JGI JAIN DEEMED-TO-BE UNIVERSITY | SCHOOL OF COMPUTER SCIENCE AND IT

Master of Computer Applications

23MCAC102 – Advanced Computer Networks

Module - III

MODULE 3 Syllabus

NETWORK LAYER

- Network Layer Services
- IPv4 Packet format
- IPv4 Addresses
- Network Layer Protocols: IP, ICMPv4
- Unicast Routing Protocols
- IPV6 Addressing
- IPV6 Protocol
- Mobile IP

NETWORK LAYER SERVICES

NETWORK LAYER SERVICES

- The network layer in the TCP/IP protocol suite is **responsible for the host-to-host delivery of datagrams.**
- 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**

i) Switching

ii) Routing and

iii) 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

i) 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.**

Services provided by Network layer (Contd...)

ii) Routing

- Routing is the concept of **applying strategies and running routing protocols to create the decision-making (routing) tables** for each router.
- 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**.

iii) 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.**

Services provided by network layer (Contd…)

iv) 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.**
- 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.

v) 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 knowing the readiness of the receiver.**

Services provided by Network layer (Contd…)

vi) 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.**

vii) 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 **IPSec (IP Security).**
- **IPSec is a secure network protocol suite** that **authenticates and encrypts packets of data to provide secure encrypted communication** between two computers over an Internet Protocol network

IPv4 ADDRESSES

IPv4 ADDRESSES

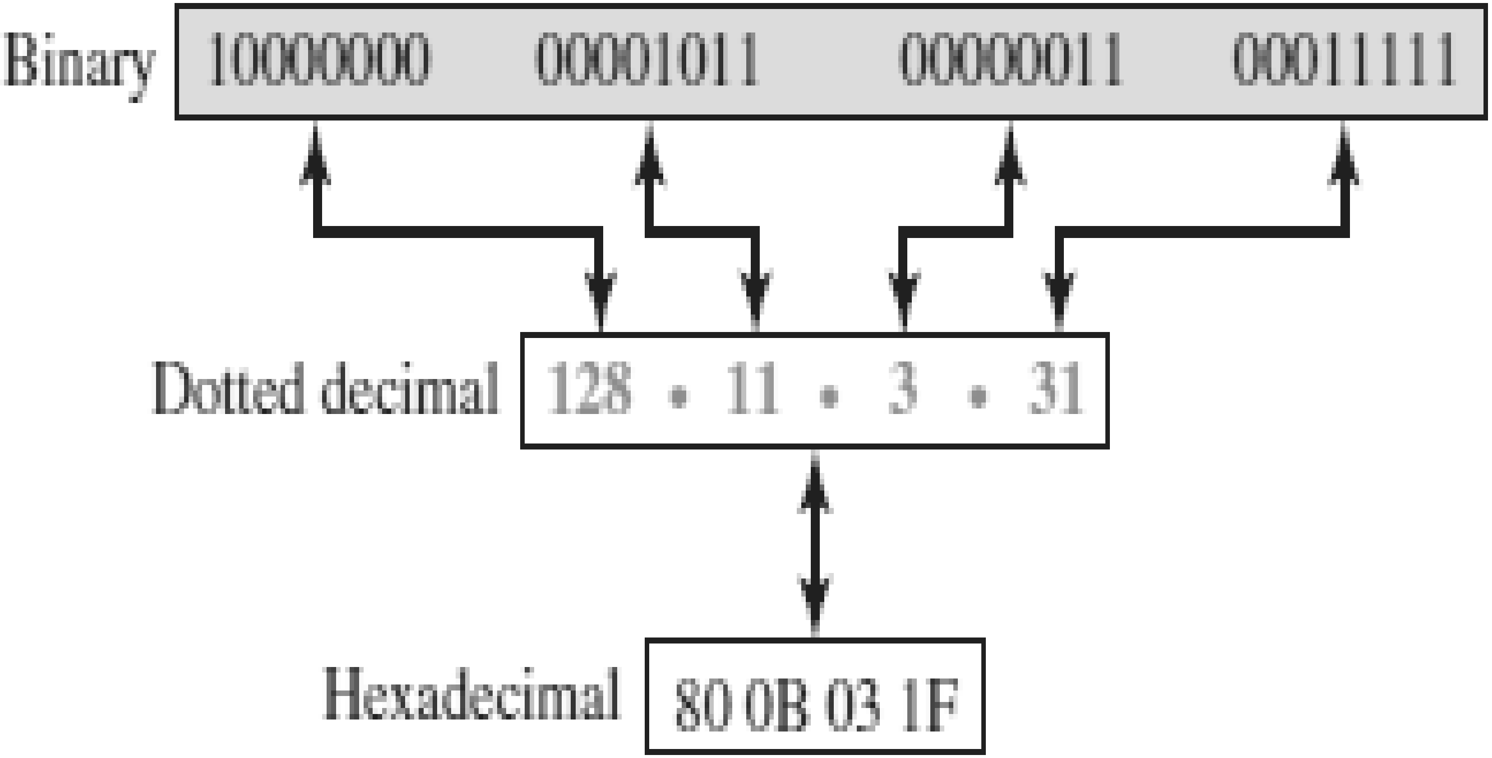
- The identifier used in the IP layer of the TCP/IP protocol suite **to identify the connection of each device to the Internet** is called the Internet address or IP address.
- The **IP address is the address of the connection**, **not the host or the router.**
- An IPv4 address is a **32-bit address that uniquely and universally defines the connection.**
- **If the device is moved to another network**, the IP address may be changed.
- If a device has two connections to the Internet, via two networks, **it has two IPv4 addresses.**

IPv4 ADDRESS SPACE

- An address space is the **total number of addresses used by the protocol.**
- **IPv4 uses 32-bit addresses**, which means that the address space is 2^{32} or 4,294,967,296 **(more than four billion).**

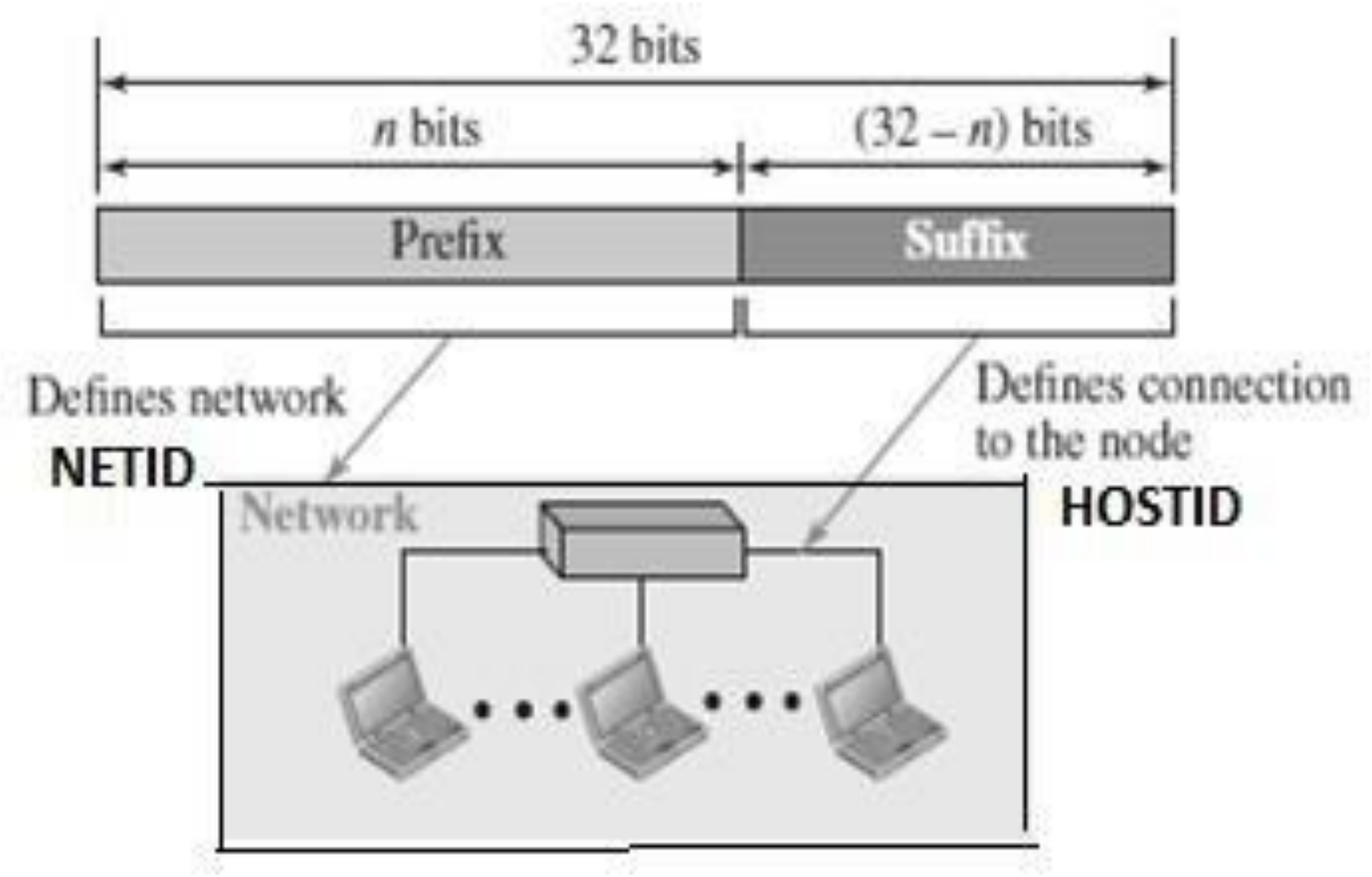
IPv4 ADDRESS NOTATION

- 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.


Binary
10000000
00001011
00000011
00011111
Dotted decimal
128 • 11 • 3 • 31
Hexadecimal
80 0B 03 1F

HIERARCHY IN IPv4 ADDRESSING

- A 32-bit IPv4 address is 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.**


32 bits
n bits
(32 – n) bits
Prefix
Suffix
Defines network
NETID
Defines connection to the node
HOSTID
Network

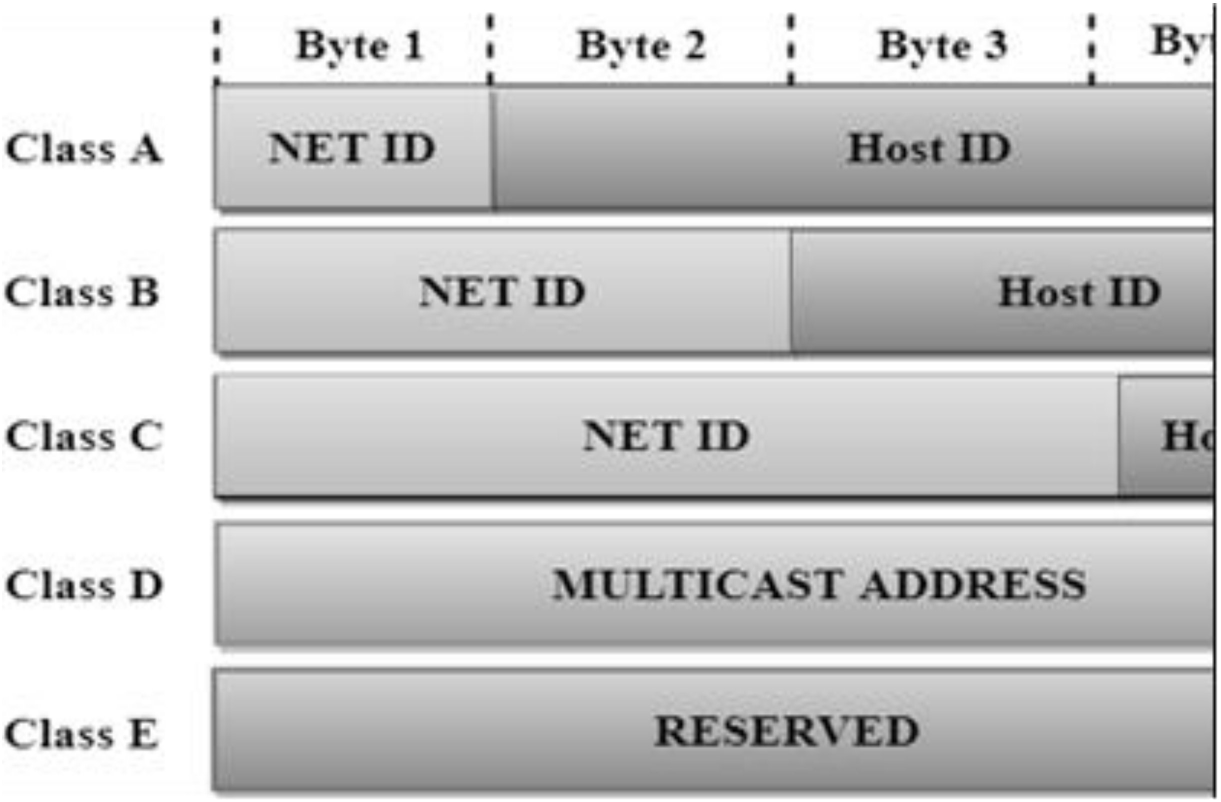
CATEGORIES OF IPV4 ADDRESSING

- 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:

CLASSFUL ADDRESSING



Byte 1 | Byte 2 | Byte 3 | By

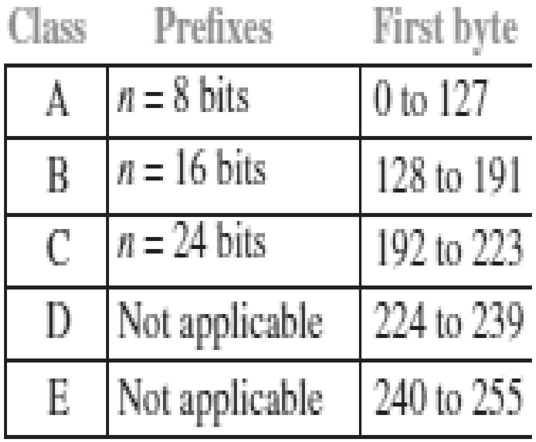
Class A: NET ID | Host ID

Class B: NET ID | Host ID

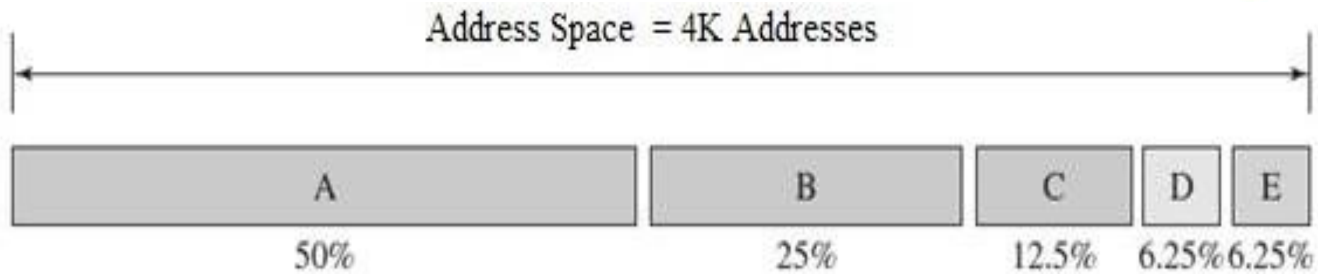
Class C: NET ID | Ho

Class D: MULTICAST ADDRESS

Class E: RESERVED



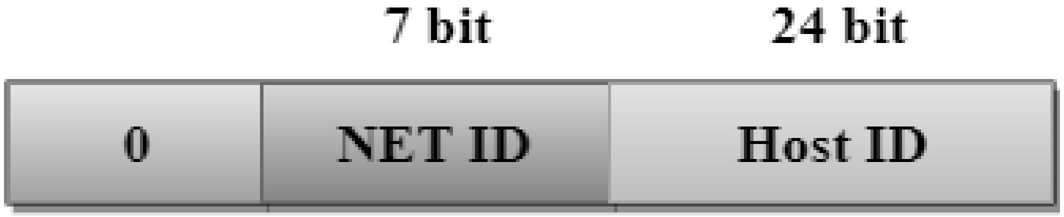
Class	Prefixes	First byte
A	n = 8 bits	0 to 127
B	n = 16 bits	128 to 191
C	n = 24 bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255



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

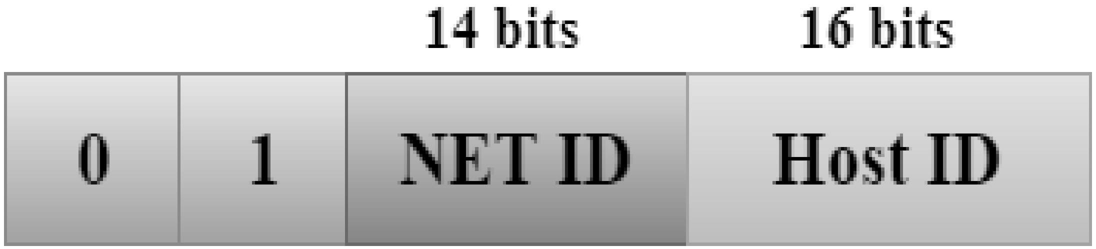
- In Class A, an IP address is **assigned to those networks that contain a large number of hosts.**
- The **network ID is 8 bits** long. The **host ID is 24 bits** long.
- In Class A, the **first bit in higher order bits of the first octet is always set to 0** and the **remaining 7 bits determine the 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



	7 bit	24 bit
0	NET ID	Host ID

Class B

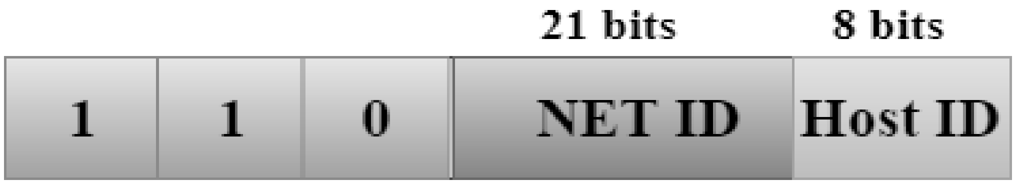
- In Class B, an IP address is assigned to those networks that range from small- sized to large-sized networks.
- The **Network ID is 16 bits** long. The **Host ID is 16 bits** long.
- In Class B, the higher order bits of the first octet is always set to 10, and the **remaining 14 bits determine the 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



		14 bits	16 bits
0	1	NET ID	Host ID

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



21 bits 8 bits

1 1 0 NET ID Host ID

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.

				28 bits
1	1	1	0	Host ID

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.**

				28 bits
1	1	1	1	Host ID

Address Depletion in Classful Addressing

- The reason that **classful addressing has become obsolete is address depletion.**
- 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 using a block in this address class.
- **Class E addresses were almost never used, wasting the whole class**

Advantage of Classful Addressing

- 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 ii) Supernetting.

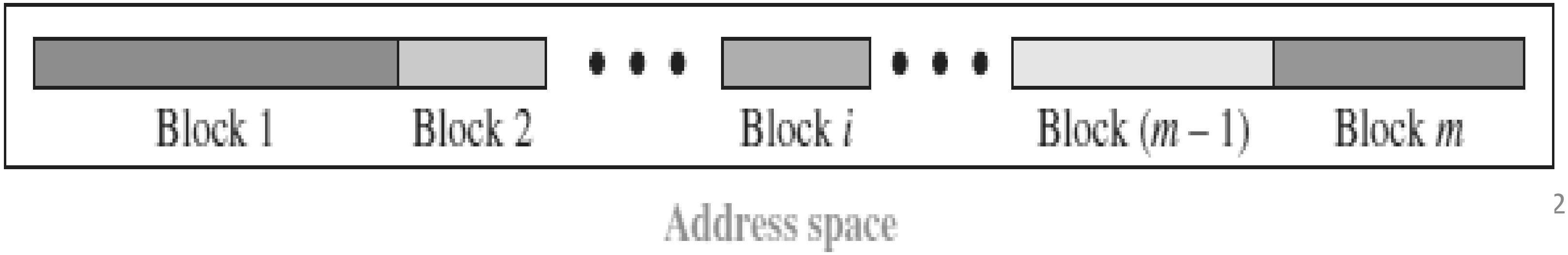
Subnetting

- 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

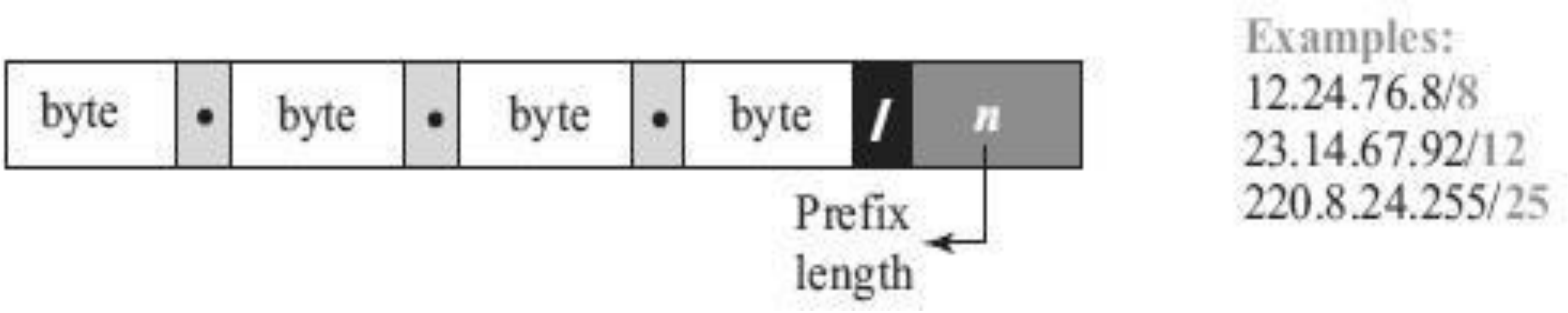
- In 1996, the Internet authorities announced a new architecture called **classless addressing.**
- In classless addressing, **variable-length blocks are used that belong to no classes**.
- We can have a block of 1 address, 2 addresses, 4 addresses, and so on.
- In classless addressing, **the whole address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

- 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** **(in terms of number of hosts it can accommodate)**
- 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 (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.



Examples:
12.24.76.8/8
23.14.67.92/12
220.8.24.255/25

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

- Special addresses that are used for special purposes:

i) ***this-host*** **address,**

ii) ***limited-broadcast*****address,**

iii) ***loopback*** **address,**

iv) ***private*** **addresses, and**

v) ***multicast*** **addresses.**

i) 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.

ii) 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.**

iii) 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.**

iv) 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**.

v) Multicast Addresses

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

NETWORK LAYER PROTOCOLS

IP, ICMPv4

NETWORK LAYER PROTOCOLS:
IP, ICMPv4

- The main protocol **IP is responsible for packetizing, forwarding, and delivery of a packet** at the network layer.

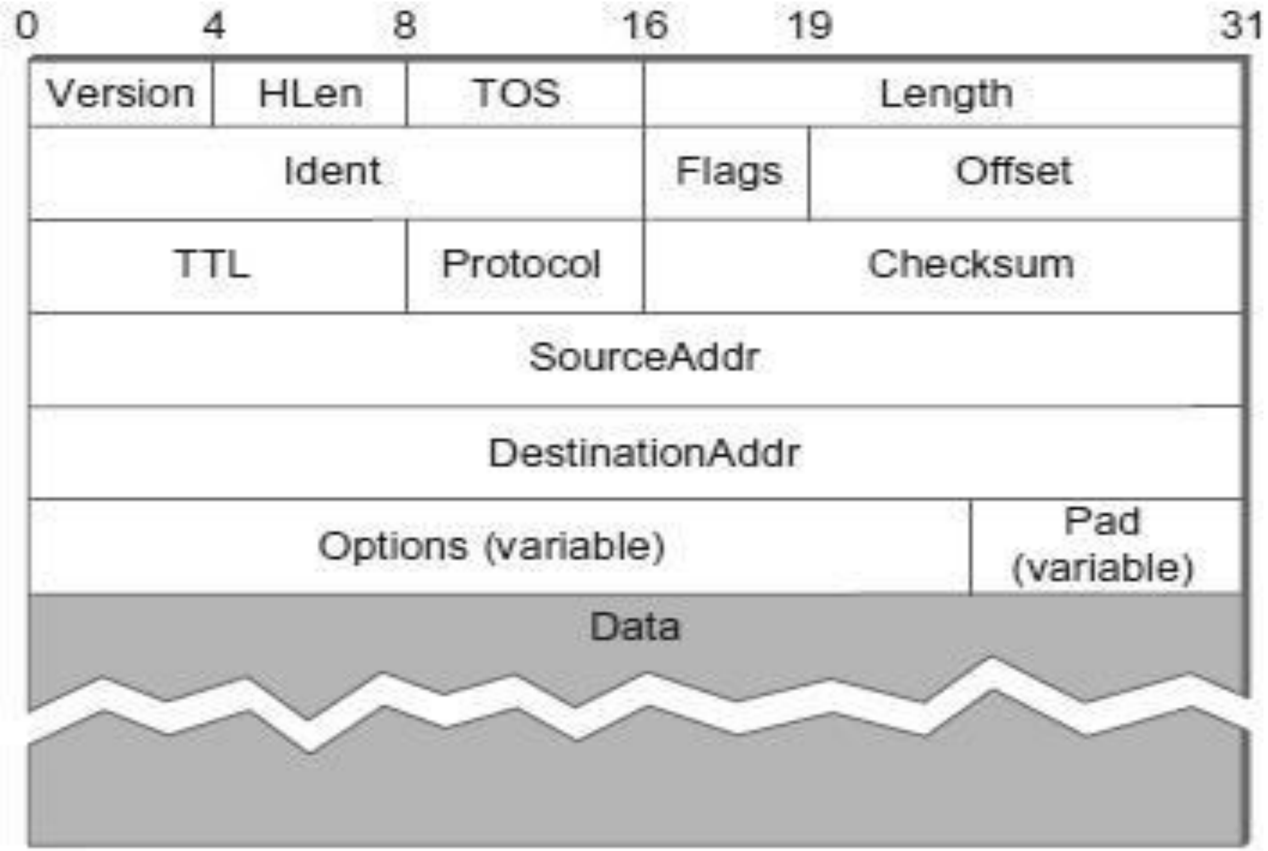
- The **Internet Control Message Protocol version 4 (ICMPv4) helps IPv4 to handle some errors** that may occur in the network-layer delivery.

IP - INTERNET PROTOCOL

- IP is used **to build a 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
  - 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.**

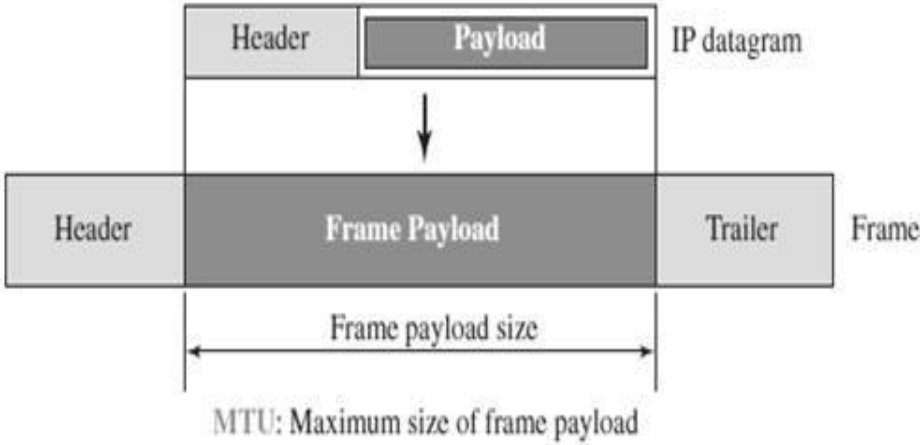


FIELD	DESCRIPTION
Version	Specifies the version of IP. Two versions exists – **IPv4 and IPv6**.
HLen	Specifies the length of the header
TOS *(*Type of Service)	An indication of the parameters of the quality of service desired such as **Precedence, Delay, Throughput and Reliability.**
Length	Length of 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 (**TCP or UDP**)
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.**

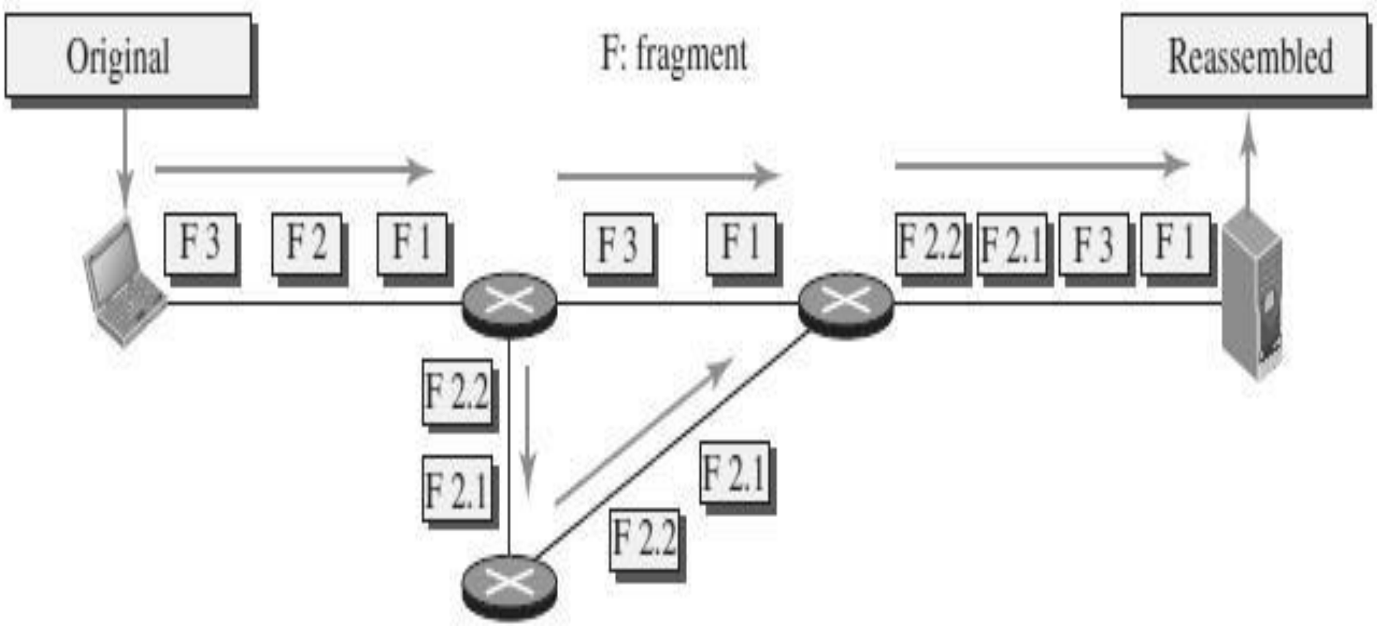
IP DATAGRAM - FRAGMENTATION AND REASSEMBLY

- **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.

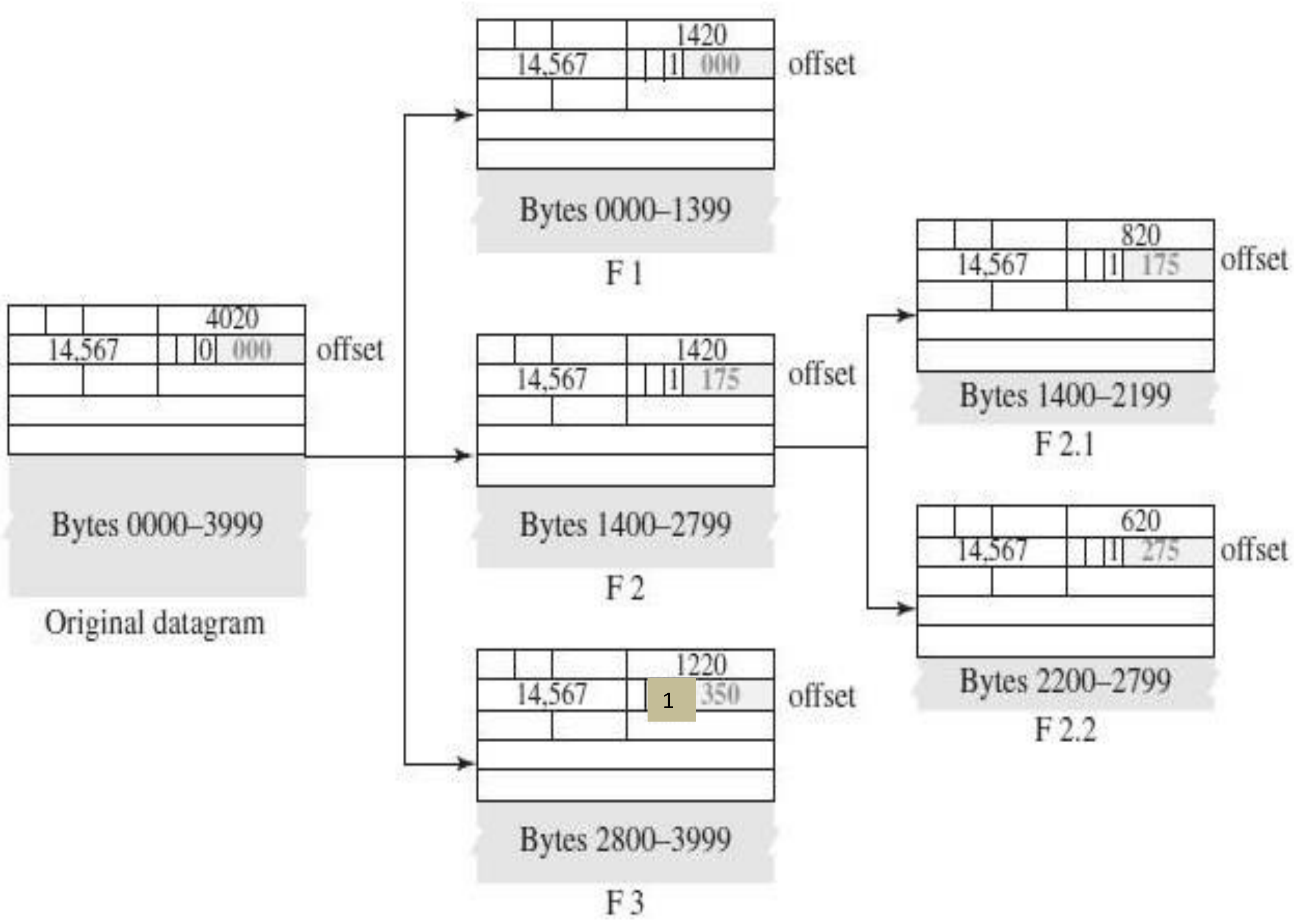


MTU: Maximum size of frame payload

- 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.**



- 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.


4020
14,567
0
000
offset
Bytes 0000–3999
Original datagram
1420
14,567
1
000
offset
Bytes 0000–1399
F 1
1420
14,567
1
175
offset
Bytes 1400–2799
F 2
1220
14,567
1
350
offset
Bytes 2800–3999
F 3
820
14,567
1
175
offset
Bytes 1400–2199
F 2.1
620
14,567
1
275
offset
Bytes 2200–2799
F 2.2

  - 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:

- The first fragment has an offset field value of zero.
- Divide the length of the first fragment by 8. The second fragment has an offset value equal to that result.
- Divide the total length of the first and second fragment by 8. The third fragment has an offset value equal to that result.
- Continue the process. The last fragment has its M bit set to 0.
- Continue the process. The last fragment has a *more* bit value of 0.

Reassembly

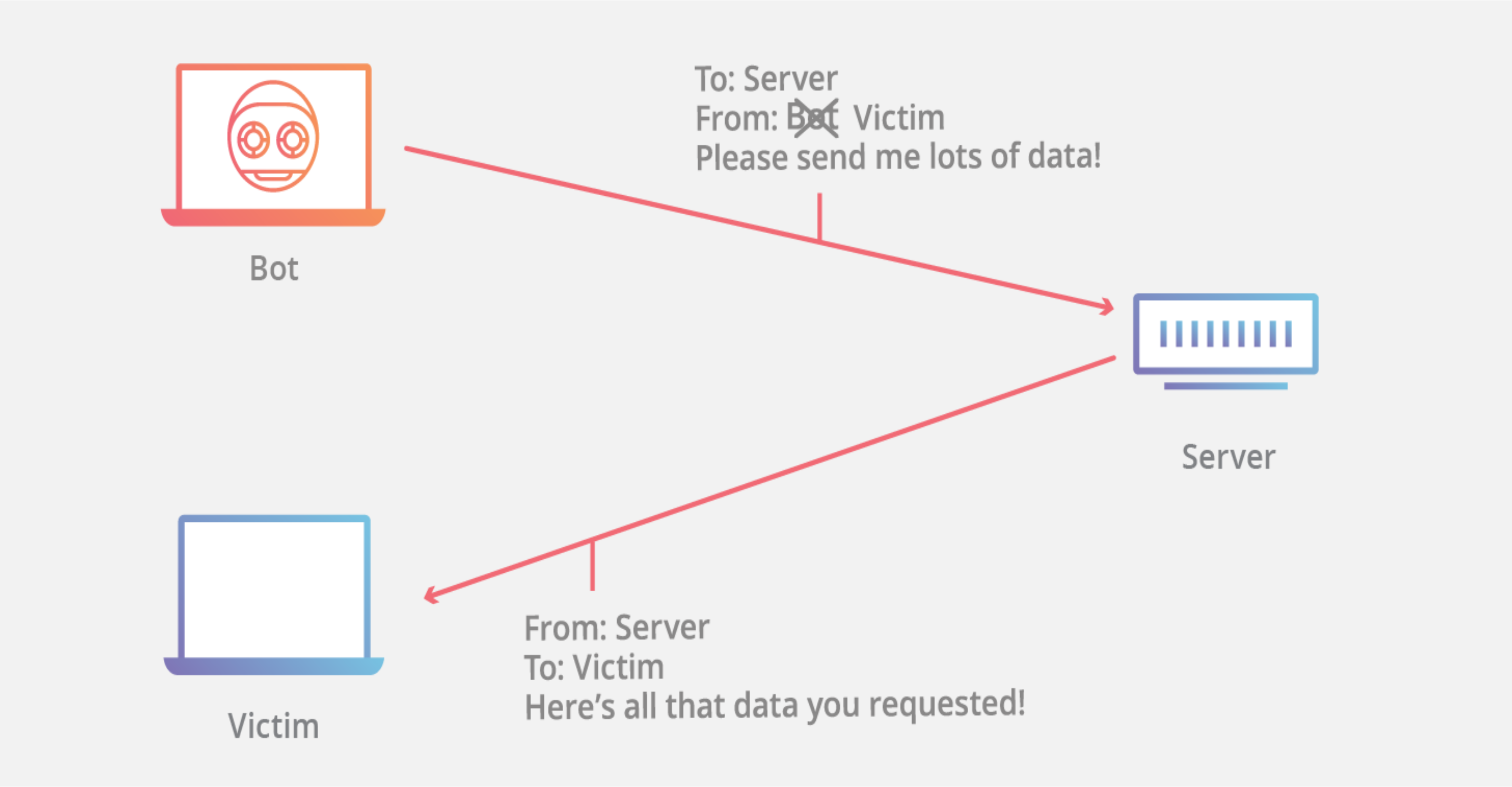
- Reassembly is done at the receiving host and not at each router.
- To enable these fragments to be reassembled at the receiving host, they all carry the **same identifier in the Ident field.**
- This identifier is chosen by the sending host and is intended to be unique among all the datagrams that 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 Spoofing

IP Spoofing

- IP spoofing is the **creation of Internet Protocol (IP) packets which have a modified source address** in order **to either hide the identity of the sender, or to impersonate another** computer system, or both.
- It is a technique often **used by bad actors to invoke** DDoS attacks against a target device or the surrounding infrastructure.
- All IP packets contain a **header** which precedes the body of the packet and **contains important routing information, including the source address.**
- In a normal packet, the source IP address is the address of the sender of the packet.
- **If the packet has been spoofed**, the source address will be forged.

IP Spoofing



To: Server
From: ~~Bot~~ Victim
Please send me lots of data!

From: Server
To: Victim
Here's all that data you requested!

IP Spoofing

- IP Spoofing is analogous to **an attacker sending a package to someone with the wrong return address** listed.
- The ability to spoof the addresses of packets is a core vulnerability **exploited by many DDoS attacks.**
- DDoS attacks will often **utilize spoofing with a goal of overwhelming a target with traffic** while masking the identity of the malicious source, **preventing mitigation efforts.**
- If the **source IP address is falsified and continuously randomized**, blocking malicious requests becomes difficult.
- IP spoofing also **makes it tough for law enforcement and cyber security teams to track down** the perpetrator of the attack.

How to protect against IP spoofing (Packet Filtering)

- While IP spoofing **can't be prevented, measures can be taken to stop spoofed packets** from infiltrating a network.
- A very common **defense against spoofing is ingress filtering.**
- Ingress filtering is a form of packet filtering usually implemented on a network edge device which **examines incoming IP packets and looks at their source headers.**
- If the **source headers on those packets don't match their origin** or they otherwise look fishy, the **packets are rejected.**
- Some networks will also implement **egress filtering**, which **looks at IP packets exiting the network**, ensuring that those packets have legitimate source headers to **prevent** someone within the network **from launching an outbound malicious attack** using IP spoofing.

ICMPv4 - INTERNET CONTROL MESSAGE PROTOCOL VERSION 4

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**.

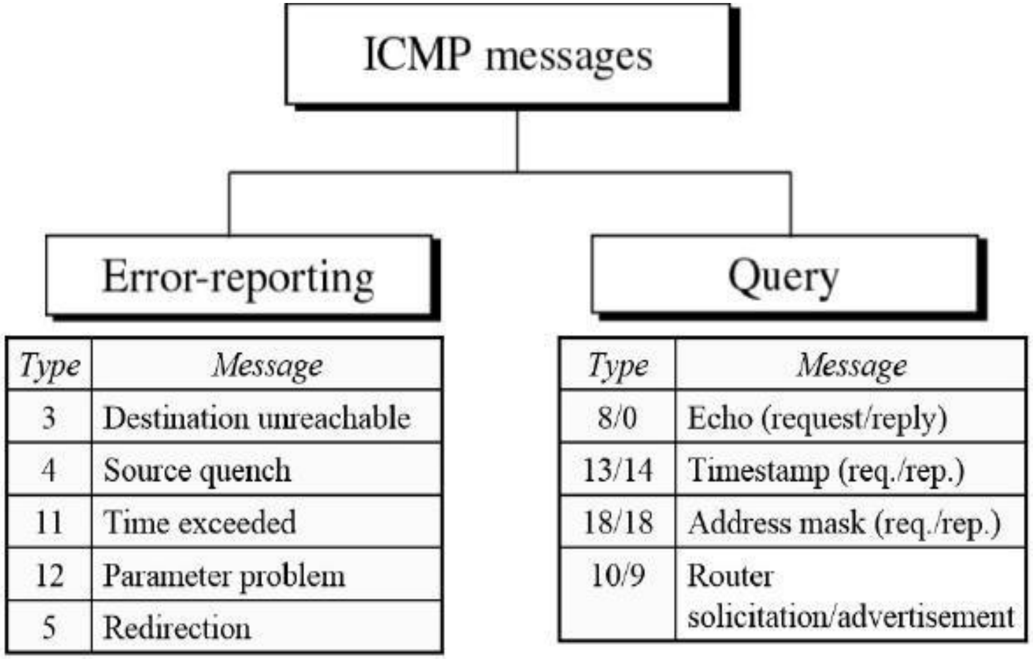
ICMP MESSAGE TYPES

- ICMP messages are divided into **two broad categories**:

  i) error-reporting messages and **ii)** ***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.

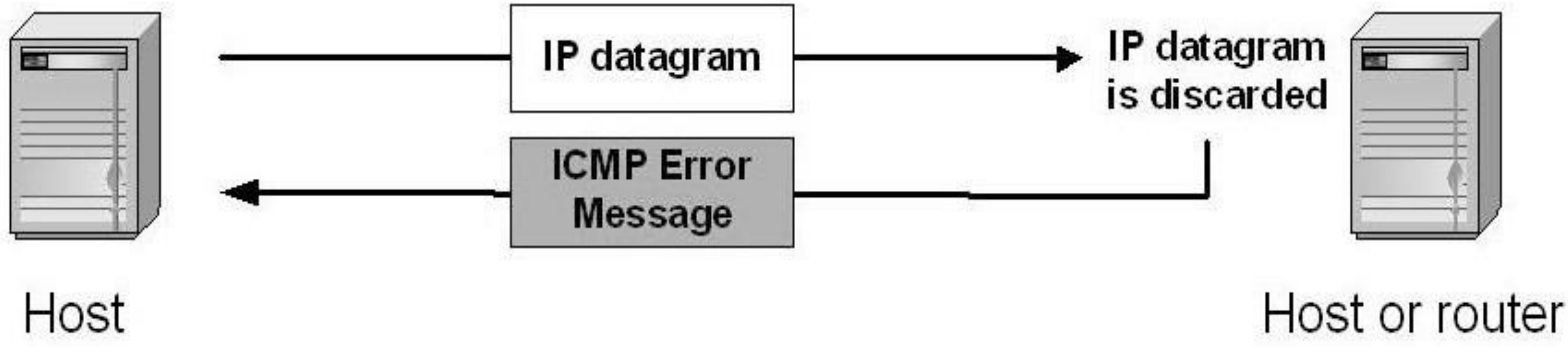


Type	*Message*
3	Destination unreachable
4	Source quench
11	Time exceeded
12	Parameter problem
5	Redirection

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

- ***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 (buffer overload)*, 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.**

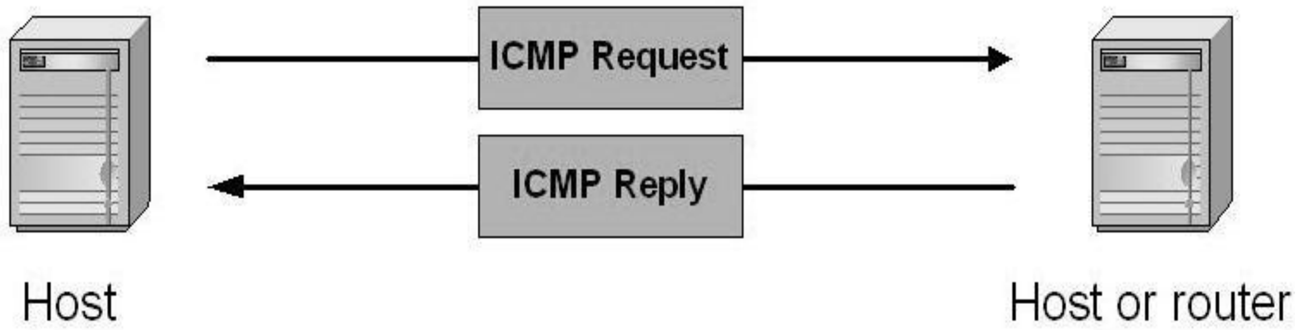

IP datagram
IP datagram is discarded
ICMP Error Message
Host
Host or router

ICMP Query Messages

→ **Request sent by host to a router or host**

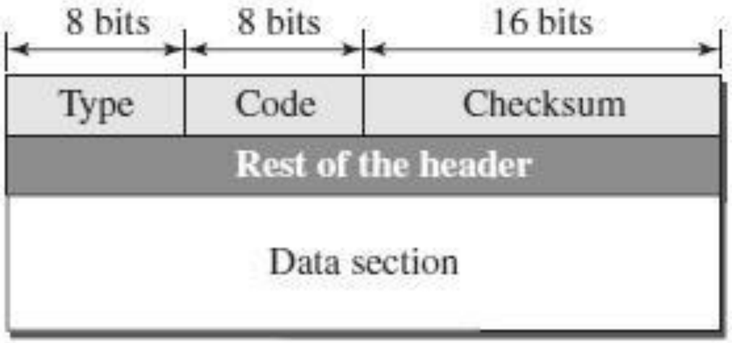
→ **Reply sent back to querying node**

- ***Echo Request & Reply***—Combination of echo request and reply messages **determines whether two systems can 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.**

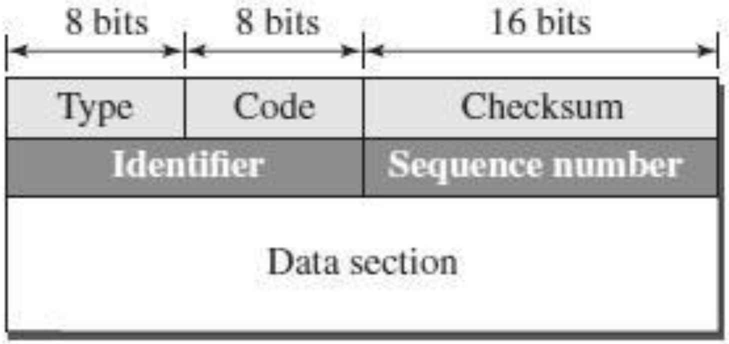


ICMP MESSAGE FORMAT

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



Error-reporting messages



Query messages

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

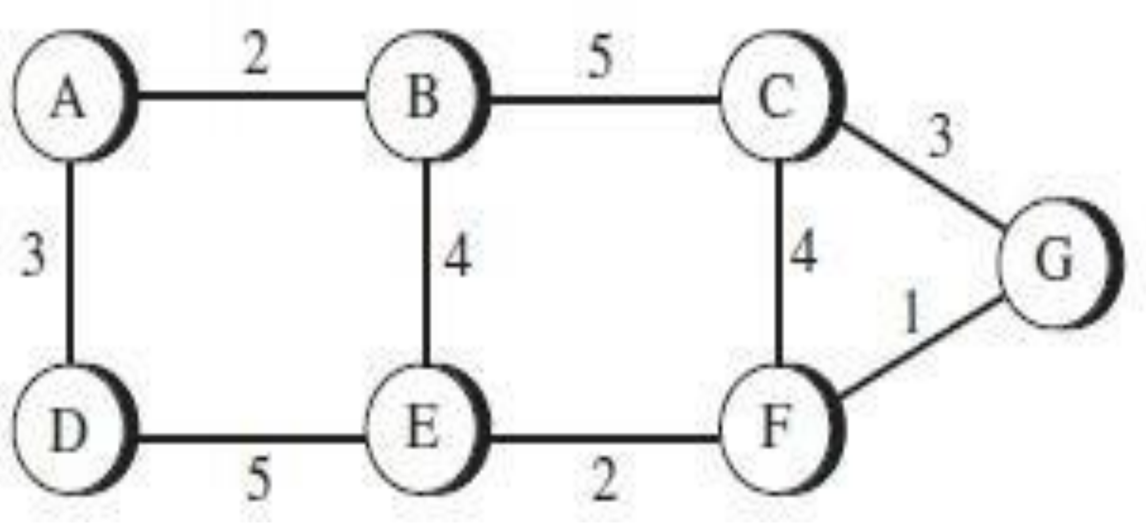
UNICAST ROUTING

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.



Static Vs. Dynamic

– 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, Dynamic 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:

- **Distance Vector Routing Algorithm – i) Bellman - Ford Algorithm & Routing Information Protocol (RIP)**
- **Link State Routing Algorithm – Open Shortest Path First Protocol (OSPF)**
- **Path-Vector Routing Algorithm - Border Gateway Protocol (BGP)**

DISTANCE VECTOR ROUTING (DSR)

i) BELLMAN - FORD ALGORITHM

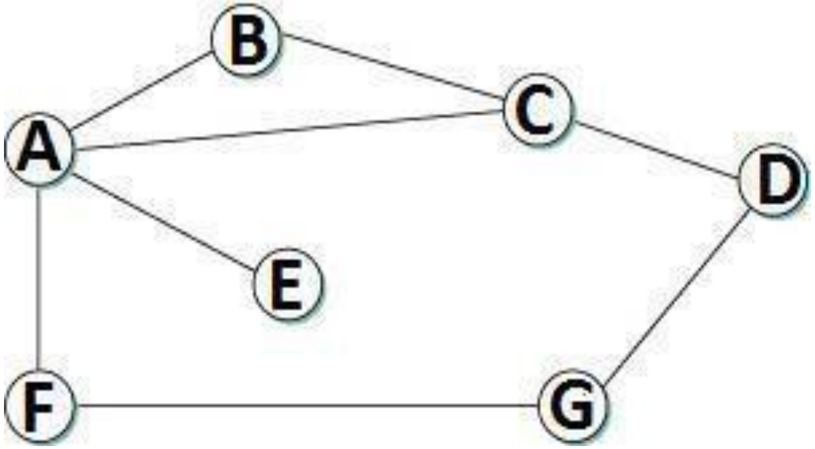
ii) ROUTING INFORMATION PROTOCOL (RIP)

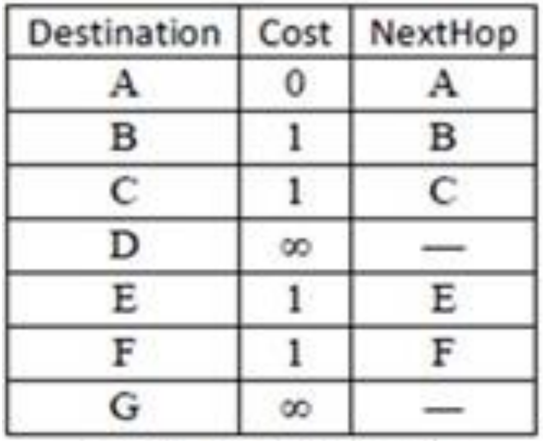
BELLMAN - FORD ALGORITHM

- Distance vector routing is ***distributed***, i.e., **algorithm runs 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

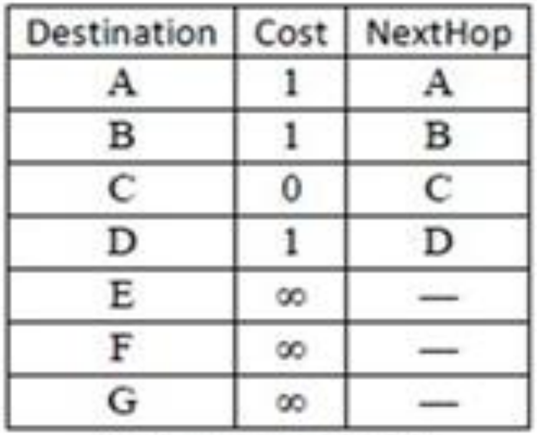
- In given network, ***cost* of each link is 1 hop.**
- Each node sets a **distance of 1 (hop) to its *immediate* neighbor** and **cost to itself as 0.**
- **Distance for non-neighbors** is **marked as *unreachable*** with value **∞ (infinity).**
- For node *A*, nodes *B*, *C*, *E* and *F* are *reachable*,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





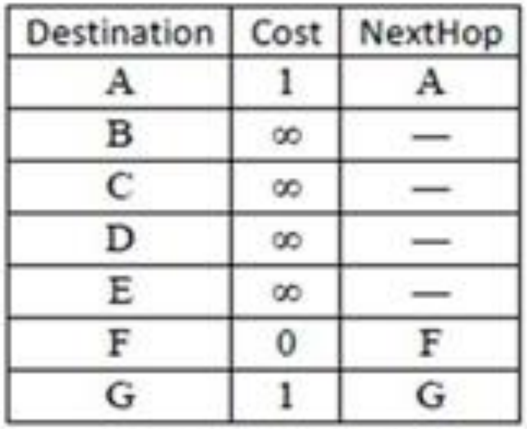
Destination	Cost	NextHop
A	0	A
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

Node A's initial table



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

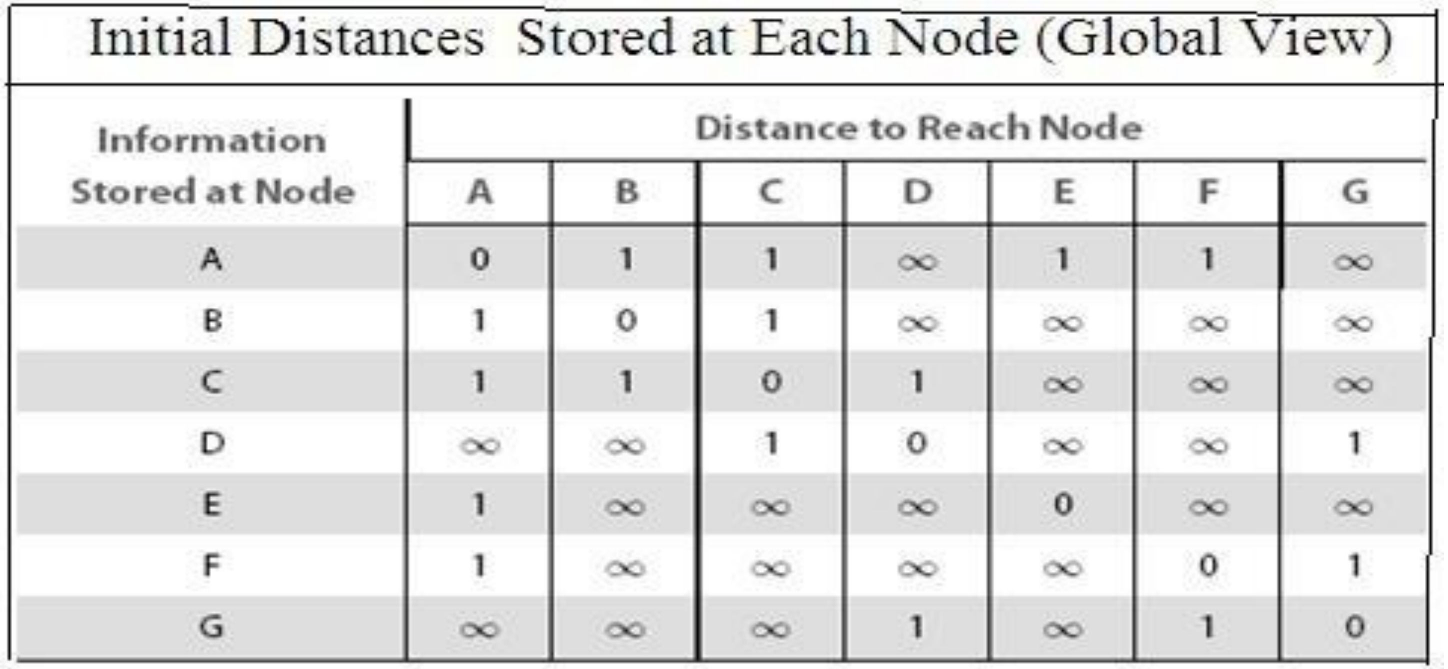
Node C's initial table



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*.

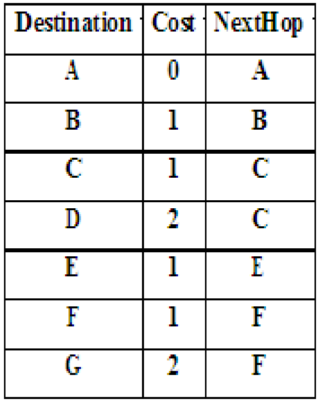
➔ Total Cost to reach node *D* via *C* = Cost (*A* to *C*) + Cost(*C* to *D*)

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.

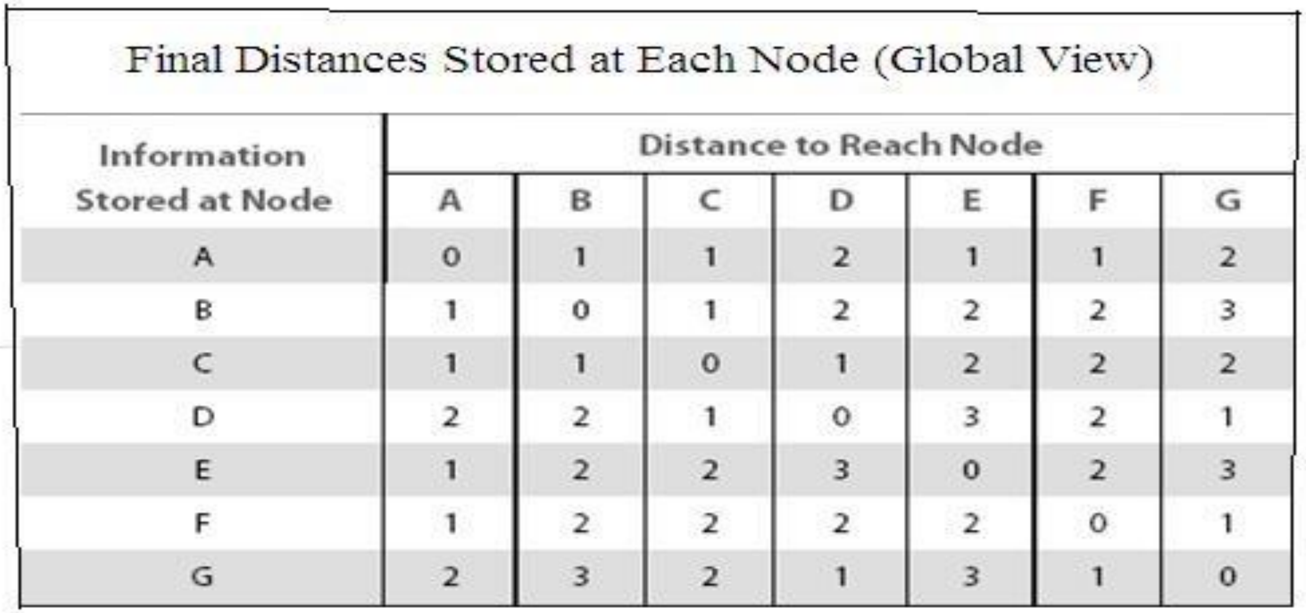
Node A's final routing table



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.**

i) Periodic Update

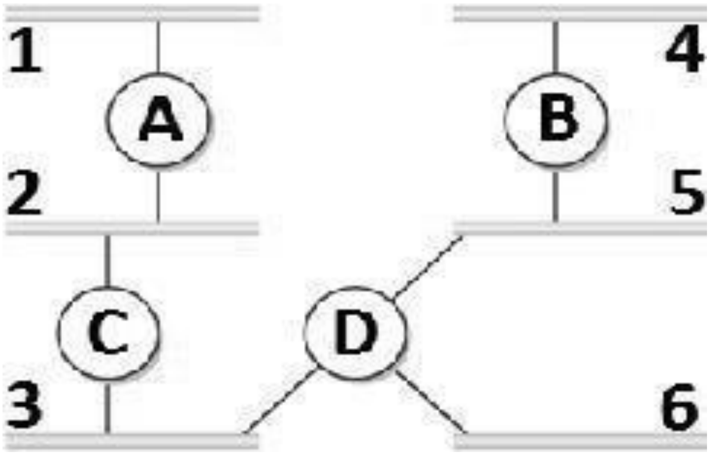
- In this case, each node automatically sends an update message every so often, even if nothing has changed.
- The frequency of these periodic updates varies from protocol to protocol, but it is typically on the order of several seconds to several minutes.

ii) 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.

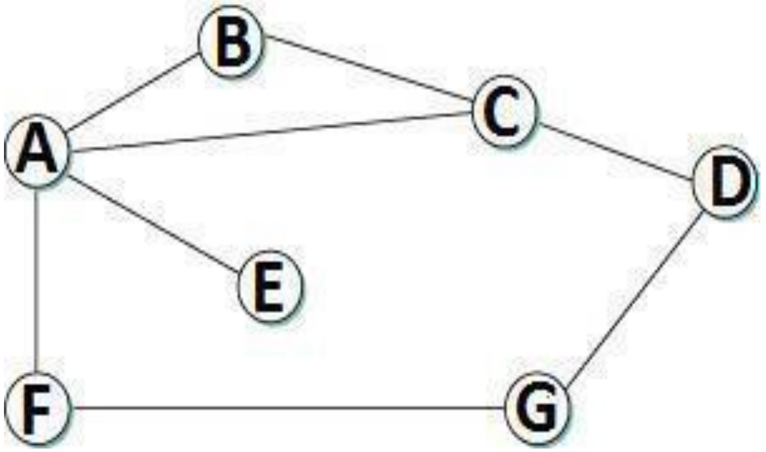
ROUTING INFORMATION PROTOCOL (RIP)

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



Count-To-Infinity (or) Loop Instability Problem

- 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)

LINK STATE ROUTING (LSR)

- Each node **knows *state* of link to its neighbors and *cost*.**

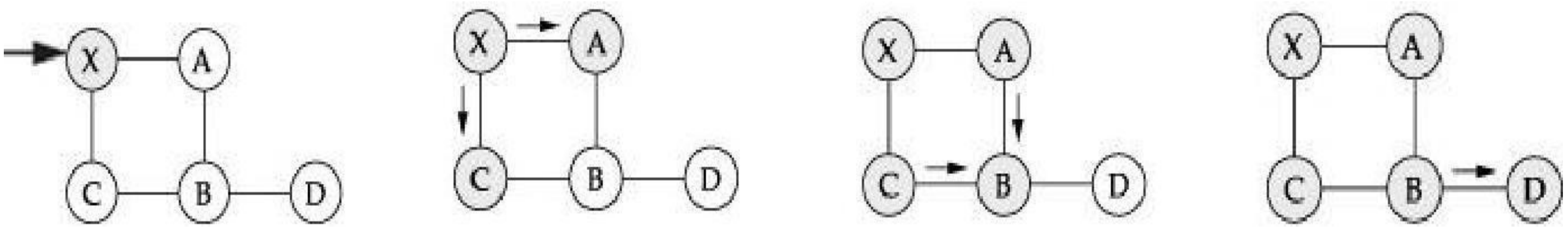
- **Nodes create an update packet called *link-state packet* (LSP)** that contains:
  - **ID of the node**
  - **List of neighbors for that node and associated cost**
  - **64-bit Sequence number**
  - **Time to live (TTL)**

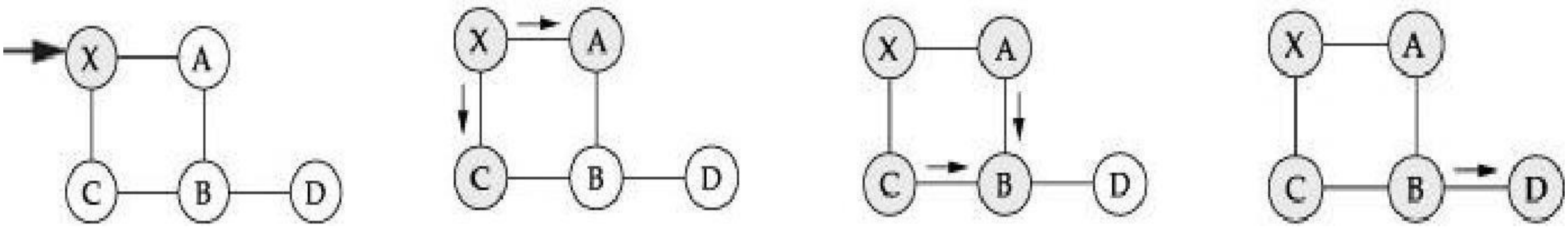
- **Link-State routing protocols rely on two mechanisms:**
  - ***Reliable flooding*** of link-state information to all other nodes
  - ***Route calculation*** from the accumulated link-state knowledge

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*.**





- Flooding of LSP in a small network is as follows:
  - When node *X* receives *Y*'s LSP , it floods onto its neighbors *A and C*
  - **Nodes *A* and *C* forward it to *B*, but does not sends it back to *X*.**
  - **Node *B* receives two copies of LSP with same sequence number.**
  - **Accepts one LSP and forwards it to *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.

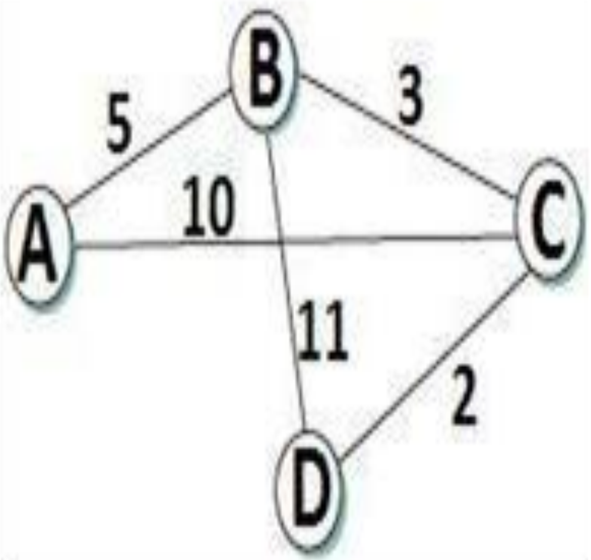
- **Forward search algorithm is used to compute routing table** from the received LSPs.

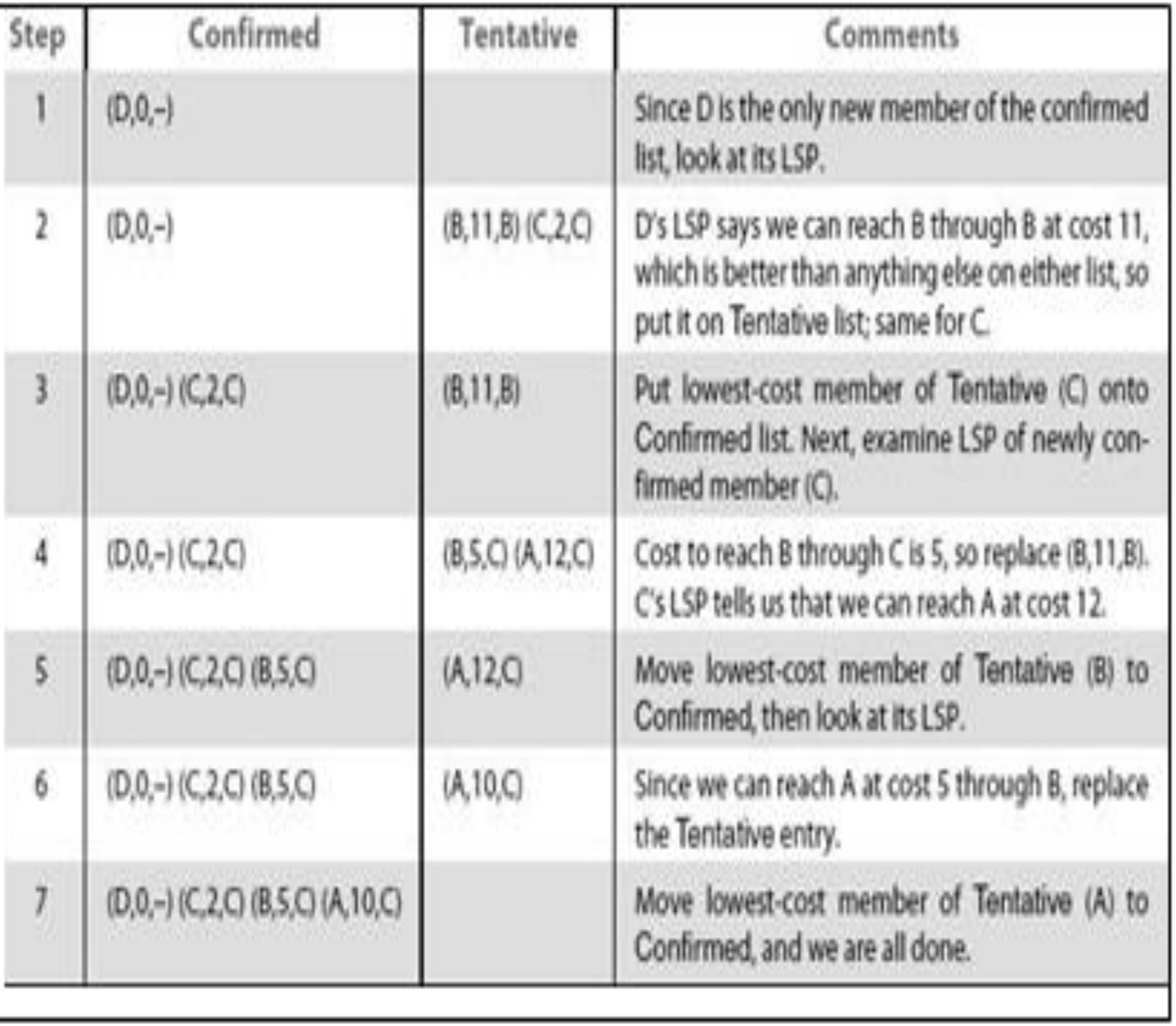
- **Each node maintains two lists, namely Tentative** and **Confirmed with entries of the form (Destination, Cost, NextHop).**

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).**
   - If Neighbor is neither in Confirmed nor in Tentative list, then add (Neighbor, Cost, NextHop) to Tentative list.
   - **If Neighbor is in Tentative list,** and Cost is less than existing cost, then replace the entry with (Neighbor, Cost, NextHop).
5. **If Tentative list is empty then *Stop*,** otherwise move *least* cost entry from Tentative list to Confirmed list. Go to *Step 2*.

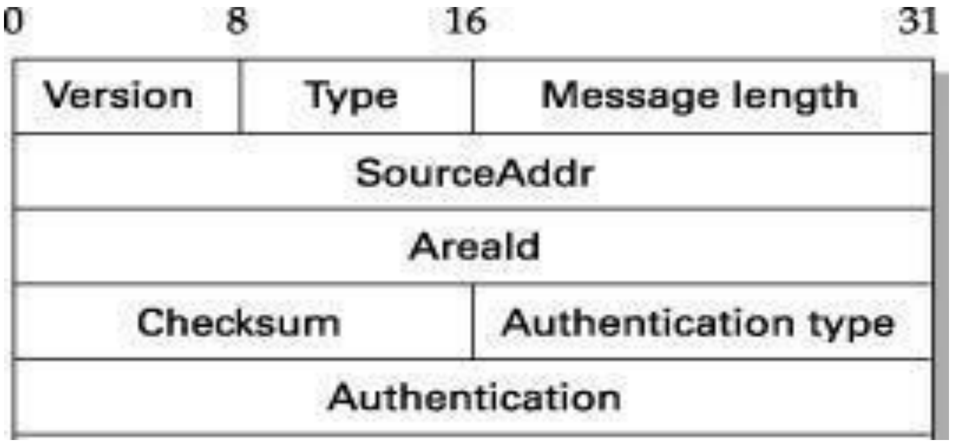
Example





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.

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.

Type 1 **-** "hello" message, *Type 2* **-** request, *Type 3* – send ,

Type 4 **-** acknowledge the receipt of link state messages ,

Type 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)

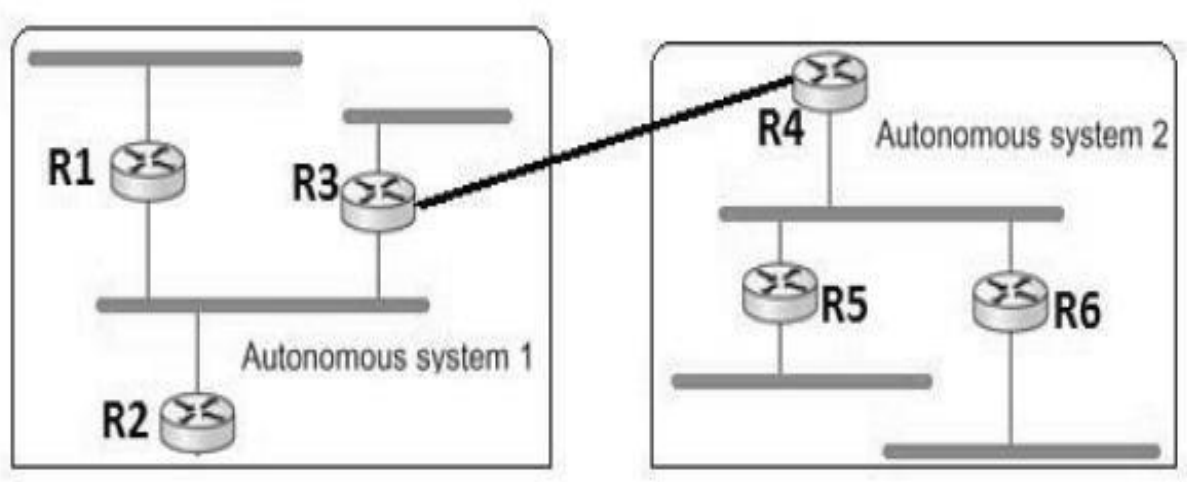
BORDER GATEWAY PROTOCOL (BGP)

- 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.**
- The **Border Gateway Protocol version** (BGP) is the **only interdomain routing protocol used today.**
- It **provides information about the reachability of networks** in the Internet.

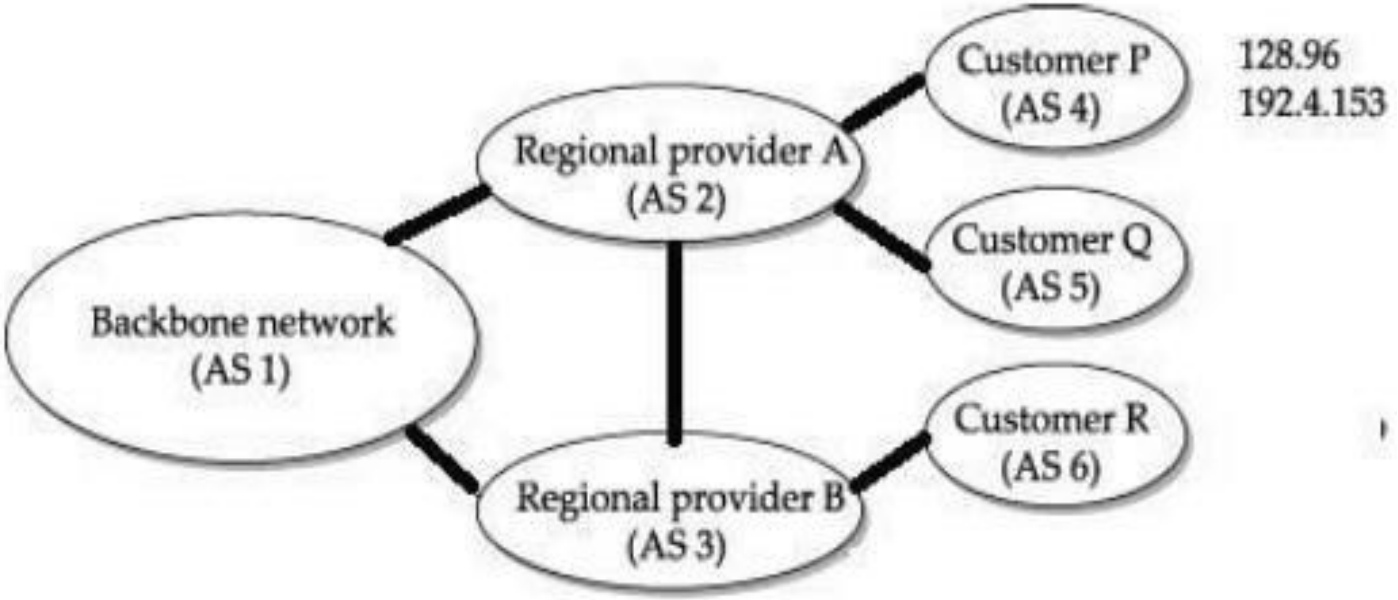
BORDER GATEWAY PROTOCOL (BGP)

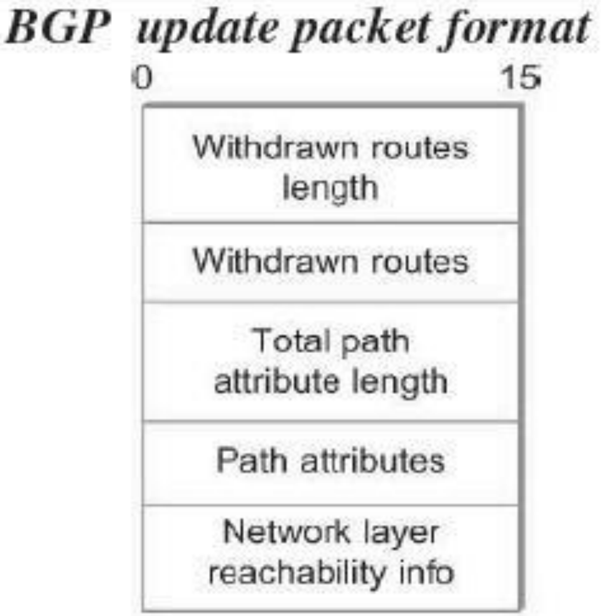
- **BGP views internet as a set of autonomous systems interconnected arbitrarily**.
- **Each AS have a *border router* (gateway), by which packets enter and leave that AS.**
- **One of the router** in **each autonomous system is designated as BGP *speaker*.**
- **BGP Speaker *exchange* reachability information 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.**

R3 and *R4* are border routers.



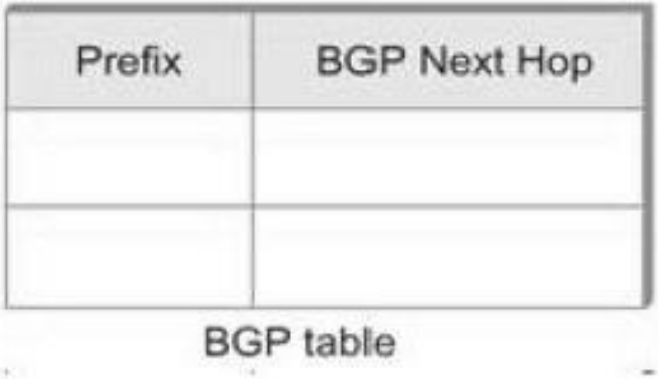
- **Paths must be without any *loop,* i.e., AS list is unique.**
- **For *example***, backbone network advertises that networks 128.96 and 192.4.153 **can be reached along the path** <*AS1*, *AS2, AS4*>.
- If there are *multiple* routes to a destination, **BGP speaker chooses one based on policy.**
- **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**.





BGP update packet format

0 15
Withdrawn routes length
Withdrawn routes
Total path attribute length
Path attributes
Network layer reachability info



Prefix	BGP Next Hop

BGP table

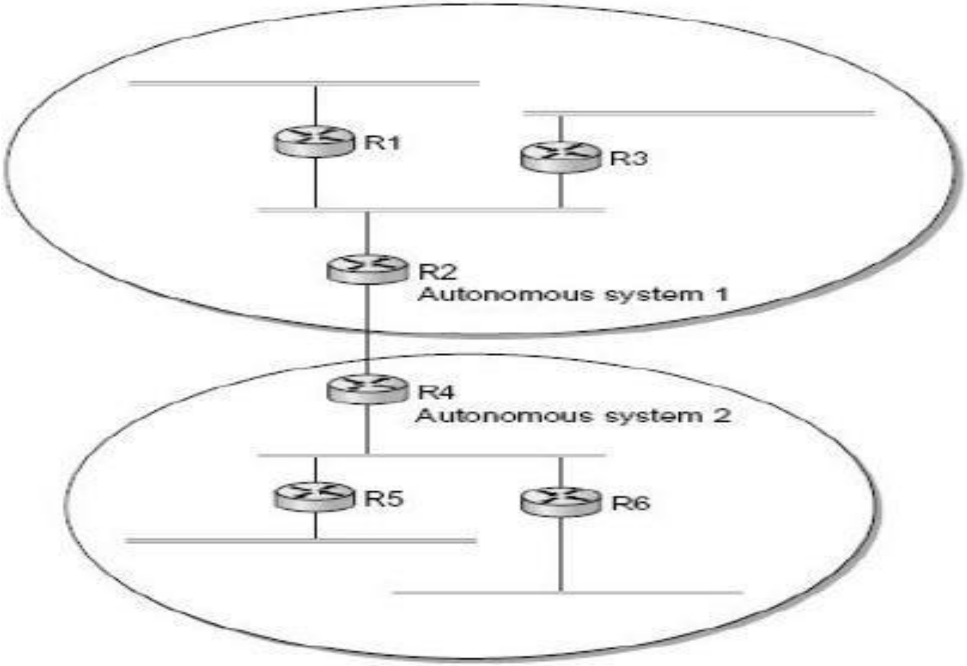
iBGP - interior BGP

A Variant of BGP

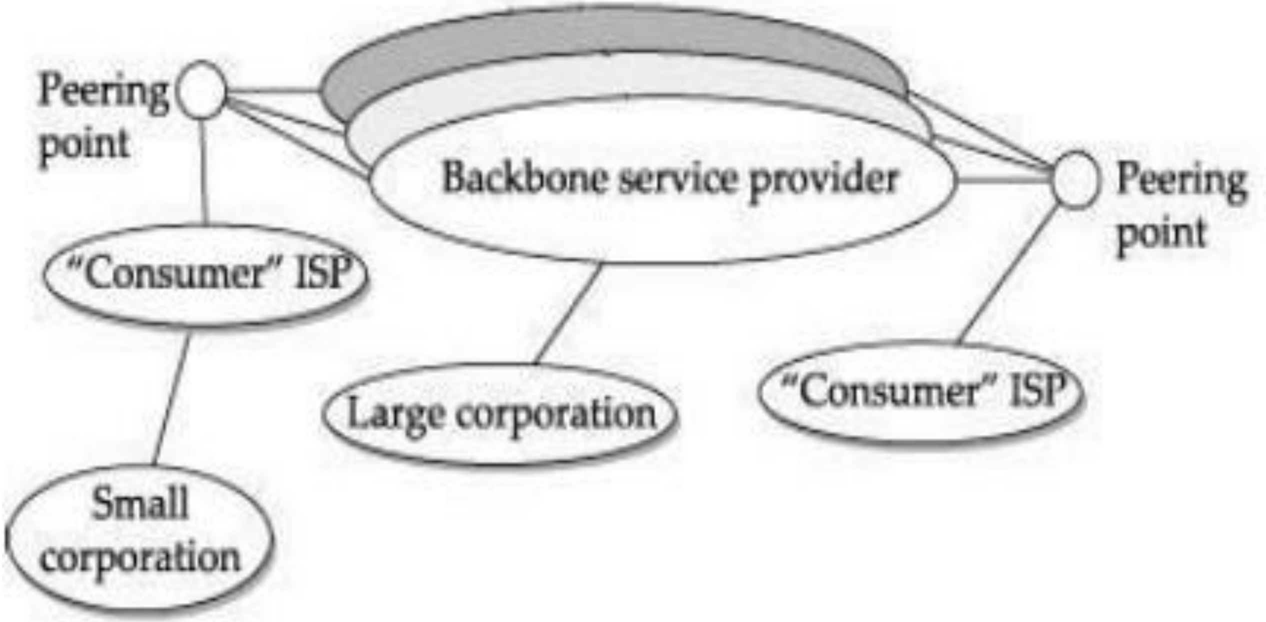
- **Used by routers to update routing information** **learnt from other speakers** to routers inside the autonomous system.
- Each router in the AS is able to determine the appropriate next hop for all prefixes.

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 be 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.**



- **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 AS and carries local traffic only** (e.g. Small corporation).
  - ***Multihomed AS*** - has **connections to multiple AS but refuses to carry transit traffic** (e.g. Large corporation).
  - ***Transit AS*** - has **connections to multiple AS 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.

CONGESTION CONTROL

CONGESTION CONTROL

Congestion at the network layer is **related to two issues,**

➔ **throughput**

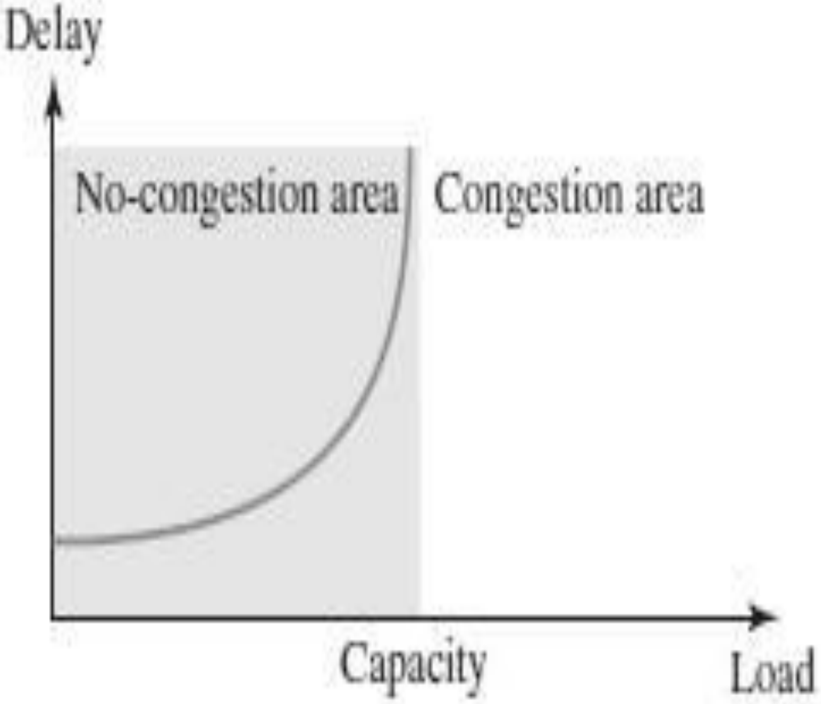
➔ **delay**

Based on Delay

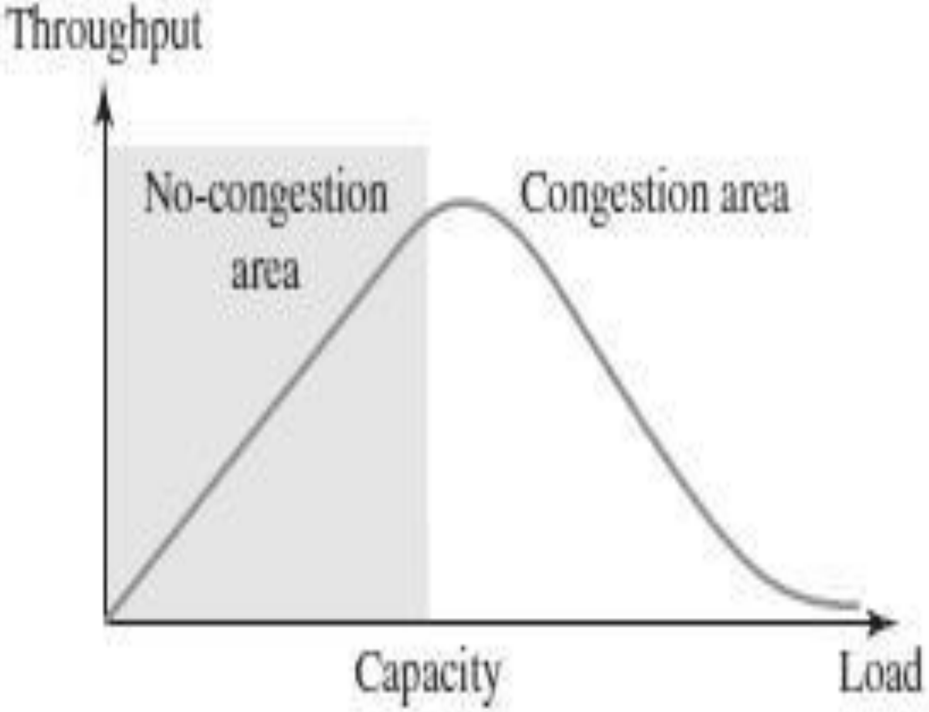
- When the **load is much less than the capacity of the network**, the delay is at a minimum.
- This minimum delay is **composed of propagation delay and processing delay**, both of which are negligible.
- However, **when the load reaches the network capacity, the delay increases** sharply because we 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 Throughput

- 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

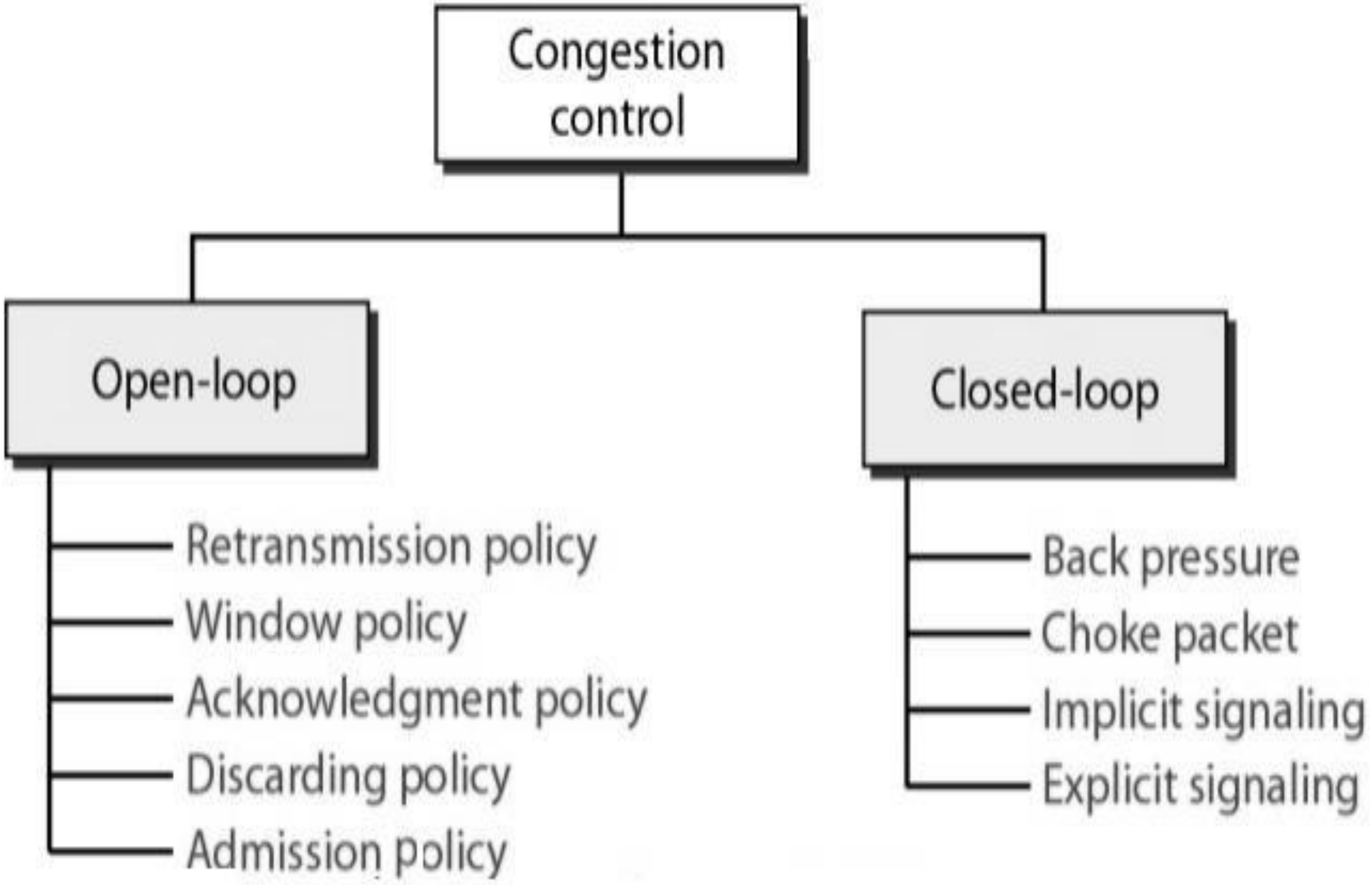
- Congestion control is a mechanism for improving performance.
- It refers to **techniques and mechanisms that can either** **prevent congestion** before it happens

  or

  remove congestion after it has happened.
- In general, we can divide congestion control mechanisms into **two broad categories:**

  i) Open-loop Congestion control (prevention)

  ii) Closed-loop Congestion control (removal)



Congestion control

Open-loop
- Retransmission policy
- Window policy
- Acknowledgment policy
- Discarding policy
- Admission policy

Closed-loop
- Back pressure
- Choke packet
- Implicit signaling
- Explicit signaling

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**.

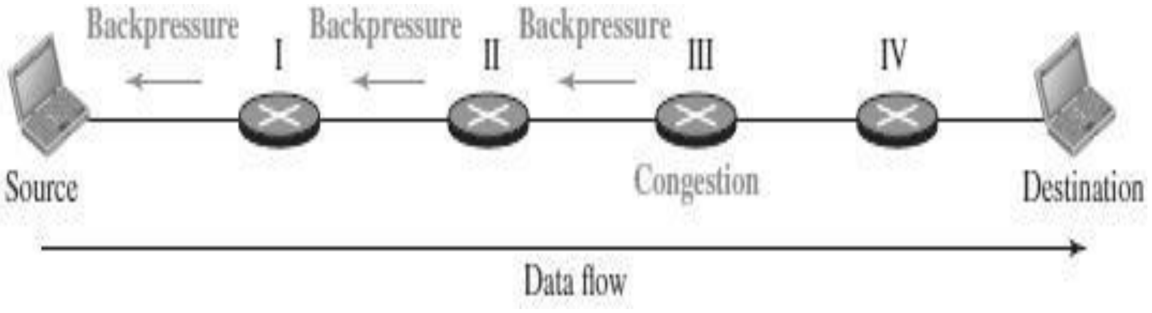
- **Window Policy**
  - The type of window at the sender may also affect congestion.
  - The **Selective Repeat window is better than the Go-Back-N** window for congestion control.
  - In the **Go-Back-N window**, **when the timer for a packet times out, several packets may be resent**, although some 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.

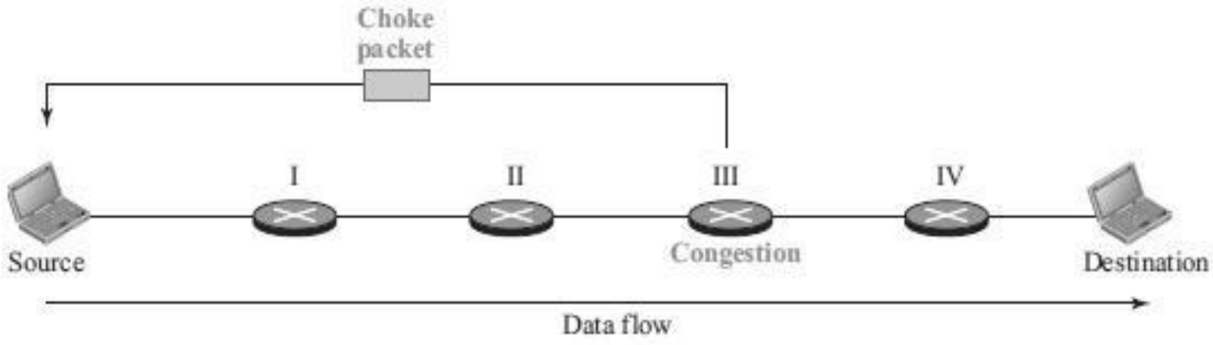
CLOSED-LOOP CONGESTION CONTROL

- Closed-loop congestion control mechanisms try to **alleviate congestion after it happens.**
- **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**.

IPv6 - NEXT GENERATION IP

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

- ***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.**

- ***New options*** **-** IPv6 has **new options to allow for additional functionalities**.

  ➔ ***Allowance for extension*** **-** IPv6 is **designed to allow the extension of the protocol if required** by new technologies or applications.

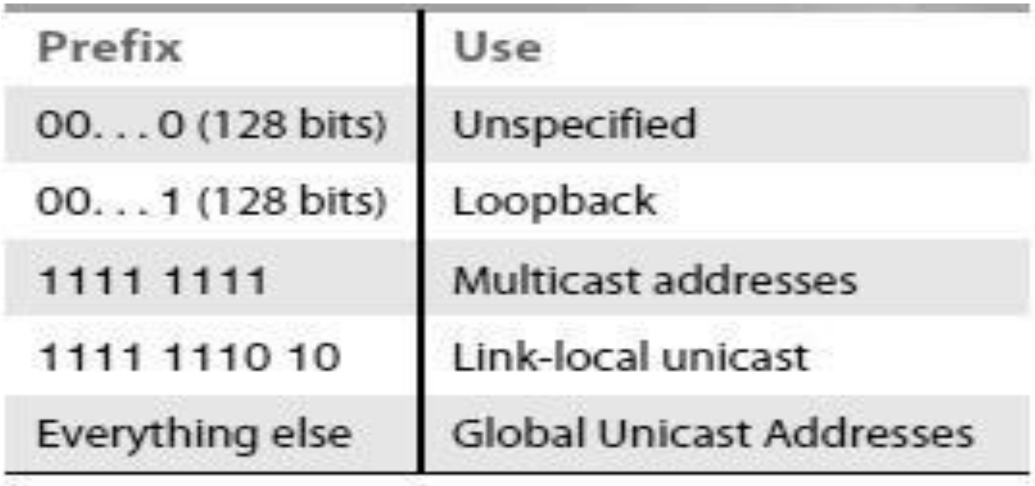
Additional Features:

- Need to accommodate **scalable routing and addressing**
- **Support for real-time services**
- **Security support**
- **Auto-configuration**

- The **ability of hosts to automatically configure themselves** with such information as their **own IP address and domain name.**
- **Enhanced routing functionality, including support for mobile hosts**
- ***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.**

ADDRESS SPACE ALLOCATION OF IPv6

- IPv6 provides a **128-bit address space (ie., 2^{128})**
- IPv6 **uses *classless* addressing**
- A node may be assigned an **"IPv4-compatible IPv6 address" by zero-extending a 32-bit IPv4 address to 128 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.



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

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.**
  - o ***Unspecified address*** is used when a host does not know its address
  - o ***Loopback address*** is used **for testing purposes** before connecting
  - o ***Compatible address*** is used when IPv6 hosts uses IPv4 network
  - o 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,

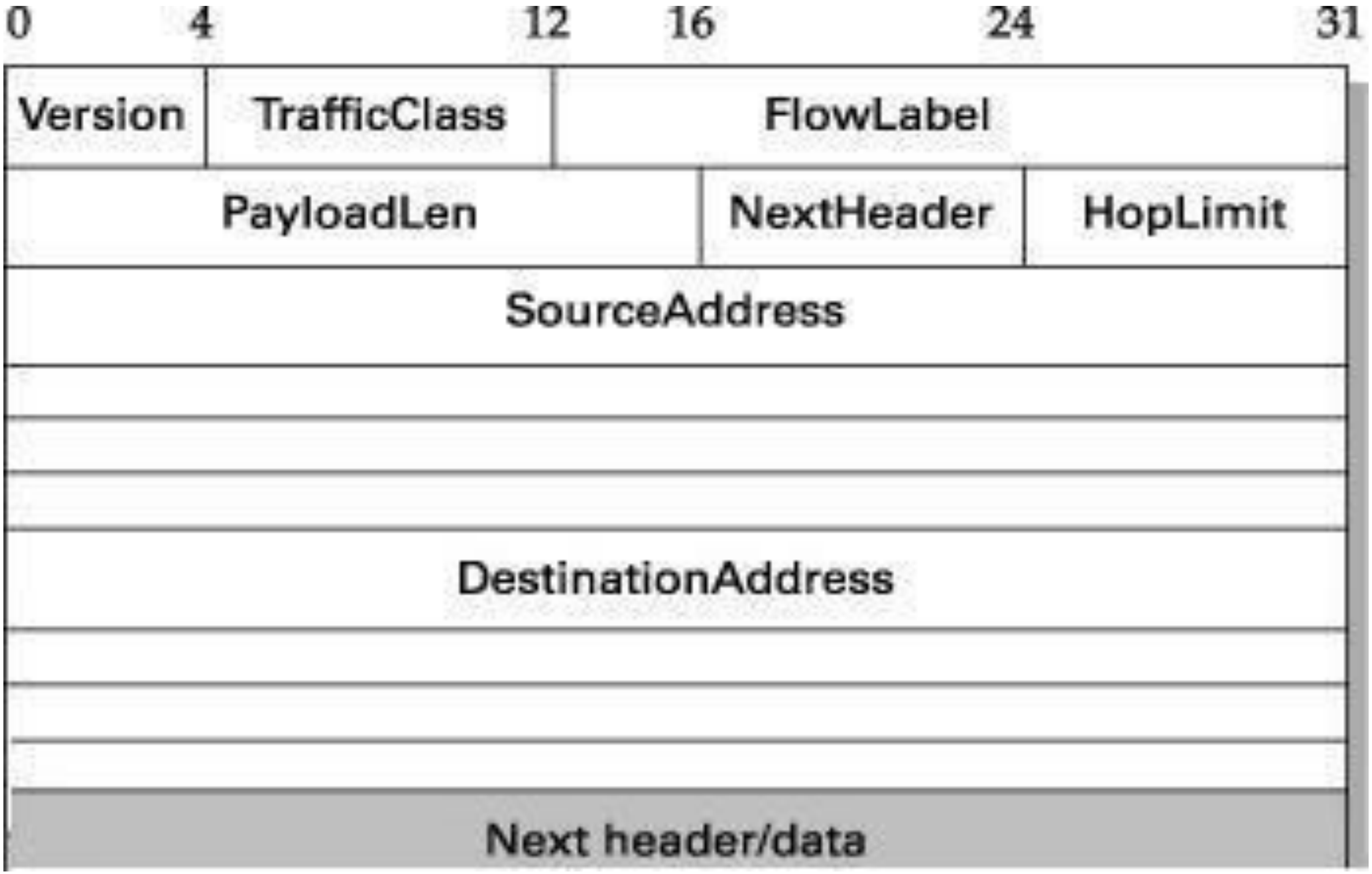
128. 96.33.81 → : : FFFF : 128.96.33.81 ***[4 x 32 bit = 128 bits]***

- This notation is called as CIDR notation or slash notation.

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.
- ***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


0
4
12
16
24
31
Version
TrafficClass
FlowLabel
PayloadLen
NextHeader
HopLimit
SourceAddress
DestinationAddress
Next header/data

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.

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**.