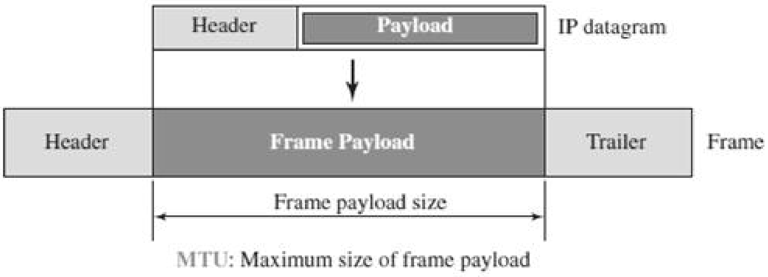
FIELD	DESCRIPTION
Version	Specifies the version of IP. Two versions exists – IPv4 and IPv6.
HLen	Specifies the length of the header
TOS *(*Type of Service)	An indication of the parameters of the quality of service desired such as Precedence, Delay, Throughput and Reliability.
Length	Length of the entire datagram, including the header. The maximum size of an IP datagram is 65,535(2^{10})bytes
Ident **(Identification)**	Uniquely identifies the packet sequence number. Used for fragmentation and re-assembly.

Flags	Used to control whether routers are allowed to fragment a packet. If a packet is fragmented , this flag value is 1.If not, flag value is 0.
Offset (Fragmentation offset)	Indicates where in the datagram, this fragment belongs. The fragment offset is measured in units of 8 octets (64 bits). The first fragment has offset zero.
TTL (Time to Live)	Indicates the maximum time the datagram is allowed to remain in the network. If this field contains the value zero, then the datagram must be destroyed.
Protocol	Indicates the next level protocol used in the data portion of the datagram
Checksum	Used to detect the processing errors introduced into the packet
Source Address	The IP address of the original sender of the packet.
Destination Address	The IP address of the final destination of the packet.
Options	This is optional field. These options may contain values for options such as Security, Record Route, Time Stamp, etc
Pad	Used to ensure that the internet header ends on a 32 bit boundary. The padding is zero.

IP DATAGRAM - FRAGMENTATION AND REASSEMBLY

Fragmentation :

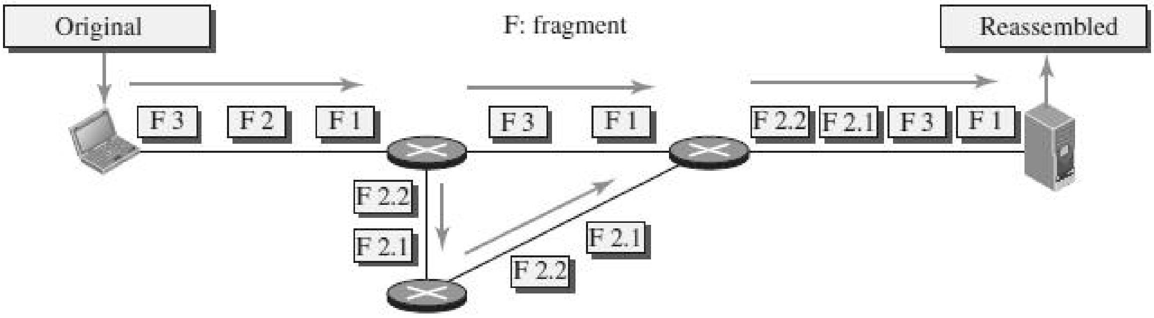
- Every network type has a ***maximum transmission unit*** (MTU), which is the largest IP datagram that it can carry in a frame.



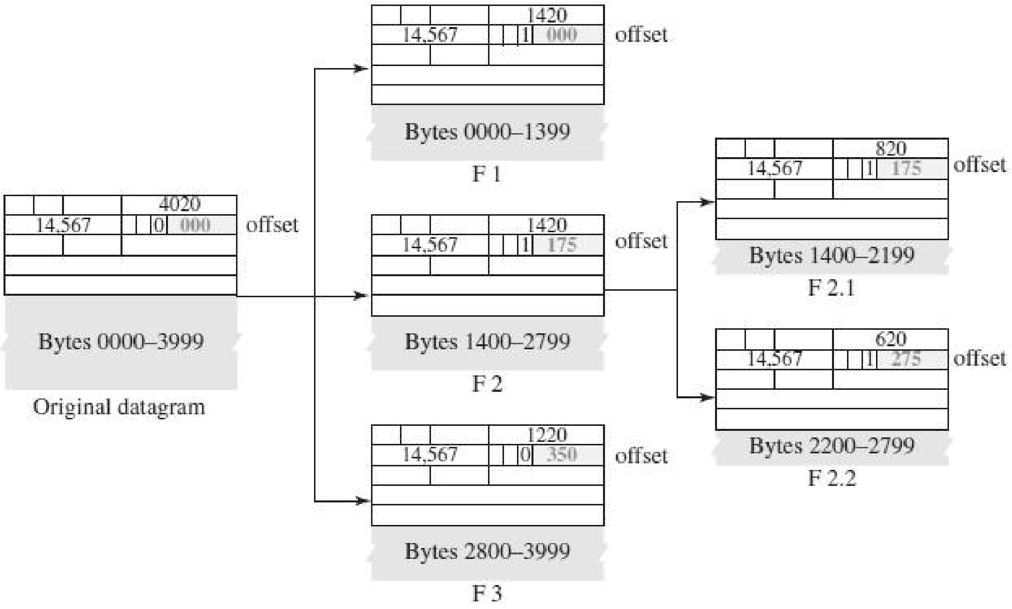
- Fragmentation of a datagram will only be necessary if the path to the destination includes a network with a smaller MTU.
- When a host sends an IP datagram,it can choose any size that it wants.
- Fragmentation typically occurs in a router when it receives a datagram that it wants to forward over a network that has an MTU that is smaller than the received datagram.
- Each fragment is itself a self-contained IP datagram that is transmitted over a sequence of physical networks, independent of the other fragments.
- Each IP datagram is re-encapsulated for each physical network over which it travels.

- For example , if we consider an Ethernet network to accept packets up to 1500 bytes long.
- This leaves two choices for the IP service model:
  - Make sure that all IP datagrams are small enough to fit inside one packet on any network technology
  - Provide a means by which packets can be fragmented and reassembled when they are too big to go over a given network technology.
- Fragmentation produces smaller, valid IP datagrams that can be readily reassembled into the original datagram upon receipt, independent of the order of their arrival.

Example:



- The original packet starts at the client; the fragments are reassembled at the server.
- The value of the identification field is the same in all fragments, as is the value of the flags field with the more bit set for all fragments except the last.
- Also, the value of the offset field for each fragment is shown.
- Although the fragments arrived out of order at the destination, they can be correctly reassembled.



- The value of the offset field is always relative to the original datagram.
- Even if each fragment follows a different path and arrives out of order, the final destination host can reassemble the original datagram from the fragments received (if none of them is lost) using the following strategy:
    1) The first fragment has an offset field value of zero.
    2) Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.
    3) Divide the total length of the first and second fragment by 8. The third fragment has an offset value equal to that result.
    4) Continue the process. The last fragment has its M bit set to 0.
    5) Continue the process. The last fragment has a *more* bit value of 0.

Reassembly:

- Reassembly is done at the receiving host and not at each router.
- To enable these fragments to be reassembled at the receiving host, they all carry the same identifier in the Ident field.
- This identifier is chosen by the sending host and is intended to be unique among all the datagrams that might arrive at the destination from this source over some reasonable time period.
- Since all fragments of the original datagram contain this identifier, the reassembling host will be able to recognize those fragments that go together.
- For example, if a single fragment is lost, the receiver will still attempt to reassemble the datagram, and it will eventually give up and have to garbage-collect the resources that were used to perform the failed reassembly.
- Hosts are now strongly encouraged to perform "path MTU discovery," a process by which fragmentation is avoided by sending packets that are small enough to traverse the link with the smallest MTU in the path from sender to receiver.

IP SECURITY

There are three security issues that are particularly applicable to the IP protocol:

(1) Packet Sniffing (2) Packet Modification and (3) IP Spoofing.

Packet Sniffing

- An intruder may intercept an IP packet and make a copy of it.
- Packet sniffing is a passive attack, in which the attacker does not change the contents of the packet.
- This type of attack is very difficult to detect because the sender and the receiver may never know that the packet has been copied.
- Although packet sniffing cannot be stopped, encryption of the packet can make the attacker's effort useless.
- The attacker may still sniff the packet, but the content is not detectable.

Packet Modification

- The second type of attack is to modify the packet.
- The attacker intercepts the packet,changes its contents, and sends the new packet to the receiver.
- The receiver believes that the packet is coming from the original sender.

- This type of attack can be detected using a data integrity mechanism.
- The receiver, before opening and using the contents of the message, can use this mechanism to make sure that the packet has not been changed during the transmission.

IP Spoofing

- An attacker can masquerade as somebody else and create an IP packet that carries the source address of another computer.
- An attacker can send an IP packet to a bank pretending that it is coming from one of the customers.
- This type of attack can be prevented using an origin authentication mechanism

IP Sec

- The IP packets today can be protected from the previously mentioned attacks using a protocol called IPSec (IP Security).
- This protocol is used in conjunction with the IP protocol.
- IPSec protocol creates a connection-oriented service between two entities in which they can exchange IP packets without worrying about the three attacks such as Packet Sniffing, Packet Modification and IP Spoofing.
- IP Sec provides the following four services:
  1) **Defining Algorithms and Keys :** The two entities that want to create a secure channel between themselves can agree on some available algorithms and keys to be used for security purposes.
  2) **Packet Encryption :** The packets exchanged between two parties can be encrypted for privacy using one of the encryption algorithms and a shared key agreed upon in the first step. This makes the packet sniffing attack useless.
  3) **Data Integrity :** Data integrity guarantees that the packet is not modified during the transmission. If the received packet does not pass the data integrity test, it is discarded.This prevents the second attack, packet modification.
  4) **Origin Authentication :** IPSec can authenticate the origin of the packet to be sure that the packet is not created by an imposter. This can prevent IP spoofing attacks.

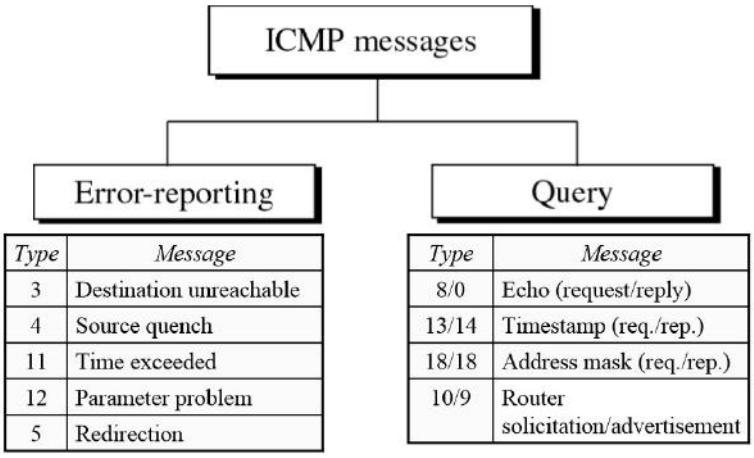
ICMPV4 - INTERNET CONTROL MESSAGE PROTOCOL VERSION 4

- ICMP is a network-layer protocol.
- It is a companion to the IP protocol.
- Internet Control Message Protocol (ICMP) defines a collection of error messages that are sent back to the source host whenever a router or host is unable to process an IP datagram successfully.

ICMP MESSAGE TYPES

- ICMP messages are divided into two broad categories: *error-reporting messages* and *query messages.*
- The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.

- The query messages help a host or a network manager get specific information from a router or another host.



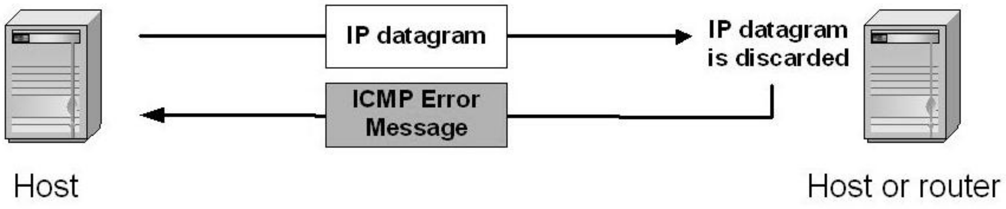
Type	Message
3	Destination unreachable
4	Source quench
11	Time exceeded
12	Parameter problem
5	Redirection

Type	Message
8/0	Echo (request/reply)
13/14	Timestamp (req./rep.)
18/18	Address mask (req./rep.)
10/9	Router solicitation/advertisement

ICMP Error – Reporting Messages

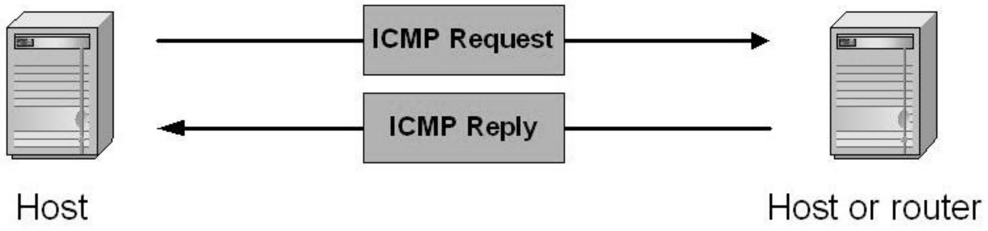
- ICMP error messages report error conditions
- Typically sent when a datagram is discarded
- Error message is often passed from ICMP to the application program

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



ICMP Query Messages

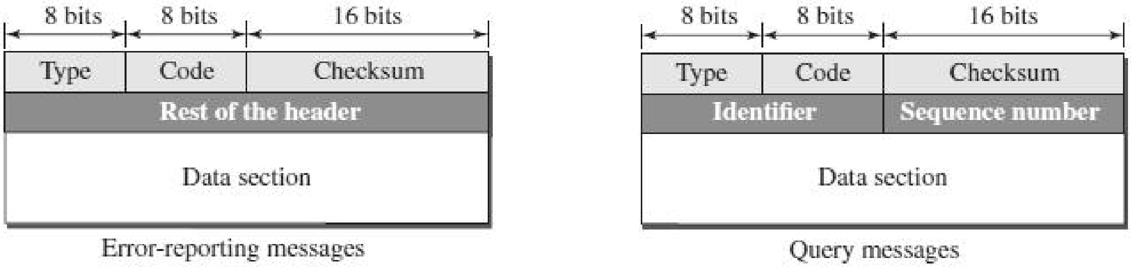
- Request sent by host to a router or host
- Reply sent back to querying host



- ***Echo Request & Reply***—Combination of echo request and reply messages determines whether two systems communicate or not.
- ***Timestamp Request & Reply***—Two machines can use the timestamp request and reply messages to determine the round-trip time (RTT).
- ***Address Mask Request & Reply***—A host to obtain its subnet mask, sends an address mask request message to the router, which responds with an address mask reply message.
- ***Router Solicitation/Advertisement***—A host broadcasts a router solicitation message to know about the router. Router broadcasts its routing information with router advertisement message.

ICMP MESSAGE FORMAT

- An ICMP message has an 8-byte header and a variable-size data section.



Type	Defines the type of the message
Code	Specifies the reason for the particular message type
Checksum	Used for error detection
Rest of the header	Specific for each message type
Data	Used to carry information
Identifier	Used to match the request with the reply
Sequence Number	Sequence Number of the ICMP packet

ICMP DEBUGGING TOOLS

Two tools are used for debugging purpose. They are (1) Ping (2) Traceroute

Ping

- The *ping* program is used to find if a host is alive and responding.
- The source host sends ICMP echo-request messages; the destination, if alive, responds with ICMP echo-reply messages.

- The *ping* program sets the identifier field in the echo-request and echo-reply message and starts the sequence number from 0; this number is incremented by 1 each time a new message is sent.
- The *ping program* can calculate the round-trip time.
- It inserts the sending time in the data section of the message.
- When the packet arrives, it subtracts the arrival time from the departure time to get the round-trip time (RTT).

```
$ ping  google.com
```

Traceroute or Tracert

- The *traceroute* program in UNIX or *tracert* in Windows can be used to trace the path of a packet from a source to the destination.
- It can find the IP addresses of all the routers that are visited along the path.
- The program is usually set to check for the maximum of 30 hops (routers) to be visited.
- The number of hops in the Internet is normally less than this.

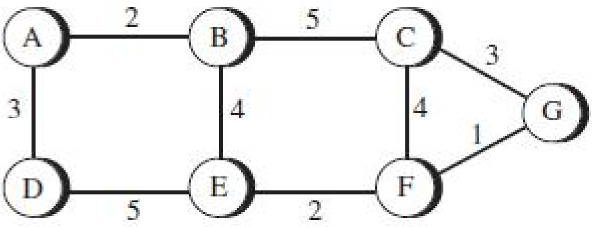
```
$ traceroute  google.com
```

9. UNICAST ROUTING

- Routing is the process of selecting best paths in a network.
- In unicast routing, a packet is routed, hop by hop, from its source to its destination by the help of forwarding tables.
- Routing a packet from its source to its destination means routing the packet from a *source router* (the default router of the source host) to a *destination router* (the router connected to the destination network).
- The source host needs no forwarding table because it delivers its packet to the default router in its local network.
- The destination host needs no forwarding table either because it receives the packet from its default router in its local network.
- Only the intermediate routers in the networks need forwarding tables.

NETWORK AS A GRAPH

- The Figure below shows a graph representing a network.



- The nodes of the graph, labeled A through G, may be hosts, switches, routers, or networks.
- The edges of the graph correspond to the network links.
- Each edge has an associated *cost.*

- The basic problem of routing is to find the lowest-cost path between any two nodes, where the cost of a path equals the sum of the costs of all the edges that make up the path.
- This static approach has several problems:
  - It does not deal with node or link failures.
  - It does not consider the addition of new nodes or links.
  - It implies that edge costs cannot change.
- For these reasons, routing is achieved by running routing protocols among the nodes.
- These protocols provide a distributed, dynamic way to solve the problem of finding the lowest-cost path in the presence of link and node failures and changing edge costs.

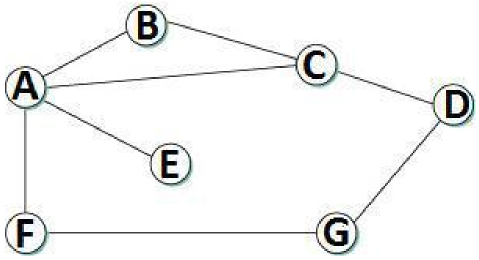
UNICAST ROUTING ALGORITHMS

- There are three main classes of routing protocols:
  1) **Distance Vector Routing Algorithm – Routing Information Protocol**
  2) **Link State Routing Algorithm – Open Shortest Path First Protocol**
  3) **Path-Vector Routing Algorithm - Border Gateway Protocol**

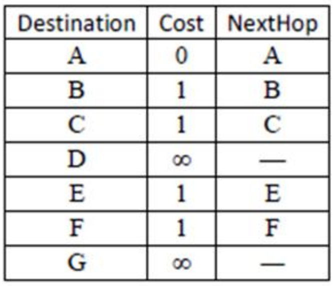
DISTANCE VECTOR ROUTING (DSR)
ROUTING INFORMATION PROTOCOL (RIP)
BELLMAN - FORD ALGORITHM

- Distance vector routing is *distributed*, i.e., algorithm is run on all nodes.
- Each node *knows* the distance (cost) to each of its directly connected neighbors.
- Nodes construct a *vector* (Destination, Cost, NextHop) and distributes to its neighbors.
- Nodes compute routing table of *minimum* distance to every other node via NextHop using information obtained from its neighbors.

Initial State

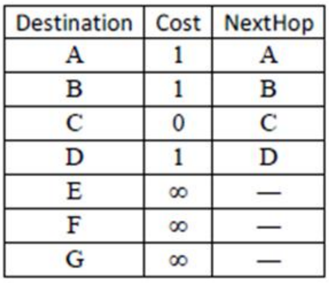


- In given network, *cost* of each link is 1 hop.
- Each node sets a distance of 1 (hop) to its *immediate* neighbor and cost to itself as 0.
- Distance for non-neighbors is marked as *unreachable* with value ∞ (infinity).
- For node *A*, nodes *B*, *C*, *E* and *F* are *reachable*, whereas nodes *D*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



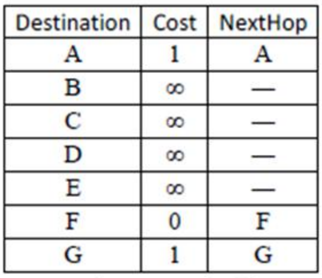
Destination	Cost	NextHop
A	0	A
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

Node A's initial table



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

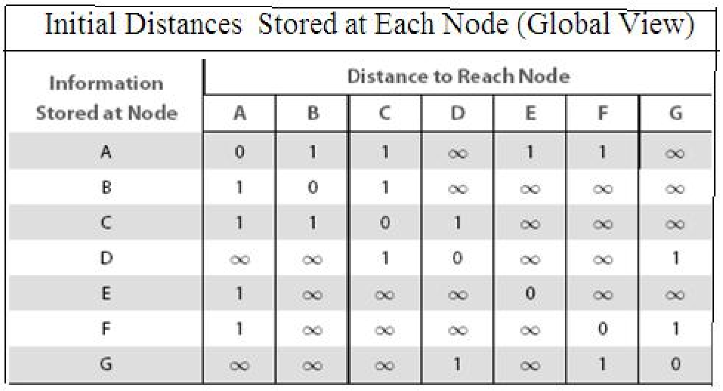
Node C's initial table



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

Node F's initial table

- ➢ The initial table for all the nodes are given below



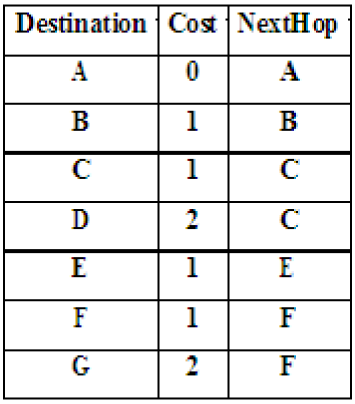
Initial Distances Stored at Each Node (Global View)

Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	∞	1	1	∞
B	1	0	1	∞	∞	∞	∞
C	1	1	0	1	∞	∞	∞
D	∞	∞	1	0	∞	∞	1
E	1	∞	∞	∞	0	∞	∞
F	1	∞	∞	∞	∞	0	1
G	∞	∞	∞	1	∞	1	0

- ➢ Each node *sends* its initial table (distance vector) to neighbors and receives their estimate.
- ➢ Node *A* sends its table to nodes *B*, *C*, *E* & *F* and receives tables from nodes *B*, *C*, *E* & *F*.
- ➢ Each node *updates* its routing table by comparing with each of its neighbor's table
- ➢ For each destination, Total Cost is computed as:
  - ▪ ***Total Cost*** = Cost (*Node* to *Neighbor*) + Cost (*Neighbor* to *Destination*)
- ➢ If Total Cost < Cost then
  - ▪ ***Cost*** = Total Cost and NextHop = *Neighbor*
- ➢ Node *A learns* from *C*'s table to reach node *D* and from *F*'s table to reach node *G*.
- ➢ Total Cost to reach node *D* via *C* = Cost (*A* to *C*) + Cost(*C* to *D*)

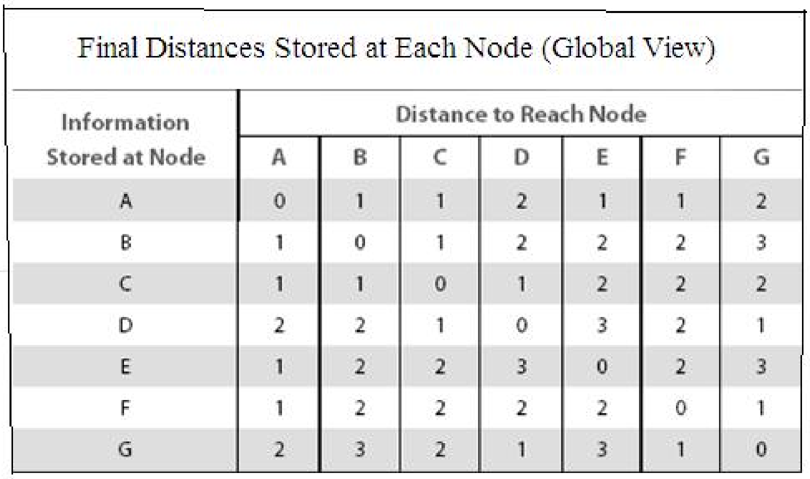
  Cost = 1 + 1 = 2.
  - ▪ Since 2 < ∞, entry for destination *D* in *A*'s table is changed to (*D*, 2, *C*)
  - ▪ Total Cost to reach node *G* via *F* = Cost(*A* to *F*) + Cost(*F* to *G*) = 1 + 1 = 2
  - ▪ Since 2 < ∞, entry for destination *G* in *A*'s table is changed to (*G*, 2, *F*)
- ➢ Each node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

Node A's final routing table



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

- ➢ System stabilizes when all nodes have complete routing information, i.e., **convergence.**
- ➢ Routing tables are exchanged *periodically or* in case of *triggered update*.
- ➢ The final distances stored at each node is given below:



Final Distances Stored at Each Node (Global View)

Information Stored at Node	Distance to Reach Node						
	A	B	C	D	E	F	G
A	0	1	1	2	1	1	2
B	1	0	1	2	2	2	3
C	1	1	0	1	2	2	2
D	2	2	1	0	3	2	1
E	1	2	2	3	0	2	3
F	1	2	2	2	2	0	1
G	2	3	2	1	3	1	0

Updation of Routing Tables

There are two different circumstances under which a given node decides to send a routing update to its neighbors.

Periodic Update

- ➢ In this case, each node automatically sends an update message every so often, even if nothing has changed.
- ➢ The frequency of these periodic updates varies from protocol to protocol, but it is typically on the order of several seconds to several minutes.

Triggered Update

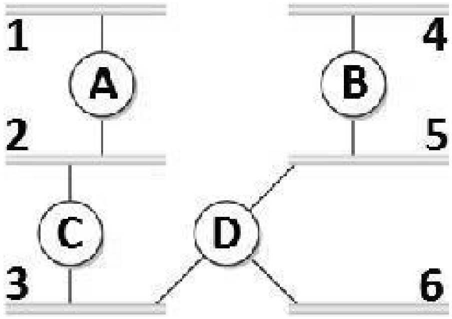
- ➢ In this case, whenever a node notices a link failure or receives an update from one of its neighbors that causes it to change one of the routes in its routing table.

- ➢ Whenever a node's routing table changes, it sends an update to its neighbors, which may lead to a change in their tables, causing them to send an update to their neighbors.

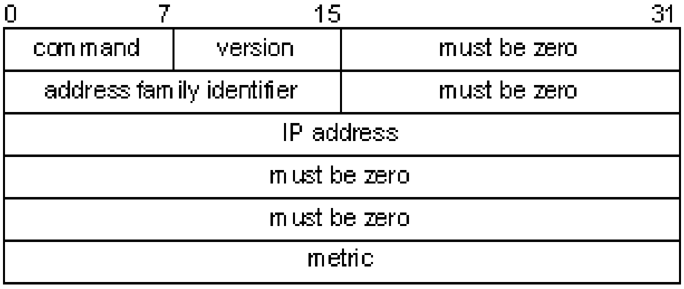
ROUTING INFORMATION PROTOCOL (RIP)

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

Example



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



0	7	15 31
command	version	must be zero
address family identifier		must be zero
IP address		
must be zero		
must be zero		
metric		

- ➢ ***Command*** - It indicates the packet type.
  Value 1 represents a request packet. Value 2 represents a response packet.
- ➢ ***Version*** - It indicates the RIP version number. For RIPv1, the value is 0x01.
- ➢ ***Address Family Identifier*** - When the value is 2, it represents the IP protocol.
- ➢ ***IP Address*** - It indicates the destination IP address of the route. It can be the addresses of only the natural network segment.
- ➢ ***Metric*** - It indicates the hop count of a route to its destination.

<u>Count-To-Infinity (or) Loop Instability Problem</u>

- Suppose link from node *A* to *E* goes *down*.
  - ❖ Node *A* advertises a distance of ∞ to *E* to its neighbors
  - ❖ Node B receives periodic update from C before A's update reaches B
  - ❖ Node *B* updated by *C*, concludes that *E* can be reached in 3 hops via *C*
  - ❖ Node *B* advertises to *A* as 3 hops to reach *E*

- ❖ Node *A* in turn updates *C* with a distance of 4 hops to *E* and so on

- Thus nodes update each other until cost to *E* reaches *infinity*, i.e., *no convergence*.
- Routing table does not stabilize.
- This problem is called *loop instability* or *count to infinity*

Solution to Count-To-Infinity (or) Loop Instability Problem :

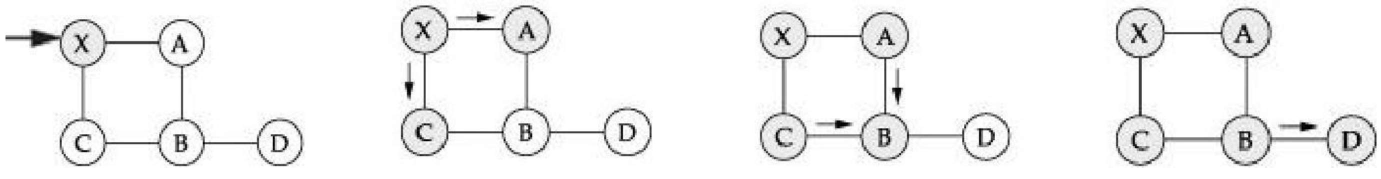
- *Infinity* is redefined to a small number, say 16.
- Distance between any two nodes can be 15 hops maximum. Thus distance vector routing *cannot be used* in large networks.
- When a node updates its neighbors, it does not send those routes it learned from each neighbor back to that neighbor. This is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

LINK STATE ROUTING (LSR)
OPEN SHORTEST PATH PROTOCOL (OSPF)
DIJKSTRA'S ALGORITHM

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

Reliable Flooding

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



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

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

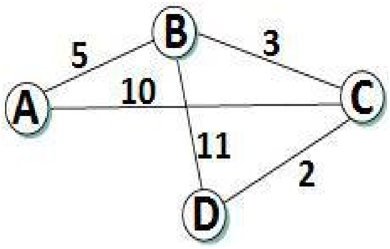
Route Calculation

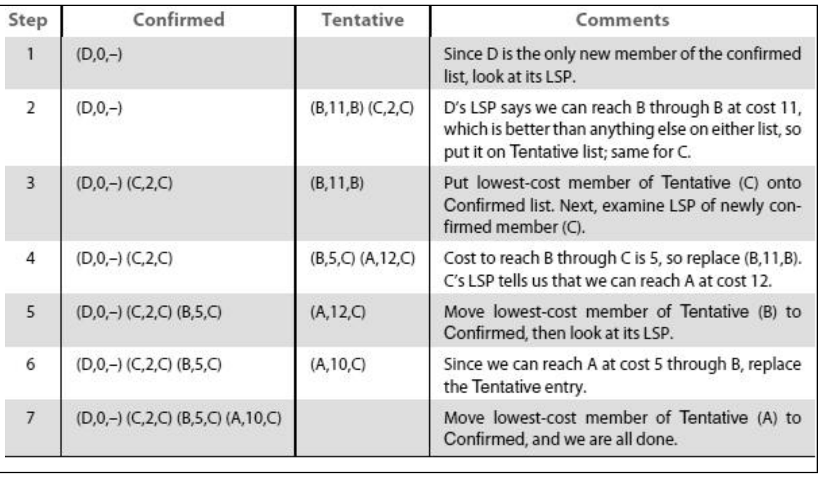
- Each node knows the entire topology, once it has LSP from every other node.
- Forward search algorithm is used to compute routing table from the received LSPs.
- Each node maintains two lists, namely Tentative and Confirmed with entries of the form (Destination, Cost, NextHop).

DIJKSTRA'S SHORTEST PATH ALGORITHM (FORWARD SEARCH ALGORITHM)

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

Example :



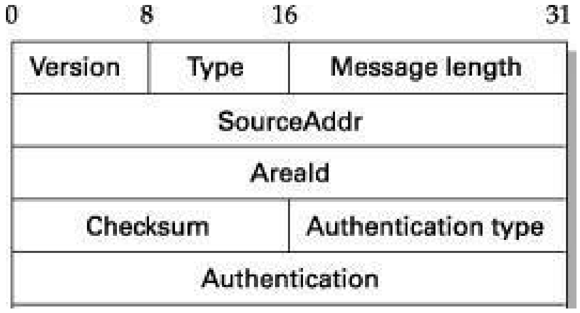


Step	Confirmed	Tentative	Comments
1	(D,0,–)		Since D is the only new member of the confirmed list, look at its LSP.
2	(D,0,–)	(B,11,B) (C,2,C)	D's LSP says we can reach B through B at cost 11, which is better than anything else on either list, so put it on Tentative list; same for C.
3	(D,0,–) (C,2,C)	(B,11,B)	Put lowest-cost member of Tentative (C) onto Confirmed list. Next, examine LSP of newly confirmed member (C).
4	(D,0,–) (C,2,C)	(B,5,C) (A,12,C)	Cost to reach B through C is 5, so replace (B,11,B). C's LSP tells us that we can reach A at cost 12.
5	(D,0,–) (C,2,C) (B,5,C)	(A,12,C)	Move lowest-cost member of Tentative (B) to Confirmed, then look at its LSP.
6	(D,0,–) (C,2,C) (B,5,C)	(A,10,C)	Since we can reach A at cost 5 through B, replace the Tentative entry.
7	(D,0,–) (C,2,C) (B,5,C) (A,10,C)		Move lowest-cost member of Tentative (A) to Confirmed, and we are all done.

OPEN SHORTEST PATH FIRST PROTOCOL (OSPF)

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

Link State Packet Format



0	8	16	31
Version	Type	Message length	
SourceAddr			
AreaId			
Checksum		Authentication type	
Authentication			

- ***Version*** **—** represents the current version, i.e., 2.
- ***Type*** **—** represents the type (1–5) of OSPF message.

  T*ype 1* - “hello” message, T*ype 2* - request, T*ype 3* – send ,
  T*ype 4* - acknowledge the receipt of link state messages ,
  T*ype 5* - reserved
- ***SourceAddr*** **—** identifies the sender
- ***AreaId*** **—** 32-bit identifier of the area in which the node is located
- ***Checksum*** **—** 16-bit internet checksum
- ***Authentication type*** **—** 1 (simple password), 2 (cryptographic authentication).
- ***Authentication*** **—** contains password or cryptographic checksum

Difference Between Distance-Vector And Link-State Algorithms

Distance vector Routing	*Link state Routing*
Each node talks only to its directly connected neighbors, but it tells them everything it has learned (i.e., distance to all nodes).	Each node talks to all other nodes, but it tells them only what it knows for sure (i.e., only the state of its directly connected links).

PATH VECTOR ROUTING (PVR)
BORDER GATEWAY PROTOCOL (BGP)

- Path-vector routing is an asynchronous and distributed routing algorithm.
- The Path-vector routing is not based on least-cost routing.
- The best route is determined by the source using the policy it imposes on the route.
- In other words, the source can control the path.
- Path-vector routing is not actually used in an internet, and is mostly designed to route a packet between ISPs.

Spanning Trees

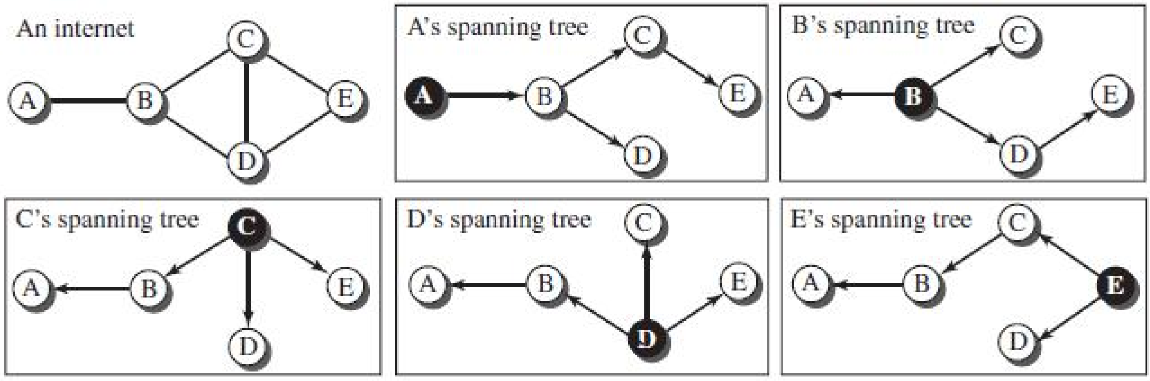
- In path-vector routing, the path from a source to all destinations is determined by the *best* spanning tree.
- The best spanning tree is not the least-cost tree.
- It is the tree determined by the source when it imposes its own policy.
- If there is more than one route to a destination, the source can choose the route that meets its policy best.
- A source may apply several policies at the same time.
- One of the common policies uses the minimum number of nodes to be visited. Another common policy is to avoid some nodes as the middle node in a route.
- The spanning trees are made, gradually and asynchronously, by each node. When a node is booted, it creates a *path vector* based on the information it can obtain about its immediate neighbor.
- A node sends greeting messages to its immediate neighbors to collect these pieces of information.
- Each node, after the creation of the initial path vector, sends it to all its immediate neighbors.
- Each node, when it receives a path vector from a neighbor, updates its path vector using the formula

$$\text{Path}(x, y) = \text{best}\ \{\text{Path}(x, y), [(x + \text{Path}(v, y)]\} \qquad \text{for all } v\text{'s in the internet.}$$

- The policy is defined by selecting the *best* of multiple paths.
- Path-vector routing also imposes one more condition on this equation.
- If Path (**v**, ***y***) includes ***x***, that path is discarded to avoid a loop in the path.
- In other words, ***x*** does not want to visit itself when it selects a path to ***y***.

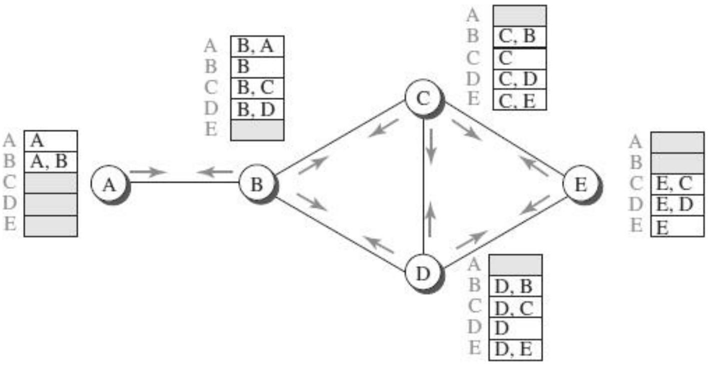
Example:

- The Figure below shows a small internet with only five nodes.
- Each source has created its own spanning tree that meets its policy.
- The policy imposed by all sources is to use the minimum number of nodes to reach a destination.
- The spanning tree selected by A and E is such that the communication does not pass through D as a middle node.
- Similarly, the spanning tree selected by B is such that the communication does not pass through C as a middle node.



Path Vectors made at booting time

- The Figure below shows all of these path vectors for the example.
- Not all of these tables are created simultaneously.
- They are created when each node is booted.
- The figure also shows how these path vectors are sent to immediate neighbors after they have been created.



	A
A	A
B	A, B
C	
D	
E	

	B
A	B, A
B	B
C	B, C
D	B, D
E	

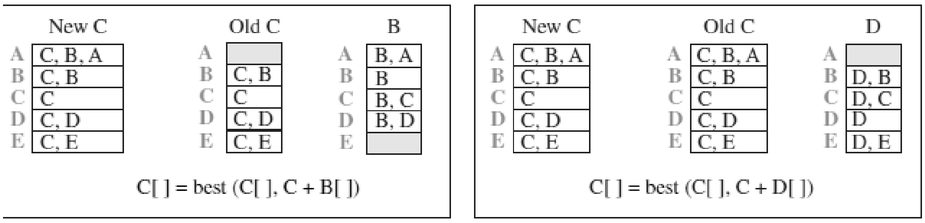
	C
A	
B	C, B
C	C
D	C, D
E	C, E

	D
A	
B	D, B
C	D, C
D	D
E	D, E

	E
A	
B	
C	E, C
D	E, D
E	E

Updating Path Vectors

- The Figure below shows the path vector of node C after two events.
- In the first event, node C receives a copy of B's vector, which improves its vector: now it knows how to reach node A.
- In the second event, node C receives a copy of D's vector, which does not change its vector.
- The vector for node C after the first event is stabilized and serves as its forwarding table.



	New C	Old C	B
A	C, B, A		B, A
B	C, B	C, B	B
C	C	C	B, C
D	C, D	C, D	B, D
E	C, E	C, E	

C[ ] = best (C[ ], C + B[ ])

	New C	Old C	D
A	C, B, A	C, B, A	
B	C, B	C, B	D, B
C	C	C	D, C
D	C, D	C, D	D
E	C, E	C, E	D, E

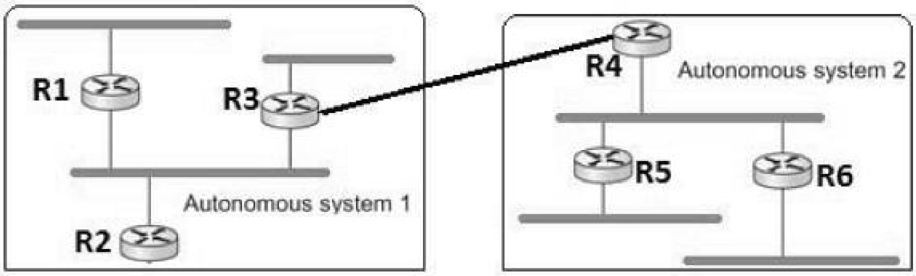
C[ ] = best (C[ ], C + D[ ])

Event 1: C receives a copy of B's vector

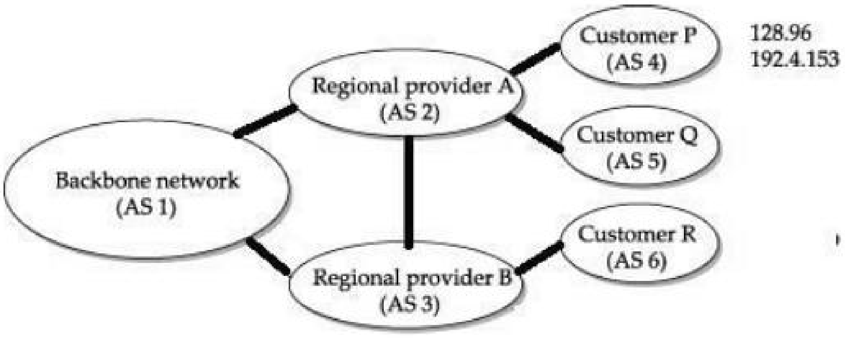
Event 2: C receives a copy of D's vector

BORDER GATEWAY PROTOCOL (BGP)

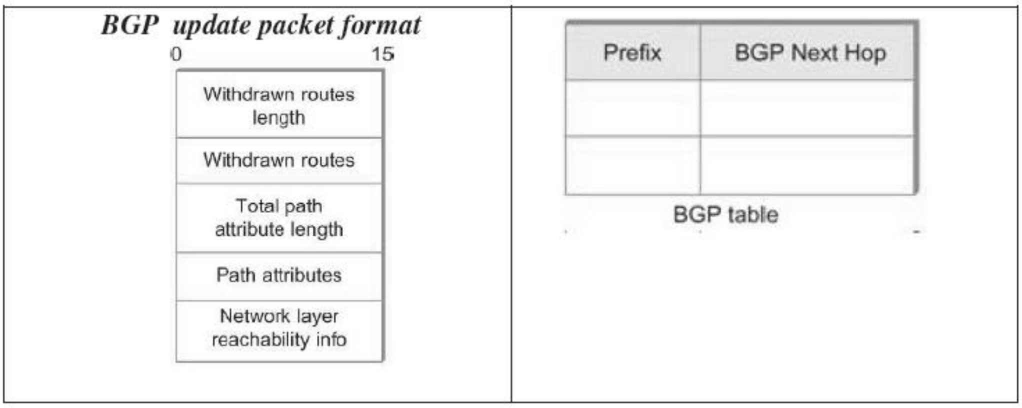
- The Border Gateway Protocol version (BGP) is the only interdomain routing protocol used in the Internet today.
- BGP4 is based on the path-vector algorithm. It provides information about the reachability of networks in the Internet.
- BGP views internet as a set of autonomous systems interconnected arbitrarily.



- Each AS have a *border router* (gateway), by which packets enter and leave that AS. In above figure, *R3* and *R4* are border routers.
- One of the router in each autonomous system is designated as BGP *speaker*.
- BGP Speaker *exchange* reachability information with other BGP speakers, known as *external* BGP session.
- BGP advertises complete *path* as enumerated list of AS (path vector) to reach a particular network.
- Paths must be without any *loop,* i.e., AS list is unique.
- For *example*, backbone network advertises that networks 128.96 and 192.4.153 can be reached along the path <*AS1*, *AS2, AS4*>.



- If there are *multiple* routes to a destination, BGP speaker chooses one based on policy.
- Speakers *need not* advertise any route to a destination, even if one exists.
- Advertised paths can be *cancelled*, if a link/node on the path goes down. This negative advertisement is known as *withdrawn* route.
- Routes are not repeatedly sent. If there is no change, *keep alive* messages are sent.



iBGP - interior BGP

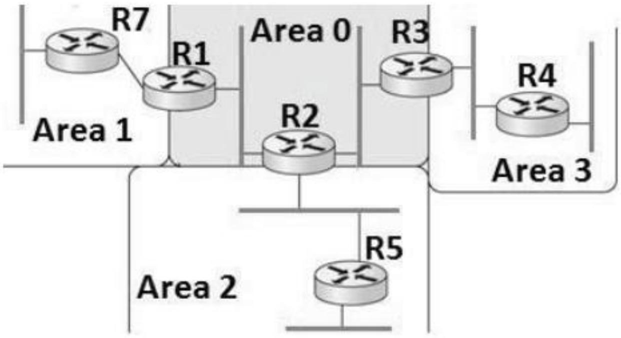
- A Variant of BGP
- Used by routers to update routing information learnt from other speakers to routers inside the autonomous system.
- Each router in the AS is able to determine the appropriate next hop for all prefixes.

10. UNICAST ROUTING PROTOCOLS

- A protocol is more than an algorithm.
- A protocol needs to define its domain of operation, the messages exchanged, communication between routers, and interaction with protocols in other domains.
- A routing protocol specifies how routers communicate with each other, distributing information that enables them to select routes between any two nodes on a computer network.
- Routers perform the "traffic directing" functions on the Internet; data packets are forwarded through the networks of the internet from router to router until they reach their destination computer.
- Routing algorithms determine the specific choice of route.
- Each router has a prior knowledge only of networks attached to it directly.
- A routing protocol shares this information first among immediate neighbors, and then throughout the network. This way, routers gain knowledge of the topology of the network.
- The ability of routing protocols to dynamically adjust to changing conditions such as disabled data lines and computers and route data around obstructions is what gives the Internet its survivability and reliability.
- The specific characteristics of routing protocols include the manner in which they avoid routing loops, the manner in which they select preferred routes, using information about hop costs, the time they require to reach routing convergence, their scalability, and other factors.

INTERNET STRUCTURE

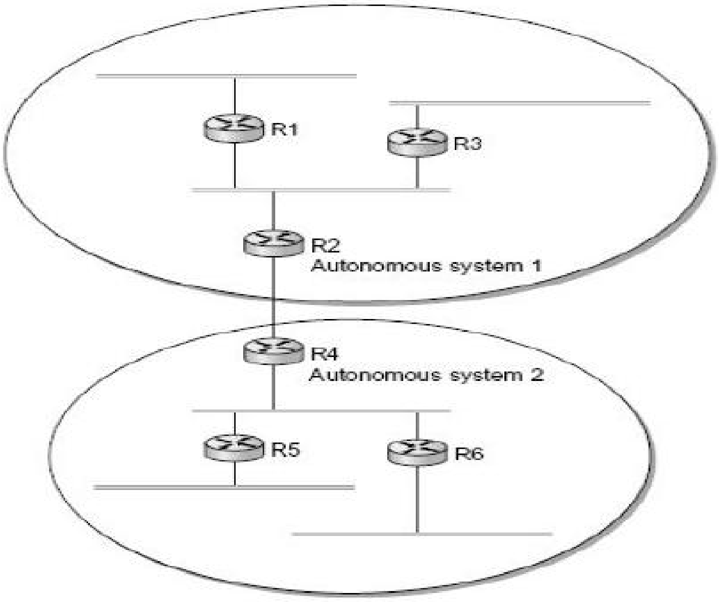
- Internet has a million networks. Routing table entries per router should be minimized.
- Link state routing protocol is used to partition domain into *areas*.
- An routing area is a set of routers configured to exchange link-state information.
- Area introduces an additional level of *hierarchy*.
- Thus domains can grow without burdening routing protocols.



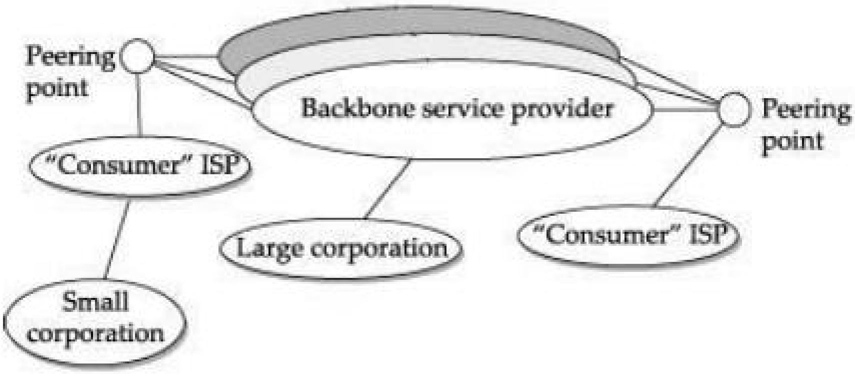
- There is one special area—the ***backbone area***, also known as area 0.
- Routers *R1*, *R2* and *R3* are part of backbone area.
- Routers in backbone area are also part of non-backbone areas. Such routers are known as **A*rea Border Routers*** (ABR).
- Link-state advertisement is *exchanged* amongst routers in a non-backbone area.
- They do not see LSAs of other areas. For example, *area 1* routers are not aware of *area* 3 routers.
- ABR *advertises* routing information in their area to other ABRs.
- For example,*R2* advertises *area 2* routing information to *R1* and *R3*, which in turn pass onto their areas.
- All routers learn how to *reach* all networks in the domain.
- When a packet is to be sent to a network in another area, it goes through backbone area via ABR and reaches the destination area.
- Routing Areas improve scalability but packets may not travel on the shortest path.

INTER DOMAIN ROUTING

- Internet is organized as autonomous systems (AS) each of which is under the control of a single administrative entity.
- A corporation's complex internal network might be a single AS, as may the network of a single Internet Service Provider (ISP).
- Interdomain routing shares reachability information between autonomous systems.



- The basic idea behind autonomous systems is to provide an additional way to hierarchically aggregate routing information in a large internet, thus improving scalability.
- Internet has *backbone* networks and *sites*. Providers connect at a peering point.



Traffic on the internet is of two types:

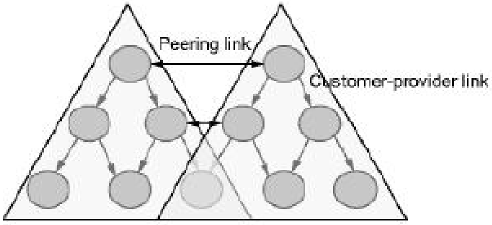
- ***Local Traffic*** - Traffic within an autonomous system is called *local*.
- ***Transit Traffic*** - Traffic that passes through an autonomous system is called *transit*.

Autonomous Systems (AS) are classified as:

- ***Stub AS*** - is connected to only one another autonomous system and carries local traffic only (e.g. Small corporation).
- ***Multihomed AS*** - has connections to multiple autonomous systems but refuses to carry transit traffic (e.g. Large corporation).
- ***Transit AS*** - has connections to multiple autonomous systems and is designed to carry transit traffic (e.g. Backbone service provider).

Policies Used By Autonomous Systems :

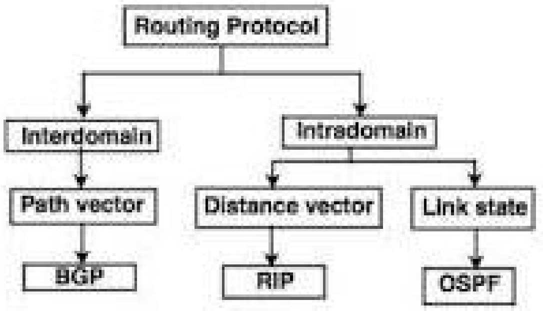
- ***Provider-Customer*—**Provider advertises the routes it knows, to the customer and advertises the routes learnt from customer to everyone.
- ***Customer-Provider*—**Customers want the routes to be diverted to them. So they advertise their own prefixes and routes learned from customers to provider and advertise routes learned from provider to customers.
- ***Peer*—**Two providers access to each other's customers without having to pay.



CHALLENGES IN INTER-DOMAIN ROUTING PROTOCOL

- Each autonomous system has an intra-domain routing protocol, its own policy and metric.
- Internet backbone must be able to route packets to the destination that complies with policies of autonomous system along a loopless path.
- Service providers have trust deficit and may not trust advertisements by other AS, or may refuse to carry traffic from other AS.

TYPES OF ROUTING PROTOCOLS



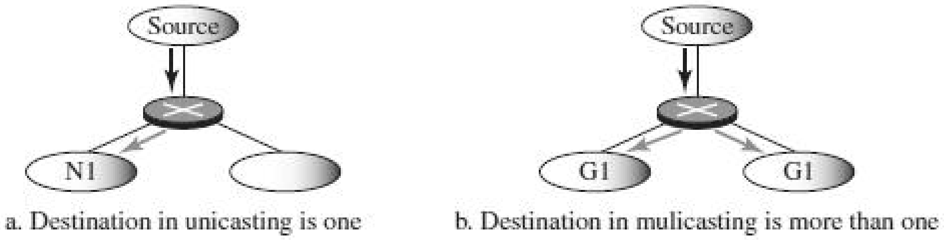
Two types of Routing Protocols are used in the Internet:

1) **Intradomain routing**
   - Routing within a single autonomous system
   - Routing Information Protocol (RIP) - based on the distance-vector algorithm - (REFER distance-vector routing algorithm)
   - Open Shortest Path First (OSPF) - based on the link-state algorithm - (REFER link-state routing algorithm)

2) **Interdomain routing**
   - Routing between autonomous systems.
   - Border Gateway Protocol (BGP) - based on the path-vector algorithm - (REFER Path Vector routing algorithm)

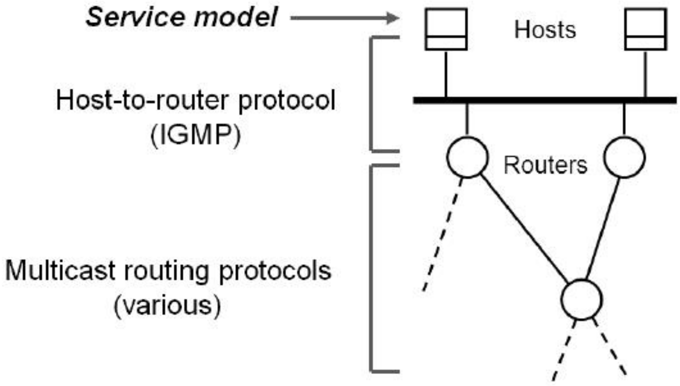
11. MULTICASTING

- In multicasting, there is one source and a group of destinations.
- Multicast supports efficient delivery to multiple destinations.
- The relationship is one to many or many-to-many.
- **One-to-Many (Source Specific Multicast)**
  - Radio station broadcast
  - Transmitting news, stock-price
  - Software updates to multiple hosts
- **Many-to-Many (Any Source Multicast)**
  - Multimedia teleconferencing
  - Online multi-player games
  - Distributed simulations
- In this type of communication, the source address is a unicast address, but the destination address is a group address.
- The group address defines the members of the group.



a. Destination in unicasting is one b. Destination in mulicasting is more than one

- In multicasting, a multicast router may have to send out copies of the same datagram through more than one interface.
- Hosts that are members of a group receive copies of any packets sent to that group's multicast address
- A host can be in multiple groups
- A host can join and leave groups
- A host signals its desire to join or leave a multicast group by communicating with its local router using a special protocol.
- In IPv4, the protocol is Internet Group Management Protocol (IGMP)
- In IPv6, the protocol is Multicast Listener Discovery (MLD)

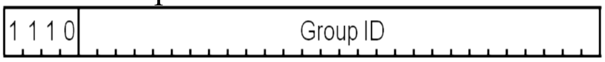

Service model
Hosts
Host-to-router protocol (IGMP)
Routers
Multicast routing protocols (various)

IGMP OR MLD PROTOCOL

- Hosts communicate their desire to *join* / *leave* a multicast group to a router using Internet Group Message Protocol (IGMP) in IPv4 or Multicast Listener Discovery (MLD) in IPv6.

- Provides multicast routers with information about the membership status of hosts connected to the network.
- Enables a multicast router to create and update list of loyal members for each group.

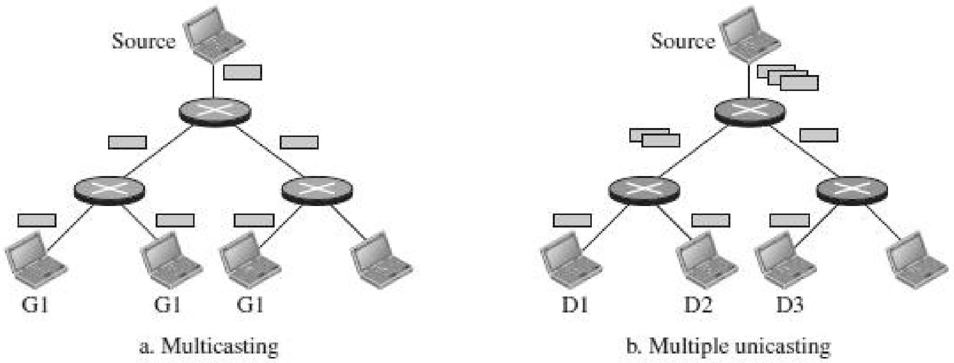
MULTICAST ADDRESSING

- Multicast address is associated with a group, whose members are dynamic.
- Each group has its own IP multicast address.
- IP addresses reserved for multicasting are Class D in IPv4 (Class D 224.0.0.1 to 239.255.255.255), 1111 1111 prefix in IPv6.
  - ○ 

- Hosts that are members of a group receive copy of the packet sent when destination contains group address.

MULTICASTING VERSUS MULTIPLE UNICASTING

- **Multicasting** starts with a single packet from the source that is duplicated by the routers. The destination address in each packet is the same for all duplicates.
- Only a single copy of the packet travels between any two routers.



a. Multicasting

b. Multiple unicasting

- In **multiple unicasting**, several packets start from the source.
- If there are three destinations, for example, the source sends three packets, each with a different unicast destination address.
- There may be multiple copies traveling between two routers

NEED FOR MULTICAST

Without support for multicast

- A source needs to send a separate packet with the identical data to each member of the group
- Source needs to keep track of the IP address of each member in the group

Using IP multicast

- Sending host does not send multiple copies of the packet
- A host sends a single copy of the packet addressed to the group's multicast address
- The sending host does not need to know the individual unicast IP address of each member

TYPES OF MULTICASTING

- ***Source-Specific Multicast*** - In *source-specific* multicast (one-to-many model), receiver specifies multicast group and sender from which it is interested to receive packets. Example: Internet radio broadcasts.

- ***Any Source Multicast*** - Supplements *any source* multicast (many-to-many model).

MULTICAST APPLICATIONS

- Access to Distributed Databases
- Information Dissemination
- Teleconferencing.
- Distance Learning

MULTICAST ROUTING

- To support multicast, a router must additionally have multicast forwarding tables that indicate, based on multicast address, which links to use to forward the multicast packet.
- Unicast forwarding tables collectively specify a set of paths.
- Multicast forwarding tables collectively specify a set of trees -Multicast distribution trees.
- Multicast routing is the process by which multicast distribution trees are determined.
- To support multicasting, routers *additionally* build multicast forwarding tables.
- Multicast forwarding table is a tree structure, known as ***multicast distribution trees.***
- Internet multicast is implemented on physical networks that support broadcasting by *extending* forwarding functions.

MULTICAST DISTRIBUTION TREES

There are two types of Multicast Distribution Trees used in multicast routing.
They are

- **Source-Based Tree**: (DVMRP)
  - For each combination of (source , group), there is a shortest path spanning tree.
  - ***Flood and prune***
    - Send multicast traffic everywhere
    - Prune edges that are not actively subscribed to group
  - ***Link-state***
    - Routers flood groups they would like to receive
    - Compute shortest-path trees on demand
- **Shared Tree** (PIM)
  - Single distributed tree shared among all sources
  - Does not include its own topology discovery mechanism, but instead uses routing information supplied by other routing protocols
  - Specify rendezvous point (RP) for group
  - Senders send packets to RP, receivers join at RP

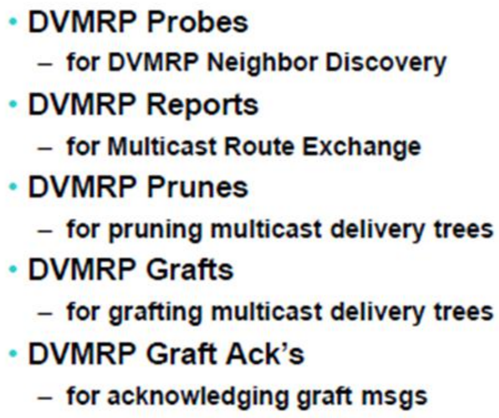
- RP multicasts to receivers; Fix-up tree for optimization
- ***Rendezvous-Point Tree***: one router is the center of the group and therefore the root of the tree.

MULTICAST ROUTING PROTOCOLS

- Internet multicast is implemented on physical networks that support broadcasting by *extending forwarding functions.*
- Major multicast routing protocols are:
  1. Distance-Vector Multicast Routing Protocol (DVMRP)
  2. Protocol Independent Multicast (PIM)

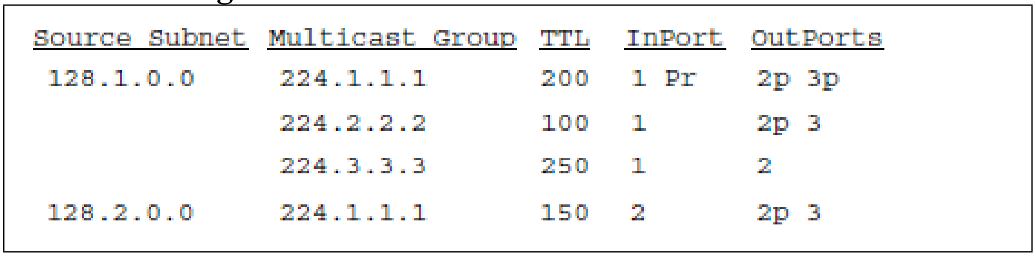
1. Distance Vector Multicast Routing Protocol

- The DVMRP, is a routing protocol used to share information between routers to facilitate the transportation of IP multicast packets among networks.
- It formed the basis of the Internet's historic multicast backbone.
- Distance vector routing for unicast is extended to support multicast routing.
- Each router maintains a routing table for all destination through exchange of distance vectors.
- DVMRP is also known as ***flood-and-prune protocol***.
- DVMRP consists of two major components:
- A conventional distance-vector routing protocol, like RIP
- A protocol for determining how to forward multicast packets, based on the routing table
- DVMRP router forwards a packet if
- The packet arrived from the link used to reach the source of the packet
- If downstream links have not pruned the tree
- DVMRP protocol uses the **basic packet types** as follows:



- DVMRP Probes
  - for DVMRP Neighbor Discovery
- DVMRP Reports
  - for Multicast Route Exchange
- DVMRP Prunes
  - for pruning multicast delivery trees
- DVMRP Grafts
  - for grafting multicast delivery trees
- DVMRP Graft Ack's
  - for acknowledging graft msgs

- The **forwarding table** of DVMRP is as follows:



Source Subnet	Multicast Group	TTL	InPort	OutPorts
128.1.0.0	224.1.1.1	200	1 Pr	2p 3p
	224.2.2.2	100	1	2p 3
	224.3.3.3	250	1	2
128.2.0.0	224.1.1.1	150	2	2p 3

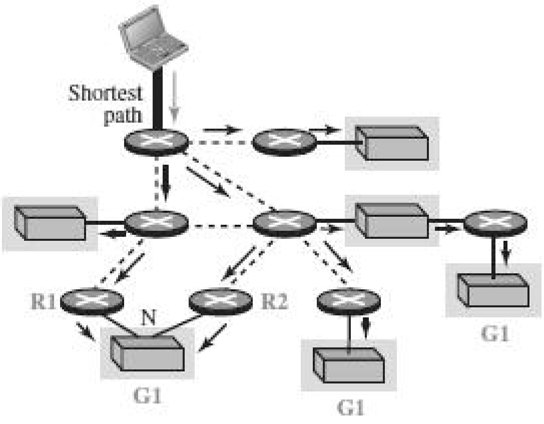
- Multicasting is added to distance-vector routing in four stages.
  - ➢ Flooding
  - ➢ Reverse Path Forwarding (RPF)
  - ➢ Reverse Path Broadcasting (RPB)
  - ➢ Reverse Path Multicast (RPM)

Flooding

- Router on receiving a multicast packet from source *S* to a Destination from NextHop, *forwards* the packet on all out-going links.
- Packet is flooded and looped back to *S*.
- The drawbacks are:
  - o It floods a network, even if it has *no members* for that group.
  - o Packets are forwarded by each router connected to a LAN, i.e., *duplicate flooding*

Reverse Path Forwarding (RPF)

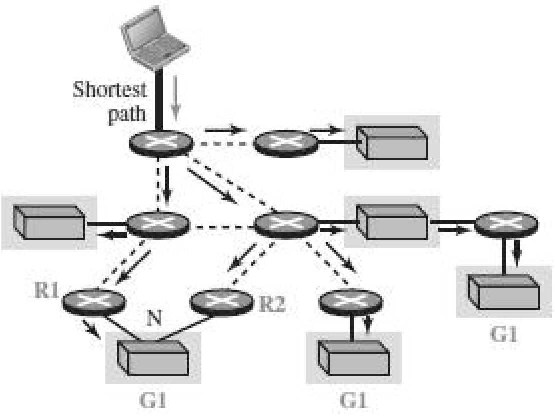
- RPF eliminates the looping problem in the flooding process.
- Only one copy is forwarded and the other copies are discarded.
- RPF forces the router to forward a multicast packet from one specific interface: the one which has come through the shortest path from the source to the router.
- Packet is flooded but not looped back to *S*.



Using RPF, N receives two copies.

Reverse-Path Broadcasting (RPB)

- RPB does not multicast the packet, it broadcasts it.
- RPB creates a shortest path broadcast tree from the source to each destination.
- It guarantees that each destination receives one and only one copy of the packet.
- We need to prevent each network from receiving more than one copy of the packet.
- If a network is connected to more than one router, it may receive a copy of the packet from each router.
- One router identified as parent called designated Router (DR).
- Only parent router *forwards* multicast packets from source *S to the attached network.*
- When a router that is not the parent of the attached network receives a multicast packet, it simply drops the packet.



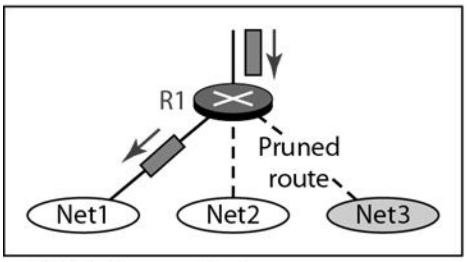
Using RPB, N receives only one copy.

Reverse-Path Multicasting (RPM)

- To increase efficiency, the multicast packet must reach only those networks that have active members for that particular group.
- RPM adds pruning and grafting to RPB to create a multicast shortest path tree that supports dynamic membership changes.

Pruning:

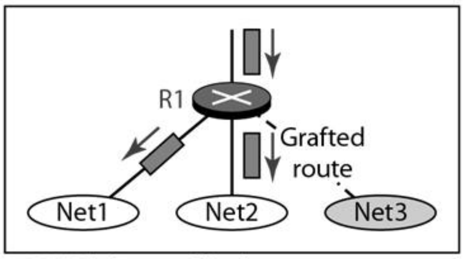
- Sent from routers receiving multicast traffic for which they have no active group members
- "Prunes" the tree created by DVMRP
- Stops needless data from being sent



RPM (after pruning)

Grafting:

- Used after a branch has been pruned back
- Sent by a router that has a host that joins a multicast group
- Goes from router to router until a router active on the multicast group is reached
- Sent for the following cases
  - A new host member joins a group
  - A new dependent router joins a pruned branch
  - A dependent router restarts on a pruned branch



RPM (after grafting)

2. Protocol Independent Multicast (PIM)

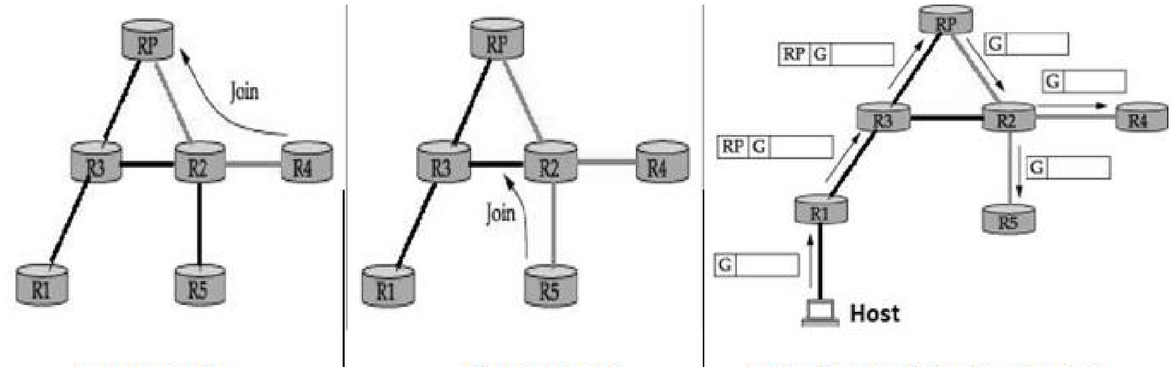
- PIM divides multicast routing problem into *sparse* and *dense* mode.
- PIM sparse mode (PIM-SM) is widely used.
- PIM does not rely on any type of unicast routing protocol, hence protocol independent.
- Routers explicitly join and leave multicast group using ***Join and Prune messages***.
- One of the router is designated as *rendezvous point* (RP) for each group in a domain to receive PIM messages.
- Multicast forwarding *tree* is built as a result of routers sending Join messages to RP.
- Two types of trees to be constructed:
  - ***Shared tree*** - used by all senders
  - ***Source-specific* tree -** used only by a specific sending host
- The normal mode of operation creates the shared tree first, followed by one or more source-specific trees

Shared Tree

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

Example

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



Join from R4 *Join from R5* *Multicast message to group G*

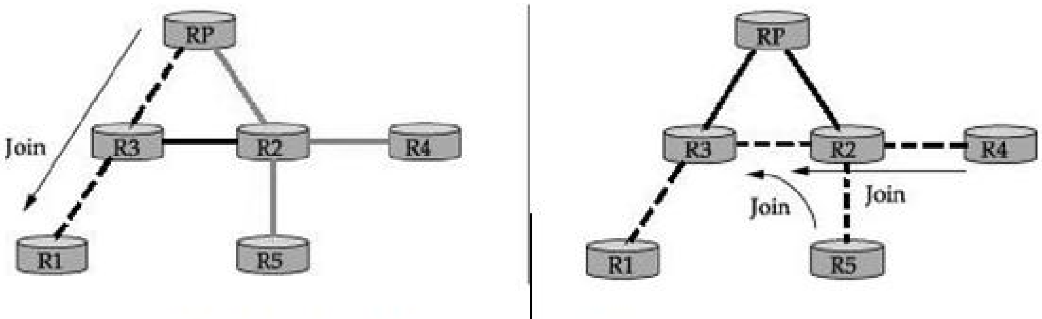
- As routers send Join message for a group, branches are *added* to the tree, i.e., shared.
- Multicast packets sent from hosts are forwarded to *designated* router RP.

- Suppose router *R1*, receives a message to group *G*.
  - *R1* has no state for group *G.*
  - Encapsulates the multicast packet in a Register message.
  - Multicast packet is tunneled along the way to RP.
- RP decapsulates the packet and sends multicast packet onto the shared tree, towards *R2*.
- *R2* forwards the multicast packet to routers *R4* and *R5* that have members for group *G*.

Source-Specific Tree

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

Example



Source-specific Join from RP

Routers switch to Source tree

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

Analysis of PIM

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

12. IPV6 - NEXT GENERATION IP

- IPv6 was evolved to solve address space problem and offers rich set of services.
- Some hosts and routers will run IPv4 only, some will run IPv4 and IPv6 and some will run IPv6 only.

DRAWBACKS OF IPV4

- Despite subnetting and CIDR, address depletion is still a long-term problem.
- Internet must accommodate real-time audio and video transmission that requires minimum delay strategies and reservation of resources.
- Internet must provide encryption and authentication of data for some applications

FEATURES OF IPV6

1. ***Better header format* -** IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
2. ***New options* -** IPv6 has new options to allow for additional functionalities.
3. ***Allowance for extension* -** IPv6 is designed to allow the extension of the protocol if required by new technologies or applications.
4. ***Support for resource allocation* -** In IPv6, the type-of-service field has been removed, but two new fields, traffic class and flow label, have been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.

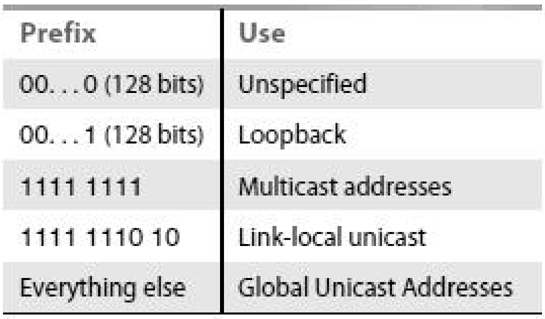
Additional Features :

1. Need to accommodate scalable routing and addressing
2. Support for real-time services
3. Security support
4. Autoconfiguration -

   The ability of hosts to automatically configure themselves with such information as their own IP address and domain name.
5. Enhanced routing functionality, including support for mobile hosts
6. Transition from ipv4 to ipv6

ADDRESS SPACE ALLOCATION OF IPV6

- IPv6 provides a 128-bit address space to handle up to 3.4 × 10^{38} nodes.
- IPv6 uses *classless* addressing, but classification is based on MSBs.
- The current assignment of prefixes is listed in Table



Prefix	Use
00. . . 0 (128 bits)	Unspecified
00. . . 1 (128 bits)	Loopback
1111 1111	Multicast addresses
1111 1110 10	Link-local unicast
Everything else	Global Unicast Addresses

- A node may be assigned an “IPv4-compatible IPv6 address” by zero-extending a 32-bit IPv4 addressto128 bits.

- A node that is only capable of understanding IPv4 can be assigned an "IPv4-mapped IPv6 address" by prefixing the 32-bit IPv4 address with 2 bytes of all 1s and then zero-extending the result to 128 bits.

GLOBAL UNICAST

- Large chunks (87%) of address space are left *unassigned* for future use.
- **IPv6 defines two types of *local* addresses for private networks**.
  - ***Link local*** - enables a host to construct an address that need not be globally unique.
  - ***Site local*** - allows valid local address for use in a isolated site with several subnets.
- ***Reserved* addresses start with prefix of eight 0's.**
  - ***Unspecified address*** is used when a host does not know its address
  - ***Loopback address*** is used for testing purposes before connecting
  - ***Compatible address*** is used when IPv6 hosts uses IPv4 network
  - M***apped address*** is used when a IPv6 host communicates with a IPv4 host
- IPv6 defines *anycast* address, assigned to a set of interfaces.
- Packet with anycast address is delivered to only one of the nearest interface.

ADDRESS NOTATION OF IPV6

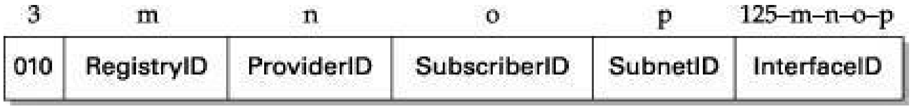
- Standard representation of IPv6 address is ***x* : *x* : *x* : *x* : *x* : *x* : *x* : *x*** where *x* is a 16-bit hexadecimal address separated by colon (:).
  For example,
  47CD **:** 1234 **:** 4422 **:** ACO2 **:** 0022 **:** 1234 **:** A456 **:** 0124

- IPv6 address with contiguous 0 bytes can be written compactly.
  For example,
  47CD **:** 0000 **:** 0000 **:** 0000 **:** 0000 **:** 0000 **:** A456 **:** 0124 → 47CD **: :** A456 **:** 0124

- IPv4 address is mapped to IPv6 address by prefixing the 32-bit IPv4 address with 2 bytes of 1s and then zero-extending the result to 128 bits.
  For example,
  128.96.33.81 → : : FFFF : 128.96.33.81
  This notation is called as CIDR notation or slash notation.

ADDRESS AGGREGATION OF IPV6

- IPv6 provides *aggregation* of routing information to reduce the burden on routers.
- Aggregation is done by assigning prefixes at *continental* level.
- For *example*, if all addresses in Europe have a common prefix, then routers in other continents would need one routing table entry for all networks in Europe.

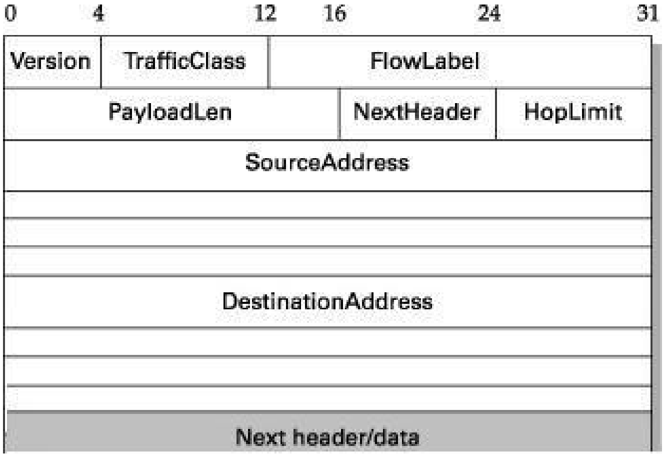


❖ ***Prefix*** - All addresses in the same continent have a common prefix
❖ ***RegistryID*** **—** identifies the continent
❖ ***ProviderID*** **—** identifies the provider for Internet access, i.e., ISP.
❖ ***SubscriberID*** **—** specifies the subscriber identifier

- ***SubnetID* —** contains subnet of the subscriber.
- ***InterfaceID* —**contains link level or physical address.

PACKET FORMAT OF IPV6

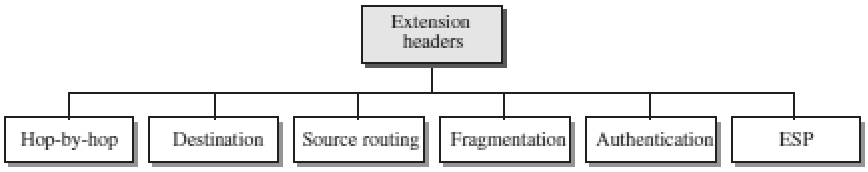
- IPv6 base header is 40 bytes long.



- ***Version*** — specifies the IP version, i.e., 6.
- ***Traffic Class*** — defines priority of the packet with respect to traffic congestion. It is either congestion-controlled or non-congestion controlled
- ***Flow Label*** — provides special handling for a particular flow of data. Router handles different flows with the help of a flow table.
- ***Payload Len*** — gives length of the packet, excluding IPv6 header.
- ***Next Heade***r — Options are specified as a header following IP header. NextHeader contains a pointer to optional headers.
- ***Hop Limit*** — Gives the TTL value of a packet.
- ***Source Address / Destination Address*** — 16-byte addresses of source and destination host

Extension Headers

- Extension header provides greater functionality to IPv6.
- Base header may be followed by six extension headers.
- Each extension header contains a NextHeader field to identify the header following it.



- ***Hop-by-Hop*** — source host passes information to all routers visited by the packet
- ***Destination*** — source host information is passed to the destination only.
- ***Source Routing*** — routing information provided by the source host.
- ***Fragmentation*** — In IPv6, only the source host can fragment. Source uses a path MTU discovery technique to find smallest MTU on the path.
- ***Authentication*** — used to validate the sender and ensures data integrity.
- ***ESP (Encrypted Security Payload)*** — provides confidentiality against eavesdropping.

ADVANCED CAPABILITIES OF IPV6

- **Auto Configuration** — Auto or stateless configuration of IP address to hosts without the need for a DHCP server, i.e., plug and play.
- **Advanced Routing** — Enhanced routing support for mobile hosts is provided.
- **Additional Functions —** Enhanced routing functionality with support for mobile hosts.
- **Security —** Encryption and authentication options provide confidentiality and integrity.
- **Resource allocation —** Flow label enables the source to request special handling of real-time audio and video packets

ADVANTAGES OF IPV6

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

Dual-Stack Operation and Tunneling

- In dual-stack, nodes run both IPv6 and IPv4, uses Version field to decide which stack should process an arriving packet.
- IPv6 packet is encapsulated with an IPv4 packet as it travels through an IPv4 network. This is known as tunneling and packet contains tunnel endpoint as its destination address.

Network Address Translation

- NAT enables hosts on a network to use Internet with local addresses.
- Addresses reserved for internal use range from 172.16.0.0 to 172.31.255.255
- Organization must have single connection to the Internet through a router that runs the NAT software.