Unit-3
The Network Layer

Introduction

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.

Forwarding and Routing

The role of the network layer is to move packets from sending host to a receiving host. To do these two important network-layer functions can be identified:

• ***Forwarding.***Itrefers to the router-local action of transferring a packet from an input link interface to the appropriate output link interface. When a packet arrives at a router's input link, the router must move the packet to the appropriate output link.

• ***Routing.****Routing* refers to the network-wide process that determines the end-to-end paths that packets take from source to destination. The algorithms that calculate these paths are referred to as **routing algorithms**.

1. Routing Algorithms(Protocols)

The main function of the network layer is routing packets from the source machine to the destination machine. The **routing algorithm** is that part of the network layer software responsible for deciding which output line an incoming packet should be transmitted on.

A routing table can be either static or dynamic.

- A ***static table*** *is* one with manual entries.
- A ***dynamic table****,* on the other hand, is one that is updated automatically when there is a change somewhere in the internet.

Adaptive algorithms: change their routing decisions to reflect changes in the topology, and sometimes changes in the traffic as well. These are **dynamic routing** algorithms.

Non**adaptive algorithms:** do not base their routing decisions on any measurements or estimates of the current topology and traffic. This procedure is sometimes called **static routing**.

The set of optimal routes from all sources to a given destination form a tree rooted at the destination. Such a tree is called a **sink tree.**

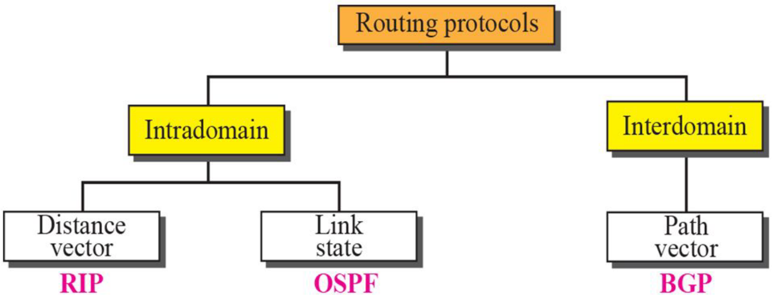


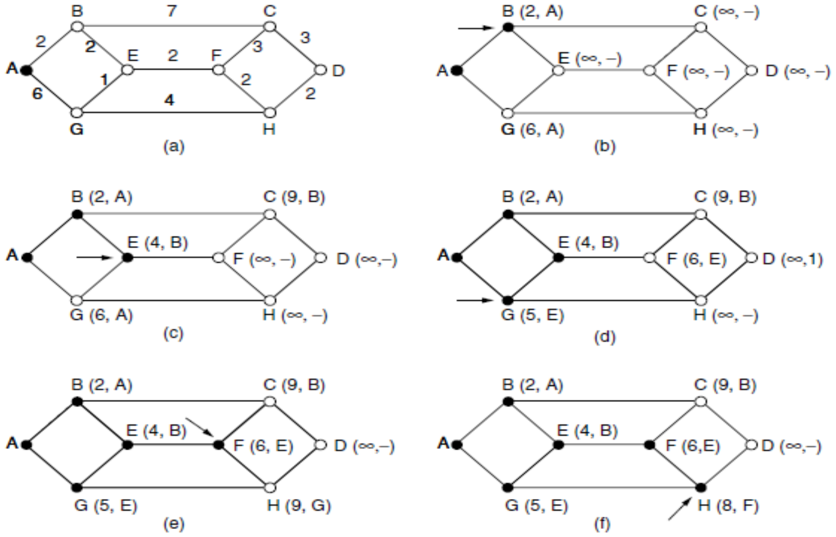
Fig: Popular routing protocols

Unicast routing algorithms

Nonadaptive routing algorithms(static)

i) **Shortest Path Algorithm: Shortest path routing** refers to the process of finding paths through a network that have a minimum of distance or other cost metric.

- A technique to study routing algorithms: The idea is to build a graph of the sub net, with each node of the graph representing a router and each arc of the graph representing a communication line (often called a link).
- To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.
- **Dijkstra's algorithm** create the shortest path algorithm



ii) Flooding: When a routing algorithm is implemented, each router must make decisions based on local knowledge, not the complete picture of the network. A simple local technique is **flooding**, in which every incoming packet is sent out on every outgoing line except the one it arrived on.

Flooding obviously generates vast numbers of duplicate packets, in fact, an infinite number unless some measures are taken to damp the process.

Adaptive algorithms (dynamic)

i) Distance Vector Routing: A **distance vector routing** algorithm operates by having each router maintain a table (vector) giving the best known distance to each destination and which link to use to get there. These tables are updated by exchanging information with the neighbors. Eventually, every router knows the best link to reach each destination.

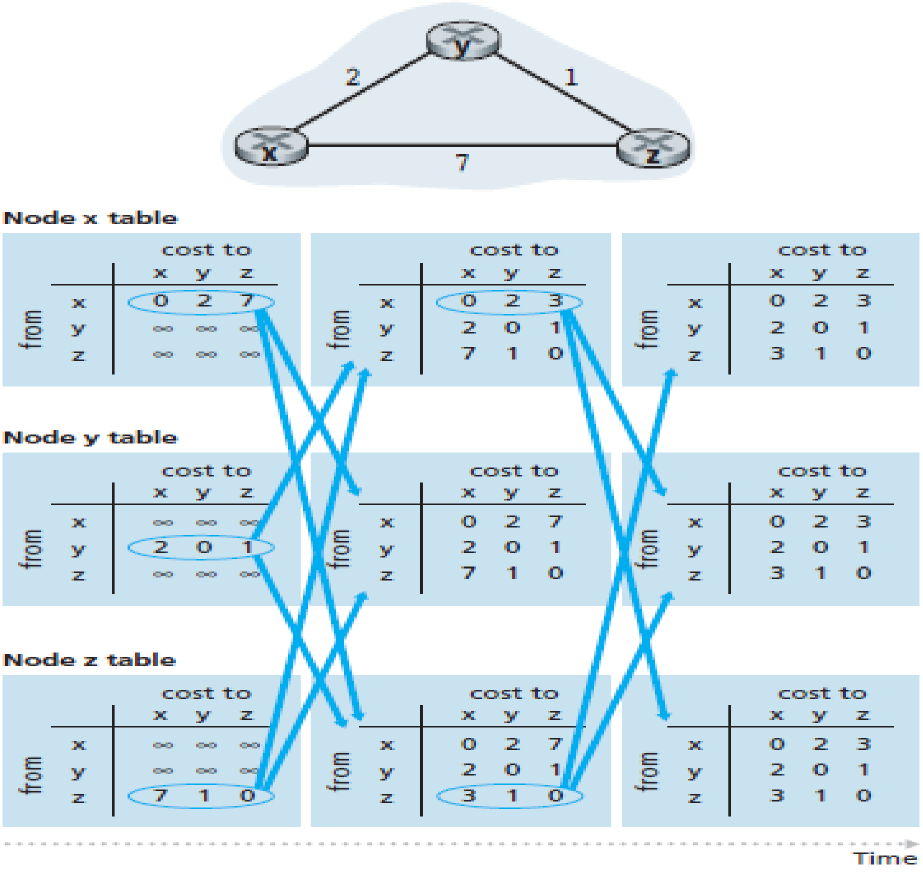
- The distance vector routing algorithm is sometimes called by other names, most commonly the distributed **Bellman-Ford** routing algorithm

Distance Vector: Let *dx*(*y*) be the cost of the least-cost path from node *x* to node *y*. Then the least costs are related by the celebrated Bellman-Ford equation, namely,

$$Dx(y) = \min_v\{c(x,v) + Dv(y)\} \text{ for each node } y \text{ in } N$$

Where the min_v in the equation is taken over all of *x*'s neighbors. Indeed, after traveling from *x* to *v*, if we then take the least-cost path from *v* to *y*, the path cost will be *c*(*x*,*v*) + *dv*(*y*). Since we must begin by traveling to some neighbor *v*, the least cost from *x* to *y* is the minimum of *c*(*x*,*v*)+ *dv*(*y*) taken over all neighbors *v*.

 Figure illustrates the operation of the DV algorithm for the simple three node network shown at the top of the figure.



Node x table

from	cost to x	cost to y	cost to z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	7	1	0

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	3	1	0

Node y table

from	cost to x	cost to y	cost to z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from	cost to x	cost to y	cost to z
x	0	2	7
y	2	0	1
z	7	1	0

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	3	1	0

Node z table

from	cost to x	cost to y	cost to z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from	cost to x	cost to y	cost to z
x	0	2	7
y	2	0	1
z	3	1	0

from	cost to x	cost to y	cost to z
x	0	2	3
y	2	0	1
z	3	1	0

Fig: Distance-vector (DV) algorithm

RIP: The Routing Information Protocol (RIP) is an intra domain routing protocol used inside an autonomous system. It is a very simple protocol based on distance vector routing.

ii) Link State Routing

Link-state routing uses link-state routers to exchange messages that allow each router to learn the entire network topology. Based on this learned topology, each router is then able to compute its routing table by using a shortest path computation.

- In link state routing, if each node in the domain has the entire topology of the domain the list of nodes and links, how they are connected including the type, cost (metric), and condition of the links (up or down)-the node can use Dijkstra's algorithm to build a routing table.

Let us define the following notation:

• *D*(*v*): cost of the least-cost path from the source node to destination *v* as of this iteration of the algorithm.

• *p*(*v*): previous node (neighbor of *v*) along the current least-cost path from the source to *v.*

• N`: subset of nodes; *v* is in N` if the least-cost path from the source to *v* is definitively known.

Link-State (LS) Algorithm for Source Node *u*

```
Initialization:
    N` = {u}
    for all nodes v
        if v is a neighbor of u
            then D(v) = c(u,v)
        else D(v) = ∞

Loop
```

```
        find w not in N` such that D(w) is a minimum
        add w to N`
        update D(v) for each neighbor v of w and not in N’:
            D(v) = min( D(v), D(w) + c(w,v) )
    until N`= N
```

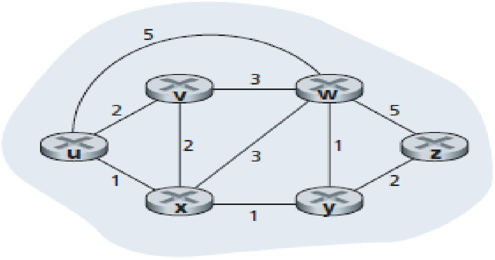
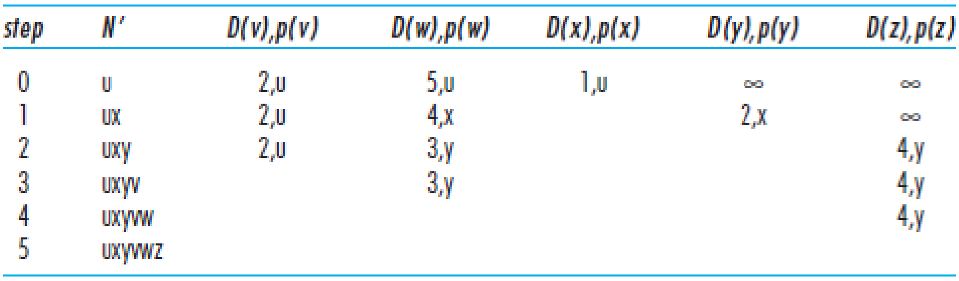


Fig: Graph model of a computer network

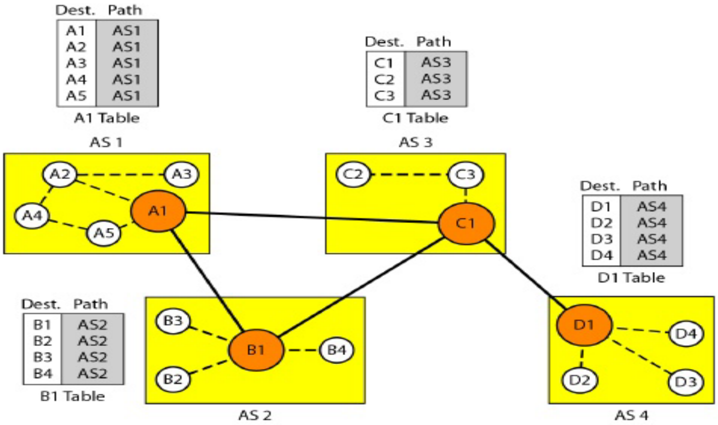


step	N’	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x		2,x	∞
2	uxy	2,u	3,y			4,y
3	uxyv		3,y			4,y
4	uxyvw					4,y
5	uxyvwz					

Table: Running the link-state algorithm on the network in Figure

OSPF: The Open Shortest Path First or OSPF protocol is an intra domain routing protocol based on link state routing. Its domain is also an autonomous system.

iii) Path Vector Routing: Path vector routing proved to be useful for inter domain routing. The principle of path vector routing is similar to that of distance vector routing. In path vector routing, we assume that there is one node in each autonomous system that acts on behalf of the entire autonomous system.
Let us call it the speaker node. The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring ASs. Only speaker nodes in each AS can communicate with each other.
Initialization: At the beginning, each speaker node can know only the reachability of nodes inside its autonomous system. Figure shows the initial tables for each speaker node in a system made of four ASs.
Node Al is the speaker node for ASl, Bl for AS2, Cl for AS3, and Dl for AS4.



Dest.	Path
A1	AS1
A2	AS1
A3	AS1
A4	AS1
A5	AS1

A1 Table

Dest.	Path
C1	AS3
C2	AS3
C3	AS3

C1 Table

Dest.	Path
D1	AS4
D2	AS4
D3	AS4
D4	AS4

D1 Table

Dest.	Path
B1	AS2
B2	AS2
B3	AS2
B4	AS2

B1 Table

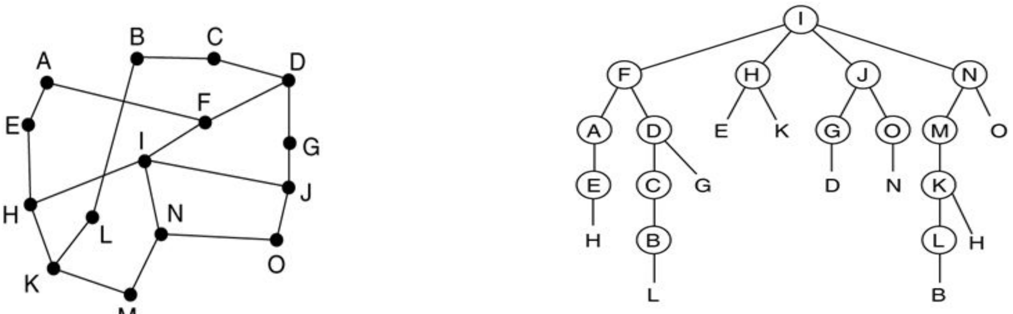
Fig: Initial routing tables in path vector routing

Sharing Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors.

BGP: Border Gateway Protocol (BGP) is an inter domain routing protocol using path vector routing.

Multicast Routing Protocols

 Broadcast Routing: In some applications, hosts need to send messages to many or all other hosts.
In broadcast communication, the relationship between the source and the destination is ***one-to-all***. There is only one source, but all the other hosts are the destinations.
For example, a service distributing weather reports, stock market updates, or live radio programs might work best by sending to all machines are interested read the data. Sending a packet to all destinations simultaneously is called **broadcasting**.



(a) Network (b) The tree built by reverse path forwarding.

 Multicast Routing: Sending a message to such a group is called **multicasting**, and the routing algorithm used is called **multicast routing**.
In multicast communication, there is one source and a group of destinations. The relationship is one-to-many. In this type of communication, the source address is a unicast address, but the destination address is a group address, which defines one or more destinations. The group address identifies the members of the group.

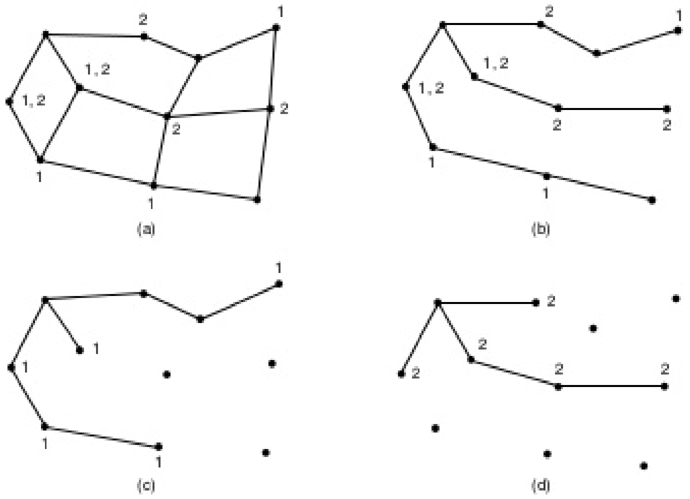


Fig: (a) A network. (b) A spanning tree for the leftmost router.
(c) A multicast tree for group 1. (d) A multicast tree for group 2.

 Anycast Routing: In anycast, a packet is delivered to the nearest member of a group. Schemes that find these paths are called **anycast routing**.
However, a packet destined for an anycast address is delivered to only one of the members of the anycast group, the nearest one (the one with the shortest route).

2. Internetworking

- We have implicitly assumed that there is a single homogeneous network, with each machine using the same protocol in each layer. Unfortunately, this assumption is wildly optimistic. Many different networks exist, including LANs, MANs, and WANs.

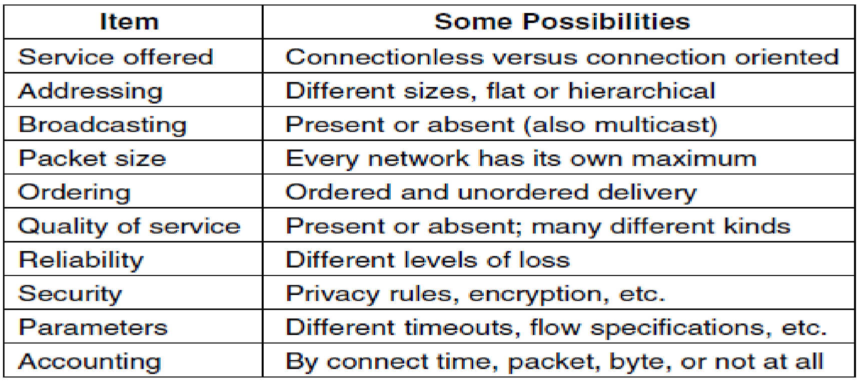
- The issues that arise when two or more networks are connected to form an **internetwork, or more simply an internet.**

How Networks Differ:

- Networks can differ in many ways. Some of the differences, such as different modulation techniques or frame formats, are internal to the physical and data link layers.

- We list some of the differences that can be exposed to the network layer.

- It is papering over these differences that makes internetworking more difficult than operating within a single network.



Item	Some Possibilities
Service offered	Connectionless versus connection oriented
Addressing	Different sizes, flat or hierarchical
Broadcasting	Present or absent (also multicast)
Packet size	Every network has its own maximum
Ordering	Ordered and unordered delivery
Quality of service	Present or absent; many different kinds
Reliability	Different levels of loss
Security	Privacy rules, encryption, etc.
Parameters	Different timeouts, flow specifications, etc.
Accounting	By connect time, packet, byte, or not at all

Fig: Some of the many ways networks can differ

How Networks Can Be Connected:

There are two basic choices for connecting different networks:

1. We can build devices that translate or convert packets from each kind of network into packets for each other network.

2. We can try to solve the problem by adding a layer of indirection and building a common layer on top of the different networks.

IP is the foundation of the modern Internet. IP provides a ***universal packet format*** that all routers recognize and that can be passed through almost every network. A router that can handle multiple network protocols is called a **multiprotocol router.**

How interconnection with a common network layer can be used to interconnect dissimilar networks. An internet comprised of 802.11, MPLS, and Ethernet network is shown in Fig. (a)

Suppose that the source machine on the 802.11 network wants to send a packet to the destination machine on the Ethernet network. Since these technologies are different, and they are further separated by another kind of network (MPLS).

An internet comprised of 802.11, MPLS, and Ethernet networks are shown in Fig.

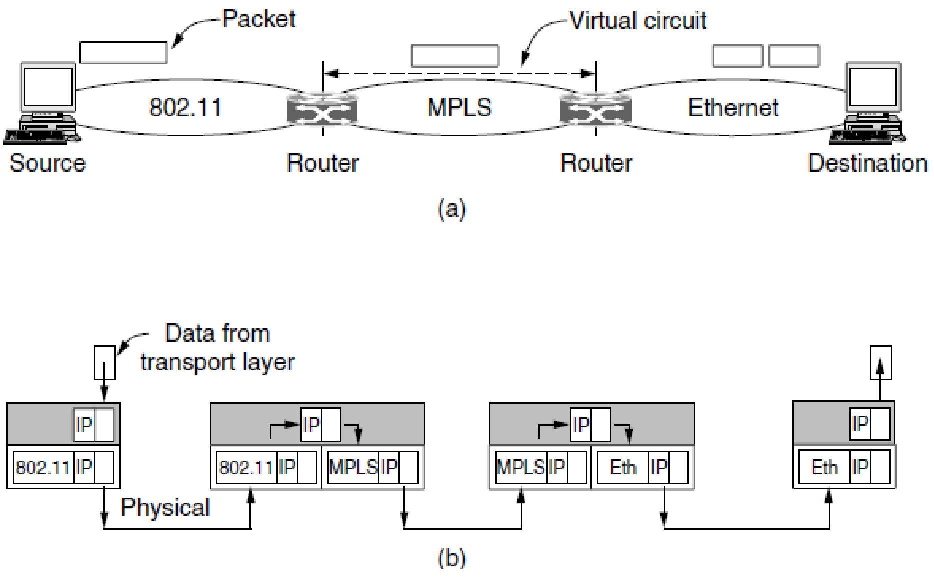


Fig: (a) A packet crossing different networks. (b) Network and link layer protocol processing.

Tunneling

The special case that is manageable even for different network protocols. This case is where the source and destination hosts are on the same type of network, but there is a different network in between. As an example, think of an international bank with an IPv6 network in Paris, an IPv6 network in London and connectivity between the offices via the IPv4 Internet. This situation is shown in Fig.

The solution to this problem is a technique called ***tunneling***. The path through the IPv4 Internet can be seen as a big tunnel extending from one multiprotocol router to the other. The IPv6 packet just travels from one end of the tunnel to the other.

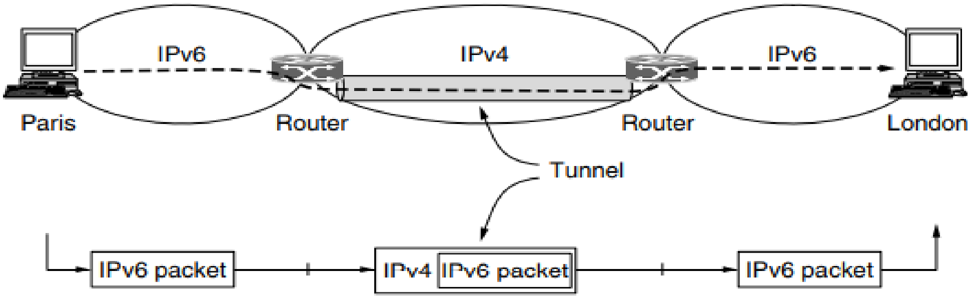


Fig: Tunneling a packet from Paris to London.

Internetwork Routing

A two-level routing algorithm. Within each network, an **intradomain or interior gateway** protocol is used for routing. Across the networks that make up the internet, an **interdomain or exterior gateway** protocol is used. The networks may all use different intradomain protocols, but they must use the same interdomain protocol.

In the Internet, the interdomain routing protocol is called **BGP** (**Border Gateway Protocol**).

Packet Fragmentation

Each network or link imposes some maximum size on its packets. These limits have various causes, among them
1. Hardware (e.g., the size of an Ethernet frame).
2. Operating system (e.g., all buffers are 512 bytes).
3. Protocols (e.g., the number of bits in the packet length field).
4. Desire to reduce error-induced retransmissions to some level. 5. Desire to prevent one packet from occupying the channel too long.

Solution:

i. The packet size is called the **Path MTU** (**Path Maximum Transmission Unit**). Even if the source did know the path MTU, packets are routed independently in a connectionless network such as the Internet.

ii. The alternative solution to the problem is to allow routers to break up packets into **fragments**, sending each fragment as a separate network layer packet. However, as every parent of a small child knows, converting a large object into small fragments is considerably easier than the reverse process.

3. The network layer in the Internet

i) The Internet Protocol(IP): Forwarding and Addressing in the Internet

Internet addressing and forwarding are important components of the Internet Protocol (IP). There are two versions of IP in use today.

IP version 4 (IPv4) and IP version 6 (IPv6). IP addresses are made up of binary values and drive the routing of all data over the Internet. IPv4 addresses are 32 bits long, and IPv6 addresses 128 bits long.

IP version 4 (IPv4): Datagram Format

The Internet Protocol version 4 (IPv4), is responsible for packetizing, forwarding, and delivery of a packet at the network layer. The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols.

IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service. IPv4 is also a connectionless protocol for a packet- switching network that uses the datagram approach. This means that each datagram is handled independently, and each datagram can follow a different route to the destination.

Packets in the IPv4 layer are called datagrams. A datagram is a variable-length packet consisting of two parts: **header** and **data**. The header is 20 to 60 bytes in length. The header has a ***20-byte*** fixed part and a variable-length optional part. The header format is shown in Fig.

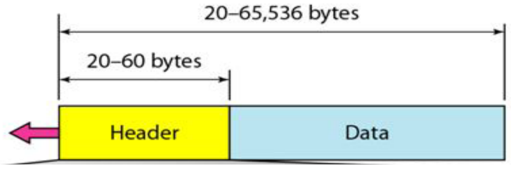
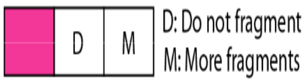


Fig: IPv4 Datagram



Flags (3 bits)

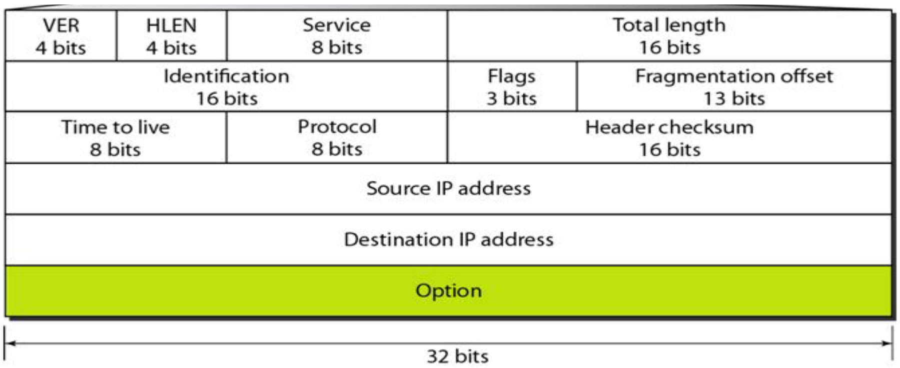
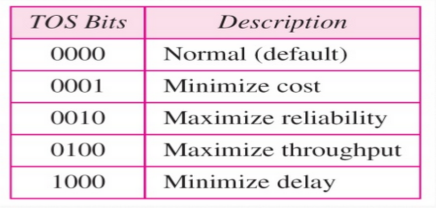


Fig: The IPv4 (Internet Protocol) header

- ***Version Number.*** The 4-bit version number (VER) field defines the version of the IPv4 protocol has the value of 4.

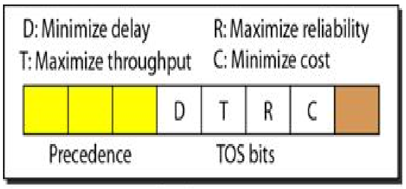
- ***Header Length.*** The 4-bit header length (HLEN) field defines the total length of the datagram header in 4-byte words. The IPv4 datagram has a variable-length header.
This 4-bit field indicates the number of 4-byte words in the IP header. The length of the header can be between 20 and 60 bytes. Therefore, the value of this field is always between 5 (5 × 4 = 20) and 15 (15 × 4 = 60).

- ***Service Type.*** In the original design of the IP header, this field was referred to as type of service (TOS), which defined how the datagram should be handled. The IETF redefined the field to provide *differentiated services.*
Precedence is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary). The precedence defines the priority of the datagram in issues such as congestion.



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

Fig: Type of Service



Service type

Total Length. This 16-bit field defines the total length (header plus data) of the IP datagram in bytes. A 16-bit number can define a total length of up to 65,535.

Length of Data =Total Length – (HLEN) x4

- ***Identification.*** This 16-bit field identifies a datagram originating from the source host. The combination of the identification and source IPv4 address must uniquely define a datagram as it leaves the source host.
To guarantee uniqueness, the IPv4 protocol uses a ***counter*** to label the datagrams and after every label assigning counter is incremented by 1. When datagram is fragmented, all fragments have the same identification number of the original datagram.

- ***Flags.*** This is a 3-bit field. The ***first bit*** is reserved.

The ***second bit*** is called the do not fragment bit.

  - If its value is 1, the machine must not fragment the datagram. If it cannot pass the datagram, it discards the datagram and sends an ICMP error message to the source host.
  - If its value is 0, the datagram can be fragmented if necessary.

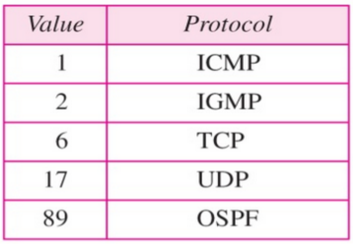
The ***third bit*** is called the more fragment bit.

- If its value is 1, it means the datagram is not the last fragment; there are more fragments after this one.
- If its value is 0, it means this is the last or only one datagram.

- ***Fragmentation Offset.*** This 13-bit field shows the relative position of this fragment with respect to the whole datagram.

- ***Time-to-live.*** A datagram has a limited lifetime in its travel through an internet. This field was originally designed to hold a timestamp, which was decremented by each visited router. The datagram was discarded when the value became zeroand is supposed to be decremented multiple times when a packet is queued for a long time in a router.

- ***Protocol.*** This 8-bit field defines the higher-level protocol that uses the services of the IPv4 layer. An IPv4 datagram can encapsulate data from several higher-level protocols such as TCP, UDP, ICMP, and IGMP. This field specifies the final destination protocol to which the IPv4 datagram is delivered.



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

Table: protocol values

- ***Header checksum.*** The checksum in the IPv4 packet covers only the header, not the data.

The implementation of the checksum in the IPv4 packet follows as; first, the value of the checksum field is set to 0. Then the entire header is divided into 16-bit sections and added together. The result (sum) is complemented and inserted into the checksum field.

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

- **Options.** A datagram header can have up to 40 bytes of options. Options can be used for network testing and debugging.

IPV4 Addresses

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

- **Address space is the total number of** addresses used by the protocol.IPv4 uses 32-bit addresses: The ***address space***=2^{32} =4,294,967,296 (more than 4 billion).
- **Notations**: There are two notations to show an IPv4 address

Binary notation: Address is displayed as 32 bits. Each octet is often referred to as byte.IPv4 address referred to as 32-bit address or 4-byte address

Dotted-decimal notation: More compact and easier to read, Written in decimal form with a decimal point (dot) separating the bytes.

Example: Each decimal value range from 0 to 255

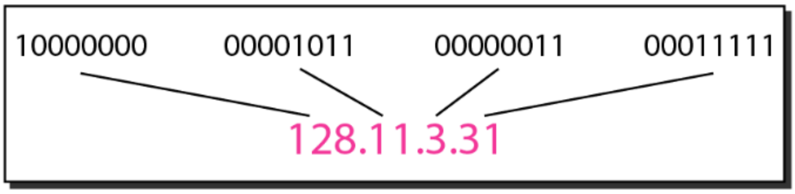
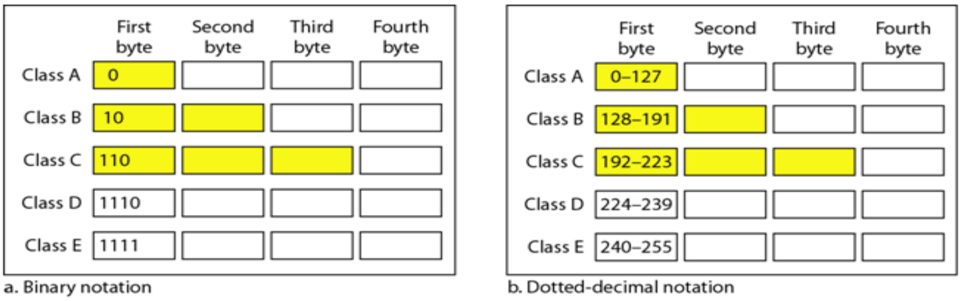


Fig: different notations in IPv4 addressing

- **Classful addressing:** IPv4 addressing used the concept of classes. This architecture is called classful addressing. In classful addressing, the address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the address space.



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

a. Binary notation

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

b. Dotted-decimal notation

Fig: *Finding the classes in binary and dotted-decimal notation*

Addresses in Classes A, B and C are unicast addresses. Addresses in Class D are for multicast address and addresses in class E are reserved.

- **Classes and Blocks:** One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size as shown in Table.

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

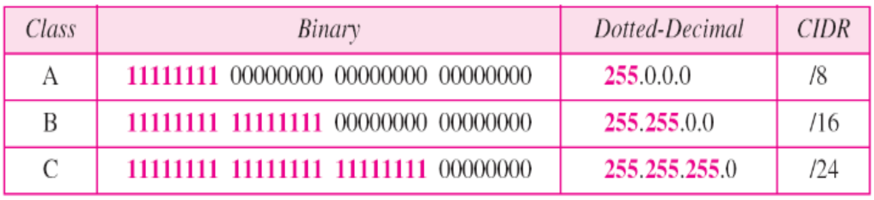
Table: Number of blocks and block size in classful IPv4 addressing

Class A addresses were designed for large organizations with a large number of attached hosts or routers. **Class B** addresses were designed for midsize organizations with tens of thousands of attached hosts or routers. **Class C** addresses were designed for small organizations with a small number of attached hosts or routers

- **Netid and Hostid:** In classful addressing, an IP address in class A, B, or C is divided into netid and hostid.

Note that the concept does not apply to classes D and E. In **Class A**, one byte defines the netid and three bytes define the hostid. In **Class B**, two bytes define the netid and two bytes define the hostid. In **Class C**, three bytes define the netid and one byte defines the hostid.

- **Mask:** The length of the netid and hostid (in bits) is predetermined in classful addressing, we can also use a mask (also called the default mask), a 32-bit number made of contiguous 1s followed by contiguous 0s. The masks for classes A, B, and C are shown in Table. The concept does not apply to classes D and E.



Class	Binary	Dotted-Decimal	CIDR
A	11111111 0000000 0000000 0000000	255.0.0.0	/8
B	11111111 11111111 0000000 0000000	255.255.0.0	/16
C	11111111 11111111 11111111 0000000	255.255.255.0	/24

Table:Default masks for classful addressing

The last column of Table shows the mask in the form / n where n can be 8, 16, or 24 in classful addressing. This notation is also called slash notation or Classless Inter domain Routing (CIDR) notation.

- **Subnetting:** During the era of classful addressing, subnetting was introduced. If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called subnets). Subnetting increases the number of 1s in the mask.

- **Supernetting:** The size of a class C block with a maximum number of 256 addresses did not satisfy the needs of most organizations. Even a midsize organization needed more addresses. One solution was supernetting.

In supernetting, an organization can combine several class C blocks to create a large range of addresses.

- **Address Depletion:** The flaws in classful addressing scheme combined with the fast growth of the Internet led to the near depletion of the available addresses. Yet the number of devices on the Internet is much less than the 2^{32} address space. We have run out of class A and B addresses, and a class C block is too small for most midsize organizations. One solution for the problem is the idea of classless addressing.

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

- **Address Blocks:** In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block (the number of addresses) varies based on the nature and size of the entity.

Restriction: To simplify the handling of addresses, the Internet authorities impose three restrictions on classless address blocks:
1. The addresses in a block must be contiguous, one after another.
2. The number of addresses in a block must be a power of 2 (1, 2, 4, 8, . . .).
3. The first address must be evenly divisible by the number of addresses.

Example: Figure shows a block of addresses granted to a small business that needs 16 addresses.

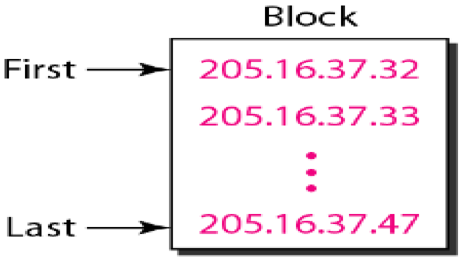


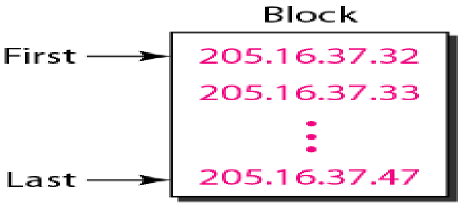
Figure: A block of 16 addresses granted to a small organization

We can see that the restrictions are applied to this block. The addresses are contiguous. The number of addresses is a power of 2 (16 = 2^4), and the first address is divisible by 16.

- **Mask:** A better way to define a block of addresses is to select any address in the block and the mask. A mask is a 32 bit number in which the ***n*** leftmost bits are 1s and the ***32 – n*** rightmost bits are 0s. However, in classless addressing the mask for a block can take any value from 0 to 32.

  - ***First address:*** The first address in the block can be found by setting the rightmost ***32 – n*** bits to ***0s.***
  - ***Last address:*** The last address in the block can be found by setting the rightmost ***32 – n*** bits to ***1s.***
  - ***Number of addresses:*** The number of addresses in the block can be found by using the formula 2^{32-n}

- ***Ex:*** A block of addresses is granted to a small organization. We know that one of the addresses is **205.16.37.39/28.**



First address: The binary representation of the given address is

11001101 00010000 00100101 0010**0111**

If we set 32−28 rightmost bits to 0, we get

11001101 00010000 00100101 0010**0000**

or

205.16.37.32

Last address: The binary representation of the given address is

11001101 00010000 00100101 0010**0111**

If we set 32 − 28 rightmost bits to 1, we get

11001101 00010000 00100101 0010**1111**

or

205.16.37.47

Number of addresses: The value of n is 28, which means that number of addresses is 2^{32-28} or 16.

- **Two-Level Hierarchy: No Subnetting:** IP addresses can define only two levels of hierarchy when not subnetted. The n left-most bits of the address x.y.z.t/ n define the network; the 32- n rightmost bits define the particular host (computer or router) to the network.

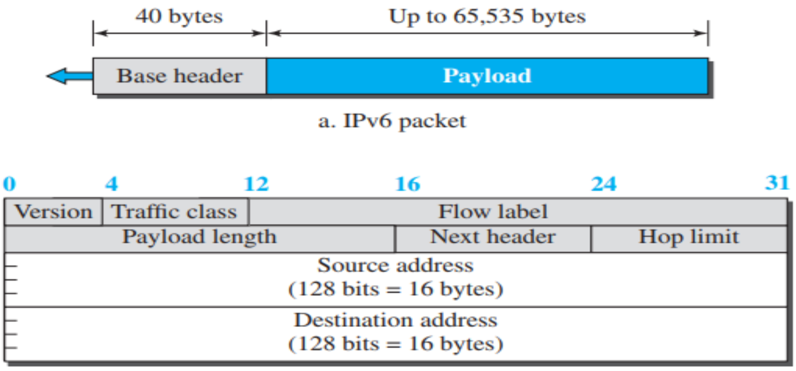
- **Three-Levels of Hierarchy: Subnetting:** An organization that is granted a large block of addresses may want to create clusters of networks (called subnets) and divide the addresses between the different subnets.

Example, suppose an organization is given the block 17.12.40.0/26, which contains 64 addresses. The organization has needs to divide the addresses into three subblocks of 32, 16, and 16 addresses.
We can find the new masks by using the following arguments:
1. Suppose the mask for the 1st subnet is n1, then 2^{32-n1}must be 32, which means that n1 = 27.
2. Suppose the mask for the second subnet is n2, then $2^{32 -n2}$must be 16, which means that n2 = 28.
3. Suppose the mask for the third subnet is n3, then 2^{32-n3}must be 16, which means that n3 = 28.
This means that we have the masks 27, 28, 28 with the organization mask being 26.

IP Version 6

An IPv6 address is 128 bits or 16 bytes (octets) long, four times the address length in IPv4. The IPV6 address is represented as colon hexadecimal notation (or colon hex for short) divides the address into eight sections, each made of four hexadecimal digits separated by colons.
Its major goals were:
1. Support billions of hosts.
2. Reduce the size of the routing tables.
3. Simplify the protocol, to allow routers to process packets faster.
4. Provide better security (authentication and privacy).
5. Pay more attention to the type of service, particularly for real-time data.
6. Aid multicasting by allowing scopes to be specified.
7. Make it possible for a host to roam without changing its address.
8. Allow the protocol to evolve in the future.
9. Better header format

The IPV6 Protocol: The IPv6 packet is shown in Figure. Each packet is composed of a base header followed by the payload. The base header occupies 40 bytes, whereas payload can be up to 65,535 bytes of information.



a. IPv6 packet

b. Base header

Fig: IPv6 datagram

- ***Version***. The 4-bit version field defines the version number of the IP. For IPv6, the value is 6.
- ***Traffic class.*** The 8-bit traffic class field is used to distinguish different payloads with different delivery requirements. It replaces the type-of-service field in IPv4.
- ***Flow label***. The flow label is a 20-bit field that is designed to provide special

handling for a particular flow of data.

- ***Payload length***. The 2-byte payload length field defines the length of the IP datagram excluding the header In IPv6, the length of the base header is fixed (40 bytes); only the length of the payload needs to be defined.
- ***Next header.*** The next header is an 8-bit field defining the type of the first extension header (if present) or the type of the data that follows the base header in the datagram.
- ***Hop limit***. The 8-bit hop limit field serves the same purpose as the TTL field in IPv4. The Hop limit field is used to keep packets from living forever.
- ***Source and destination addresses.*** The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram. The destination address field is a 16-byte (128-bit) Internet address that identifies the destination of the datagram.

IPv6 Address

Despite all short-term solutions, address depletion is still a long-term problem for the Internet. This and still a long-term problem for the Internet. This and other problems in the IP protocol itself have been the motivation for IPv6.

Structure: An IPv6 address consists of 16 bytes (octets); it is 128 bits long.

- **Hexadecimal Colon Notation:** To make addresses more readable, IPv6 specifies ***hexadecimal colon*** notation. In this notation, 128 bits is divided into eight sections, each 2 bytes in length. The address consists of ***32 hexadecimal digits***, with every four digits separated by a colon, as shown in Figure.

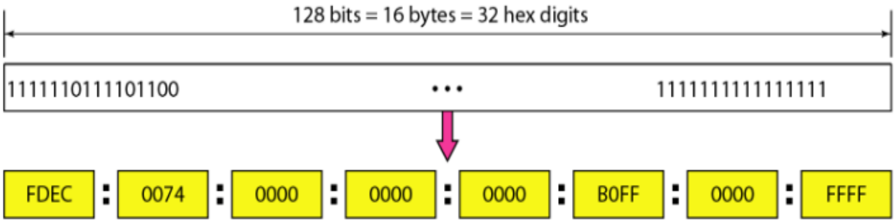
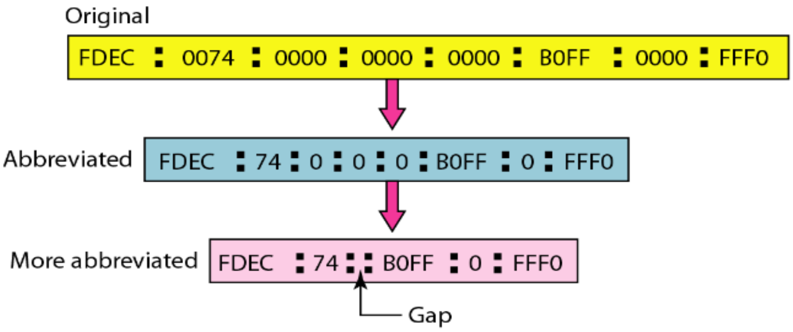


Fig: IPv6 address in binary and hexadecimal colon notation

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



Original
FDEC : 0074 : 0000 : 0000 : 0000 : B0FF : 0000 : FFF0
Abbreviated
FDEC : 74 : 0 : 0 : 0 : B0FF : 0 : FFF0
More abbreviated
FDEC : 74 :: B0FF : 0 : FFF0
Gap

Fig: Abbreviated IPv6 addresses

Address Space: IPv6 has a much larger address space; 2^{128} addresses are available. The designers of IPv6 divided the address into several categories.

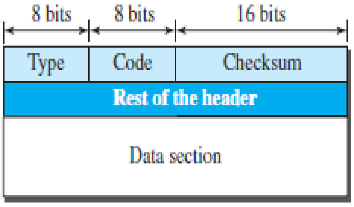
ii) Internet Control Protocols

ICMP—The Internet Control Message Protocol

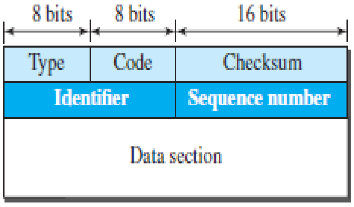
ICMP is used by hosts and routers to communicate network-layer information to each other.

The most typical use of ICMP is for error reporting. The operation of the Internet is monitored closely by the routers.
When something unexpected occurs during packet processing at a router, the event is reported to the sender by the ICMP (Internet Control Message Protocol).
ICMP is often considered part of IP but architecturally it lies just above IP, as ICMP messages are carried inside IP datagrams. Each ICMP message type is carried encapsulated in an IP packet.

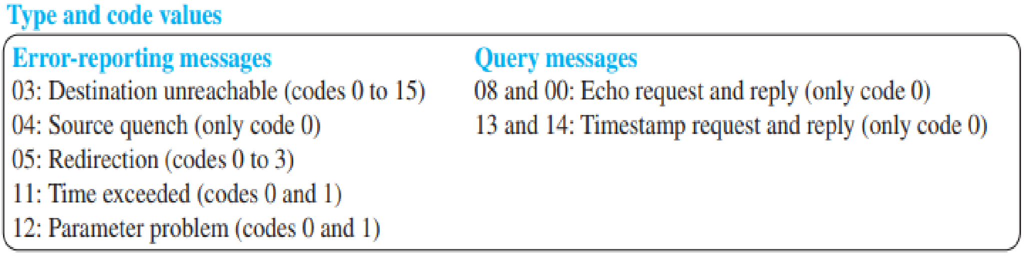
Messages: ICMP messages are divided into two broad categories: ***error-reporting messages*** and ***query messages.***
The error-reporting messages report problems that a router or a host (destination) may encounter when it processes an IP packet.
The query messages, which occur in pairs, help a host or a network manager get specific information from a router or another host.



Error-reporting messages



Query messages



Type and code values

Error-reporting messages	Query messages
03: Destination unreachable (codes 0 to 15)	08 and 00: Echo request and reply (only code 0)
04: Source quench (only code 0)	13 and 14: Timestamp request and reply (only code 0)
05: Redirection (codes 0 to 3)	
11: Time exceeded (codes 0 and 1)	
12: Parameter problem (codes 0 and 1)	

Error-Reporting and query Messages:

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

- The ***DESTINATION UNREACHABLE*** message is used when the router cannot locate the destination
- The ***TIME EXCEEDED*** message is sent when a packet is dropped because its TtL (Time to live) counter has reached zero.
- The ***PARAMETER PROBLEM*** message indicates that an illegal value has been detected in a header field.
- The ***SOURCE QUENCH*** (type 4) message, which informs the sender that the network has encountered congestion and the datagram has been dropped; the source needs to slow down sending more datagrams.
- The ***REDIRECT*** message is used when a router notices that a packet seems to be routed incorrectly. It is used by the router to tell the sending host to update to a

better route.

- The *ECHO* and *ECHOREPLY* messages are sent by hosts to see if a given destination is reachable and currently alive. Upon receiving the ECHO message, the destination is expected to send back an ECHO REPLY message.
- The *TIMESTAMPREQUEST* and *TIMESTAMPREPLY* messages are similar, except that the arrival time of the message and the departure time of the reply are recorded in the reply. This facility can be used to measure network performance.

IGMP (Internet Group Management Protocol) - Internet Multicasting

Normal IP communication is between one sender and one receiver. However, for some applications, it is useful for a process to be able to send to a large number of receivers simultaneously.

IP supports one-to-many communication, or multicasting, using class D IP addresses. Each class D address identifies a group of hosts. Twenty-eight bits are available for identifying groups, so over 250 million groups can exist at the same time.

The range of IP addresses 224.0.0.0/24 is reserved for multicast on the local network. Some examples of local multicast addresses are:

224.0.0.1	All systems on a LAN
224.0.0.2	All routers on a LAN
224.0.0.5	All OSPF routers on a LAN
224.0.0.251	All DNS servers on a LAN

The protocol that is used today for collecting information about group membership is the ***Internet Group Management Protocol (IGMP).*** IGMP is a protocol defined at the network layer; it is one of the auxiliary protocols, like ICMP, which is considered part of the IP. IGMP messages are encapsulated in an IP datagram.

Messages There are only two types of messages in IGMP, ***query*** and ***report messages***, as shown in Figure.

A query message is periodically sent by a router to all hosts attached to it to ask them to report their interests about membership in groups.

A report message is sent by a host as a response to a query message.

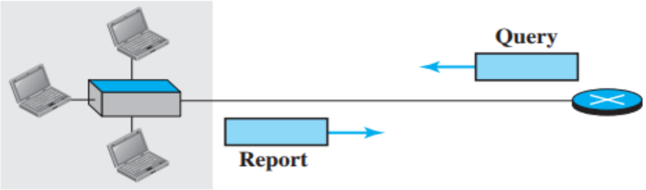


Fig: IGMP operation

Query Message The query message is sent by a router to all hosts in each interface to collect information about their membership. There are three versions of query messages, as described below:

- A ***general query message*** is sent about membership in any group. Note that all routers attached to the same network receive this message to inform them that this message is already sent and that they should refrain from resending it.
- A ***group-specific query message*** is sent from a router to ask about the membership related to a specific group.

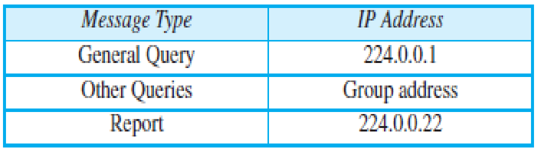
This is sent when a router does not receive a response about a specific group in the network.

The group identifier (multicast address) is mentioned in the message. The message is encapsulated in a datagram with the destination address set to the corresponding multicast address.

- A ***source-and-group-specific query message*** is sent from a router to ask about

the membership related to a specific group when the message comes from a specific source or sources.
Again the message is sent when the router does not hear about a specific group related to a specific host or hosts.

Report Message A report message is sent by a host as a response to a query message. The message contains a list of records in which each record gives the identifier of the corresponding group (multicast address) and the addresses of all sources that the host is interested in receiving messages from.

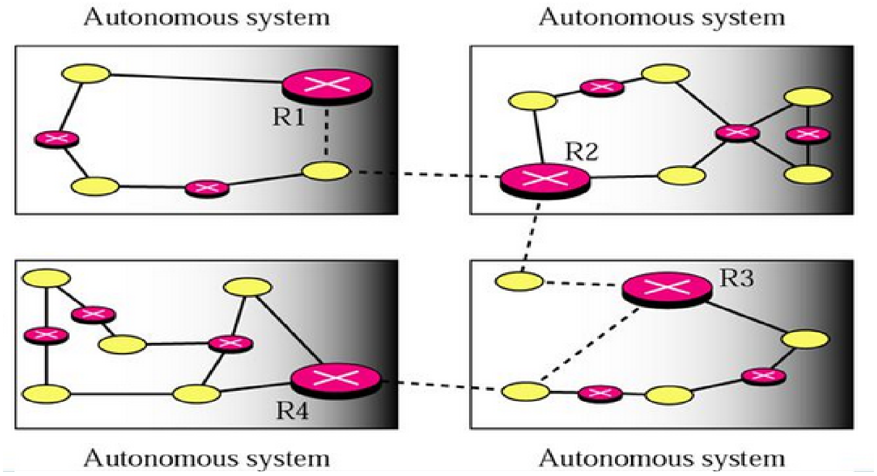


Message Type	*IP Address*
General Query	224.0.0.1
Other Queries	Group address
Report	224.0.0.22

iii) OSPF and BGP

Autonomous System (AS): Group of networks and routers under the authority of a single administration.
The routing operations performed inside an autonomous system is known as ***intradomain routing or interior gateway routing.***
When the routing is performed between two autonomous systems, it is referred to as ***interdomain routing or exterior gateway routing***.

- Solid lines show the communication between routers that use interior routing protocols.
- Broken lines show the communication between routers that use an exterior routing protocol.



Open Shortest Path First (OSPF) - An Interior Gateway Routing Protocol

Open Shortest Path First (OSPF) is also an ***intradomain*** routing protocol but it is based on the ***link-state routing protocol***. OSPF is an open protocol, which means that the specification is a public document.
Open shortest path first (OSPF) is a link-state routing protocol which is used to find the best path between the source and the destination router using its own shortest path first (SPF) algorithm.

OSPF is developed by Internet Engineering Task Force (IETF) as one of the Interior Gateway Protocol (IGP), i.e, the protocol which aims at moving the packet within a large autonomous system or routing domain. It is a network layer protocol which works on the protocol number 89.

With OSPF, a router constructs a complete topological map (that is, a graph) of the entire autonomous system. The router then locally runs Dijkstra's shortest-path algorithmto determine a shortest-path tree to all *subnets*, with itself as the root node.

With OSPF, a router broadcasts routing information to *all* other routers in the autonomous system, not just to its neighboring routers. A router broadcasts link state information whenever there is a change in a link's state. It also broadcasts a link's state periodically (at least once every 30 minutes), even if the link's state has not changed.

Border Gateway Protocol (BGP) - An exterior Gateway Routing Protocol

The Border Gateway Protocol version 4 (BGP4) is the only ***interdomain*** routing protocol used in the Internet today. BGP4 is based on the path-vector algorithm to provide information about the reachability of networks in the Internet.

- Border Gateway Protocol (BGP) is an Internet Engineering Task Force (IETF) standard, and the most scalable of all routing protocols.
- BGP is the routing protocol of the global Internet, as well as for Service Provider private networks. BGP has expanded upon its original purpose of carrying Internet reachability information, and can now carry routes for Multicast, IPv6, VPNs, and a variety of other data.
- BGP is most commonly used as an external routing protocol for large networks with multiple connections to the Internet.
- Companies or institutions that use BGP will have a unique autonomous system number that is exchanged with other BGP networks to create peering relationships with other autonomous systems.

As an inter-AS routing protocol, BGP provides each AS a means to

1. Obtain subnet reachability information from neighboring ASs.
2. Propagate the reachability information to all routers internal to the AS.

In BGP, pairs of routers exchange routing information over semipermanent TCP connections using port 179.There is typically one such BGP TCP connection for each link that directly connects two routers in two different ASs; thus, in Figure, there is a TCP connection between gateway routers 3a and 1cand another TCP connection between gateway routers 1b and 2a.

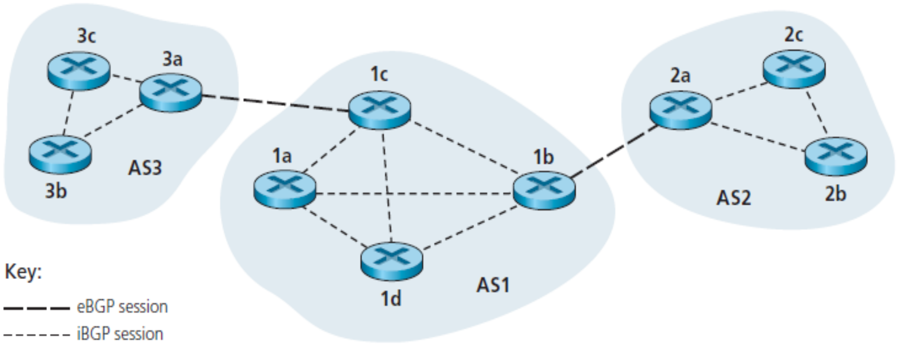


Fig: eBGP and iBGP sessions

- The TCP connection along with all the BGP messages sent over the connection is called a **BGP session**. Furthermore, a BGP session that spans two AS is called an **external BGP** (**eBGP**) **session**, and a BGP session between routers in the same AS is called an **internal BGP** (**iBGP**) **session**.