

UNIT-IV
NETWORK LAYER

Syllabus

Network Layer: Design issues, Routing algorithms, congestion control algorithms, Host to Host Delivery: Internetworking, addressing and routing, IP addressing (class full & Classless), Subnet, Network Layer Protocols: ARP, IPV4, ICMP, IPV6 and ICMPV6.

The main role of the network layer is **to move the packets from sending host to the receiving host.**

The Network Layer is the third layer of the OSI model. It **handles** the service requests from the transport layer and further **forwards** the service request to the data link layer. The network layer translates the logical addresses into physical addresses. It determines the route from the source to the destination and also manages the traffic problems such as switching, routing and controls the congestion of data packets.

Flooding:

- In this algorithm every incoming packet is sent out on every outgoing line except the line on which it has arrived.
- One disadvantage of flooding is that it generates a large number of duplicate packets. In fact it produces infinite number of duplicate packets unless we somehow dump the process. Therefore, we use **selective flooding**.
- In this algorithm every incoming packet is not sent out on every output line.
- Instead packet is sent only on those lines which are approximately going in the right direction.

The main functions performed by the network layer are:

- **Routing:** When a packet reaches the router's input link, the router will move the packets to the router's output link. For example, a packet from S1 to R1 must be forwarded to the next router on the path to S2.
- **Logical Addressing:** The data link layer implements the physical addressing and network layer implements the logical addressing. Logical addressing is also used to distinguish between source and destination system. The network layer adds a header to the packet which includes the logical addresses of both the sender and the receiver.
- **Packeting:** This is performed by Internet Protocol. The network layer converts the packets from its upper layer.
- **Internetworking:** This is the main role of the network layer that it provides the logical connection between different types of networks.
- **Fragmentation:** The fragmentation is a process of breaking the packets into the smallest individual data units that travel through different networks.

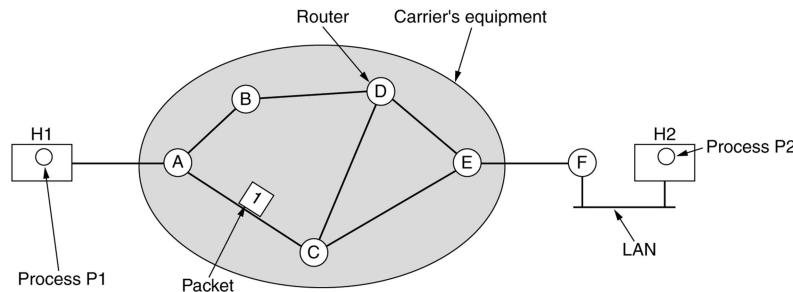
Network Layer Design Issues

The design issues can be elaborated under five heads –

- Store – and – Forward Packet Switching
- Services to Transport Layer
- Providing Connection Oriented Service
- Providing Connectionless Service
- Quality of Service (QOS)

Store – and – Forward Packet Switching

The network layer operates in an environment that uses store and forward packet switching. The node which has a packet to send, delivers it to the nearest router. The packet is stored in the router until it has fully arrived and its checksum is verified for error detection. Once, this is done, the packet is forwarded to the next router. Since, each router needs to store the entire packet before it can forward it to the next hop, the mechanism is called store – and – forward switching.



Services to Transport Layer

The network layer provides service its immediate upper layer, namely transport layer, through the network – transport layer interface. The two types of services provided are –

- Connection – Oriented Service – In this service, a path is setup between the source and the destination, and all the data packets belonging to a message are routed along this path.
- Connectionless Service – In this service, each packet of the message is considered as an independent entity and is individually routed from the source to the destination.

Providing Connection Oriented Service

In connection – oriented services, a path or route called a **virtual circuit** is setup between the source and the destination nodes before the transmission starts. All the packets in the message are sent along this route. Each packet contains an identifier that denotes the virtual circuit to which it belongs to. When all the packets are transmitted, the virtual circuit is terminated and the connection is released. An example of connection – oriented service is Multi-Protocol Label Switching (MPLS).

Providing Connectionless Service

In connectionless service, since each packet is transmitted independently, each packet contains its routing information and is termed as datagram. The network using datagrams for transmission is called datagram networks or datagram subnets. No prior setup of routes are needed before transmitting a message. Each datagram belong to the message follows its own individual route from the source to the destination. An example of connectionless service is Internet Protocol or IP.

Quality of Service (QOS)

Each service can be distinguished by its quality of service. These services can be of two types as explained below –

Reliable Services

Reliable services are those which never lose data. It is usually a reliable service implemented by having the receiver acknowledgements of receipt of each message. Therefore, the sender is sure that it arrives. For example, remote login requires reliable service. But the declarations introduce overheads and delays, which are sometimes undesirable.

Unreliable Services

Unreliable services lose minimal data or bits or pixels of the picture, but there is no significant effect on the result. For example, mobile customers should hear a bit of noise on the line or a misinterpret term from time to time than to learn a delay to wait for acceptance.

Routing Algorithms

Unicast routing

Most of the traffic on the internet and intranets known as unicast data or unicast traffic is sent with specified destination. Routing unicast data over the internet is called unicast routing. It is the simplest form of routing because the destination is already known.

Broadcast routing

By default, the broadcast packets are not routed and forwarded by the routers on any network. Routers create broadcast domains. But it can be configured to forward broadcasts in some special cases. A broadcast message is destined to all network devices.

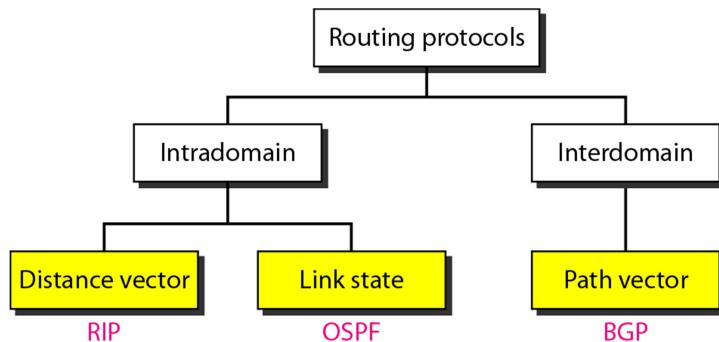
Multicast Routing

Multicast routing is special case of broadcast routing with significance difference and challenges. In broadcast routing, packets are sent to all nodes even if they do not want it. But in Multicast routing, the data is sent to only nodes which wants to receive the packets.

Anycast Routing

Anycast packet forwarding is a mechanism where multiple hosts can have same logical address. When a packet destined to this logical address is received, it is sent to the host which is nearest in routing topology. Anycast routing is done with help of DNS server. Whenever an Anycast packet is received it is enquired with DNS to where to send it. DNS provides the IP address which is the nearest IP configured on it.

Routing algorithms can be divided into two groups:



Shortestpath routing

In this algorithm, to select a route, the algorithm discovers the shortest path between two nodes. It can use multiple hops, the geographical area in kilometers or labeling of arcs for measuring path length.

The labeling of arcs can be done with mean queuing, transmission delay for a standard test packet on an hourly basis, or computed as a function of bandwidth, average distance traffic, communication cost, mean queue length, measured delay or some other factors.

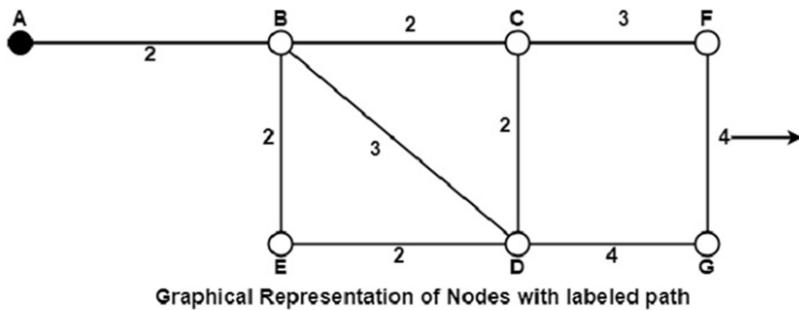
In shortest path routing, the topology communication network is defined using a directed weighted graph. The nodes in the graph define switching components and the directed arcs in the graph define communication connection between switching components. Each arc has a weight that defines the cost of sharing a packet between two nodes in a specific direction.

This cost is usually a positive value that can denote such factors as delay, throughput, error rate, financial costs, etc. A path between two nodes can go through various intermediary nodes and arcs. The goal of shortest path routing is to find a path between two nodes that has the lowest total cost, where the total cost of a path is the sum of arc costs in that path.

For example, Dijikstra uses the nodes labeling with its distance from the source node along the better-known route. Initially, all nodes are labeled with infinity, and as the algorithm proceeds, the label may change. The labeling graph is displayed in the figure.

It can be done in various passes as follows, with A as the source.

- **Pass 1. B (2, A), C(∞ , $-$), F(∞ , $-$), e(∞ , $-$), d(∞ , $-$), G 60**
- **Pass 2. B (2, A), C(4, B), D(5, B), E(4, B), F(∞ , $-$), G(∞ , $-$)**
- **Pass 3. B(2, A), C(4, B), D(5, B), E(4, B), F(7, C), G(9, D)**



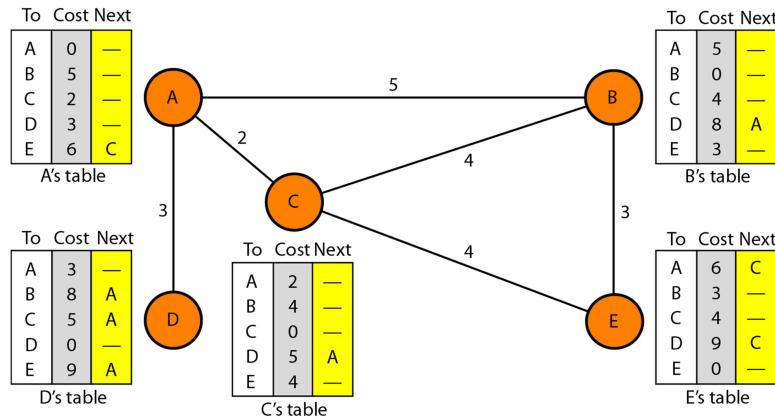
We can see that there can be two paths between A and G. One follows through ABCFG and the other through ABDG. The first one has a path length of 11, while the second one has 9. Hence, the second one, as G (9, D), is selected. Similarly, Node D has also three paths from A as ABD, ABCD and ABED. The first one has a path length of 5 rest two have 6. So, the first one is selected.

All nodes are searched in various passes, and finally, the routes with the shortest path lengths are made permanent, and the nodes of the path are used as a working node for the next round.

Distance Vector Routing

In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node. The table at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing).

We can think of nodes as the cities in an area and the lines as the roads connecting them. A table can show a tourist the minimum distance between cities. In Figure 22.14, we show a system of five nodes with their corresponding tables.



The table for node A shows how we can reach any node from this node. For example, our least cost to reach node E is 6. The route passes through C.

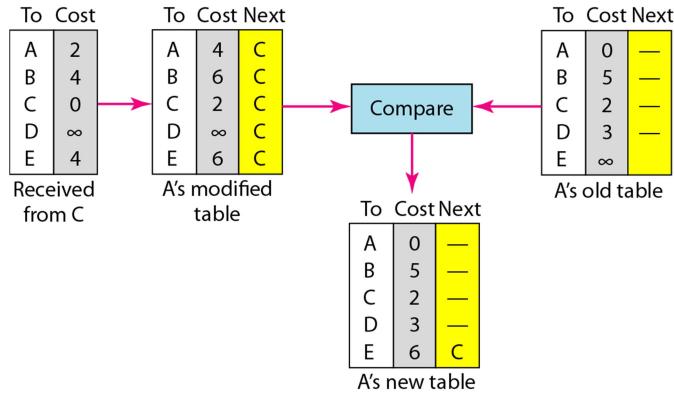
Initialization

The tables in Figure 22.14 are stable; each node knows how to reach any other node and the cost. At the beginning, however, this is not the case. Each node can know only the distance between itself and its immediate neighbors, those directly connected to it. So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors. Figure 22.15 shows the initial tables for each node. The distance for any entry that is not a neighbor is marked as infinite (unreachable).

Sharing

The whole idea of distance vector routing is the sharing of information between neighbors. Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other. There is only one problem. How much of the table must be shared with each neighbor? A node is not aware of a neighbor's table. The best solution for each node is to send its entire table to the neighbor and let the neighbor decide what part to use and what part to discard. However, the third column of a table (next stop) is not useful for the neighbor. When the neighbor receives a table, this column needs to be replaced with the sender's name. If any of the rows can be used, the next node is the sender of the table. A node therefore can send only the first two columns of its table to any neighbor. In other words, sharing means sharing only the first two columns.

Updating



When a node receives a two-column table from a neighbor, it needs to update its routing table. Updating takes three steps:

1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. The logic is clear. If node C claims that its distance to a destination is x mi, and the distance between A and C is y mi, then the distance between A and that destination, via C, is $x + y$ mi.
2. The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.
 - a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.
 - b. If the next-node entry is the same, the receiving node chooses the new row. For example, suppose node C has previously advertised a route to node X with distance 3. Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist anymore. The new route has a distance of infinity.

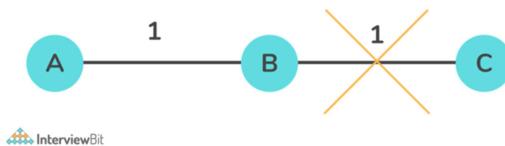
Figure shows how node A updates its routing table after receiving the partial table from node C. There are several points we need to emphasize here. First, as we know from mathematics, when we add any number to infinity, the result is still infinity. Second, the modified table shows how to reach A from A via C. If A needs to reach itself via C, it needs to go to C and come back, a distance of 4. Third, the only benefit from this updating of node A is the last entry, how to reach E. Previously, node A did not know how to reach E (distance of infinity); now it knows that the cost is 6 via C. Each node can update its table by using the tables received from other nodes. In a short time, if there is no change in the network itself, such as a failure in a link, each node reaches a stable condition in which the contents of its table remains the same.

The Count to Infinity Problem in distance vector routing

The crux of the Count to Infinity problem is that if node A tells node B that it has a path somewhere, there is no way for node B to know if the path has node B as a part of it.



Consider the above diagram, for this setup, the Bellman-Ford algorithm will work such that for each router, they will have entries for each other. Router A will infer that it can reach B at a cost of 2 units, and B will infer that it can reach C at a cost of 1 unit.



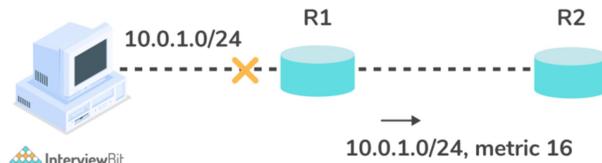
InterviewBit

Consider the case in the above diagram, where the connection between B and C gets disconnected. In this case, B will know that it cannot get to C at a cost of 1 anymore and update its table accordingly. However, it can be possible that A sends some information to B that it is possible to reach C from A at a cost of 2. Then, since B can reach A at a cost of 1, B will erroneously update its table that it can reach C via A at a cost of $1 + 2 = 3$ units. A will then receive updates from B and update its costs to 4, and so on. Thus, the process enters into a loop of bad feedback and the cost shoots towards infinity. This entire situation is called the Count to Infinity problem.

Solution for the Count to Infinity Problem

Route Poisoning:

Route Poisoning is a method used to prevent routers from sending packets through a route that has been deemed invalid within computer networks. Upon failure of a route, Distance Vector Protocols spread the *bad news* about the route failure by poisoning the route. In Route Poisoning, a special metric value called Infinity is used when advertising the route. Routers with a metric of Infinity are considered to have failed. The main disadvantage of this method is that it increases the sizes of routing announcements significantly in many common network topologies.

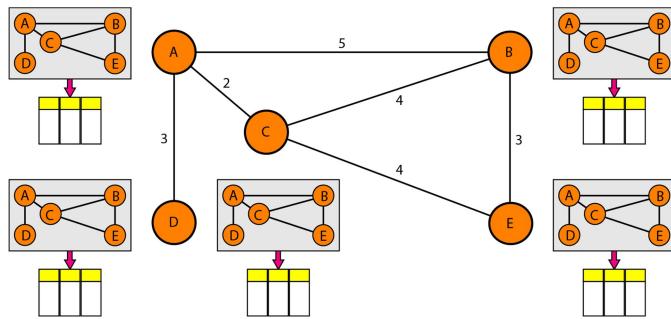


Link State Routing

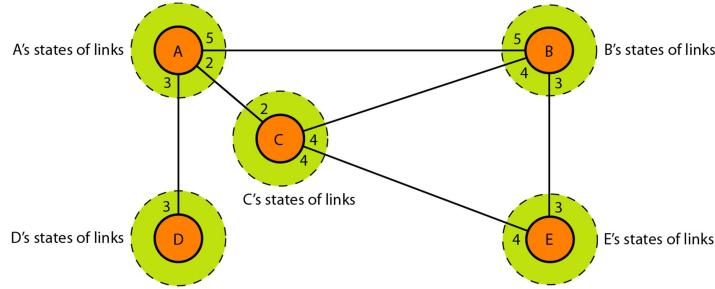
The link state routing is simple and each router has to perform the following operations

- Each router should discover its neighbors and obtain their network addresses.
- Then it should measure the delay or cost to each of these neighbors.
- It should construct a packet containing the network addresses and the delays of all the neighbors.
- Send this packet to all other routers
- Compute the shortest path to every other router.

Sometimes the network becomes so large that the size of the router table becomes excessively large and practically it becomes impossible for every router to have an entry for every other router. Then the hierarchical routing such as the one used in telephone networks should be adopted.



In link state routing, if each node in the domain has the entire topology of the domain. i.e. (the list of nodes and links, how they are connected including the type, cost (metric), and condition of the links (up or down))



Node A knows that it is connected to node B with metric 5, to node C with metric 2, and to node D with metric 3. Node C knows that it is connected to node A with metric 2, to node B with metric 4, and to node E with metric 4. Node D knows that it is connected only to node A with metric 3. And so on

4 Steps in link state routing

1. Creation of Link State Packet (LSP)

Creation of the states of the links by each node, called the link state packet (LSP).

The LSP carries a minimum amount of data: the node identity, the list of links, a sequence number, and age.

LSPs are generated on two occasions:

1. *When there is a change in the topology of the domain.*
2. *On a periodic basis*

2. Flooding of LSPs

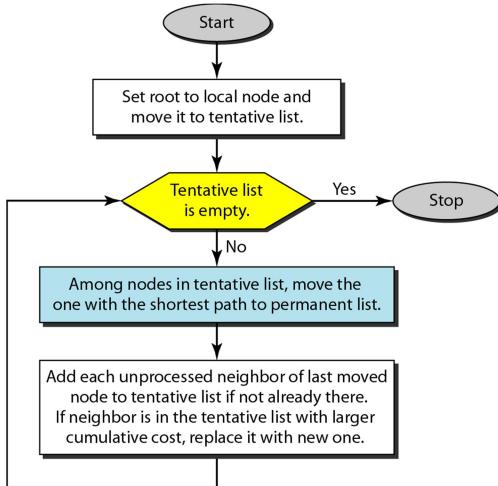
Dissemination of LSPs to every other node / router, called **flooding**, in an efficient and reliable way.

- ❖ The creating node sends a copy of the LSP out of each interface.
- ❖ The receiving node sends a copy of it out of each interface except the one from which the packet arrived.

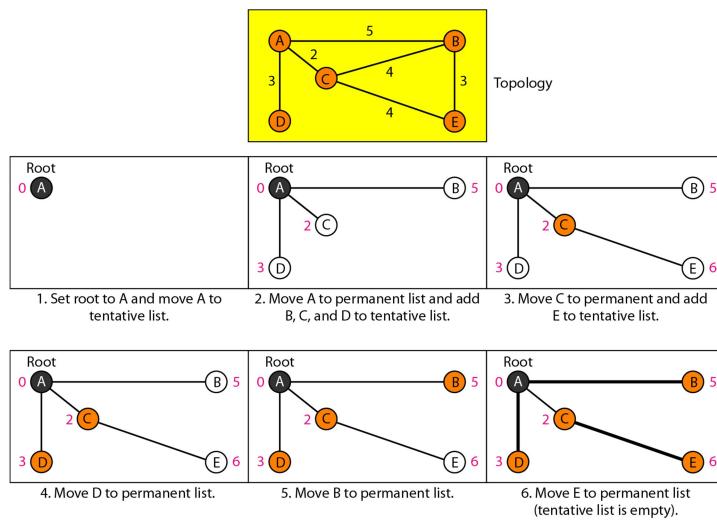
3. Formation of Shortest Path Tree: Dijkstra Algorithm

The topology carried by LSP is not sufficient to find the shortest path to every other node; a shortest path tree is needed. A shortest path tree is a tree in which the path between the root (source) and every other node is the shortest. A tree is a graph of nodes and links; one node is called the root. The Dijkstra algorithm creates a shortest path tree from a graph. The algorithm divides the nodes into two sets: tentative and permanent

Dijkstra algorithm



Example of formation of shortest path tree



4. Calculation of a routing table based on the shortest path tree

Node	Cost	Next Router
A	0	—
B	5	—
C	2	—
D	3	—
E	6	C

Routing table for node A

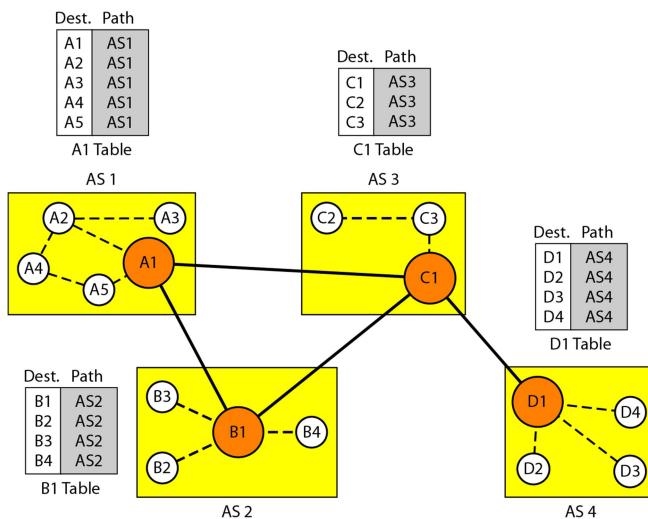
Path Vector Routing

Distance vector and link state routing are both intra-domain routing protocols. They can be used inside an autonomous system, but not between autonomous systems. These two protocols are not suitable for inter-domain routing mostly because of scalability. Both of these routing protocols become intractable when the domain of operation becomes large. Distance vector routing is subject to instability if there are more than a few hops in the domain of operation. Link state routing needs a huge amount of resources to calculate routing tables. It also creates heavy traffic because of flooding. There is a need for a third routing protocol which we call 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 (there can be more, but one is enough for our conceptual discussion) 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. The idea is the same as for distance vector routing except that only speaker nodes in each AS can communicate with each other. However, what is advertised is different. A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems.

Initialization

At the beginning, each speaker node can know only the reachability of nodes inside its autonomous system. Figure 22.30 shows the initial tables for each speaker node in a system made of four ASs.



Node A1 is the speaker node for AS1, B1 for AS2, C1 for AS3, and D1 for AS4. Node A1 creates an initial table that shows A1 to A5 are located in AS1 and can be reached through it. Node B1 advertises that B1 to B4 are located in AS2 and can be reached through B1. And so on.

Sharing Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors. In Figure 22.30, node A1 shares its table with nodes B1 and C1. Node C1 shares its table with nodes D1, B1, and A1. Node B1 shares its table with C1 and A1. Node D1 shares its table with C1.

Updating When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table. After a while each speaker has a table and knows how to reach each node in other ASs. Figure 22.31 shows the tables for each speaker node after the system is stabilized. According to the figure, if router A1 receives a packet for nodes A3, it knows that the path is in AS1 (the packet is at home); but if it receives a packet for D1, it knows

that the packet should go from AS1, to AS2, and then to AS3. The routing table shows the path completely. On the other hand, if node D1 in AS4 receives a packet for node A2, it knows it should go through AS4, AS3, and AS 1.

Loop prevention. The instability of distance vector routing and the creation of loops can be avoided in path vector routing. When a router receives a message, it checks to see if its autonomous system is in the path list to the destination. If it is, looping is involved and the message is ignored.

Dest.	Path	Dest.	Path	Dest.	Path	Dest.	Path
A1 ... A5	AS1	A1 ... A5	AS2-AS1	A1 ... A5	AS3-AS1	A1 ... A5	AS4-AS3-AS1
B1 ... B4	AS1-AS2	B1 ... B4	AS2	B1 ... B4	AS3-AS2	B1 ... B4	AS4-AS3-AS2
C1 ... C3	AS1-AS3	C1 ... C3	AS2-AS3	C1 ... C3	AS3	C1 ... C3	AS4-AS3
D1 ... D4	AS1-AS2-AS4	D1 ... D4	AS2-AS3-AS4	D1 ... D4	AS3-AS4	D1 ... D4	AS4

A1 Table B1 Table C1 Table D1 Table

Policy routing. Policy routing can be easily implemented through path vector routing. When a router receives a message, it can check the path. If one of the autonomous systems listed in the path is against its policy, it can ignore that path and that destination. It does not update its routing table with this path, and it does not send this message to its neighbors.

Optimum path. What is the optimum path in path vector routing? We are looking for a path to a destination that is the best for the organization that runs the autonomous system. We definitely cannot include metrics in this route because each autonomous system that is included in the path may use a different criterion for the metric. One system may use, internally, RIP, which defines hop count as the metric; another may use OSPF with minimum delay defined as the metric. The optimum path is the path that fits the organization. In our previous figure, each autonomous system may have more than one path to a destination. For example, a path from AS4 to AS1 can be AS4-AS3-AS2-AS1, or it can be AS4-AS3-AS1. For the tables, we chose the one that had the smaller number of autonomous systems, but this is not always the case. Other criteria, such as security, safety, and reliability, can also be applied.

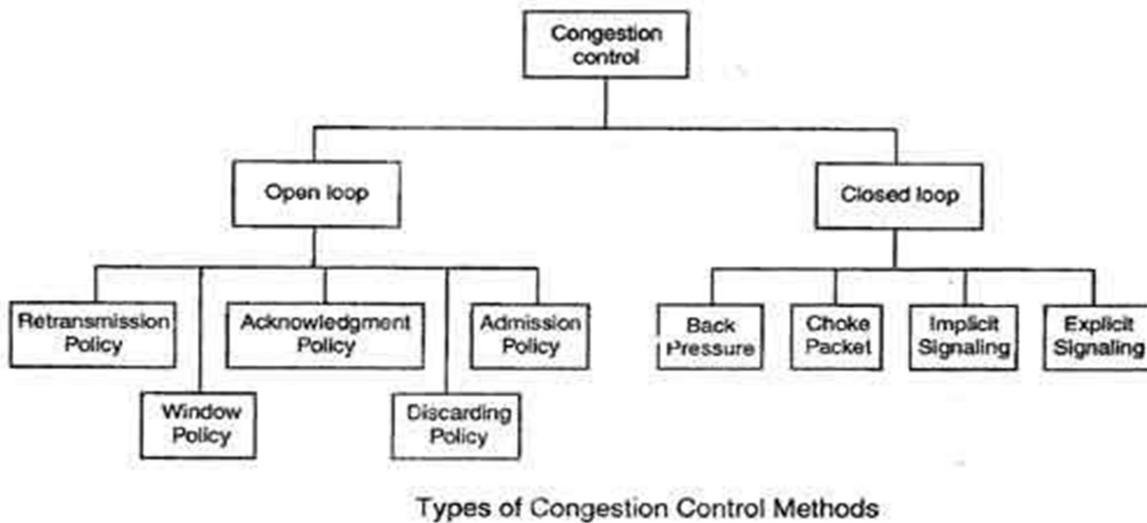
Congestion Control Techniques

Congestion control refers to the mechanisms and techniques used to control congestion and keep the traffic below the capacity of the network. As shown in Fig. , the congestion control techniques can be broadly classified two broad categories:

- **Open loop:** Protocols to prevent or avoid congestion, ensuring that the system (or network under consideration) never enters a Congested State.
- **Close loop:** Protocols that allow system to enter congested state, detect it, and remove it.

Factors that Cause Congestion

- Packet arrival rate exceeds the outgoing link capacity.
- Insufficient memory to store arriving packets
- Bursty traffic
- Slow processor



The various methods used for **open loop congestion control** are:

Retransmission Policy

- The sender retransmits a packet, if it feels that the packet it has sent is lost or corrupted..

The retransmission policy and the retransmission timers need to be designed to optimize efficiency and at the same time prevent the congestion

Window Policy

- To implement window policy, selective reject window method is used for congestion control.
- Selective Reject method is preferred because it sends only the specific lost or damaged packets.

Acknowledgement Policy

- The acknowledgement 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.

Discarding Policy

- A router may discard less sensitive packets when congestion is likely to happen.
- Such a discarding policy may prevent congestion and at the same time may not harm the integrity of the transmission.

Admission Policy

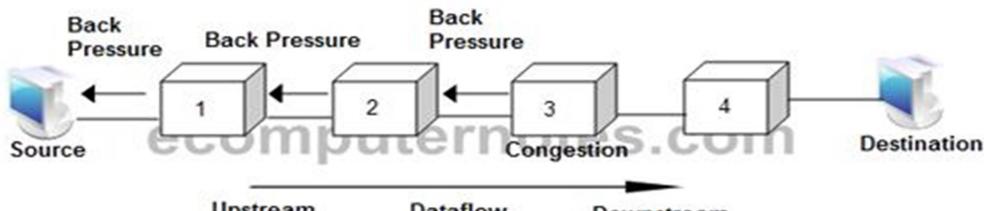
- An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual circuit networks.
- Switches in a flow should first check the resource requirement of a network flow before admitting it to the network.
- A router can deny establishing a virtual circuit connection 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 remove the congestion after it happens.
- The various methods used for closed loop congestion control are:

Backpressure

- Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow.



Backpressure Method

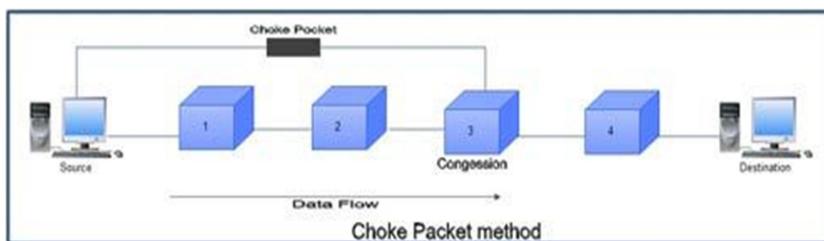
- In this method of congestion control, the congested node stops receiving data from the immediate upstream node or nodes.

This may cause the upstream node or nodes to become congested, and they, in turn, reject data from their upstream node or nodes.

- As shown in fig node 3 is congested and it stops receiving packets and informs its upstream node 2 to slow down. Node 2 in turn may be congested and informs node 1 to slow down. Now node 1 may create congestion and informs the source node to slow down. In this way the congestion is alleviated. Thus, the pressure on node 3 is moved backward to the source to remove the congestion.

Choke Packet

- In this method of congestion control, congested router or node sends a special type of packet called choke packet to the source to inform it about the congestion.
- Here, congested node does not inform its upstream node about the congestion as in backpressure method.
- In choke packet method, congested node sends a warning directly to the source station



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 when it does not receive any acknowledgment. Therefore the delay in receiving an acknowledgment is interpreted as congestion in the network.
- On sensing this congestion, the source slows down.
- This type of congestion control policy is used by TCP.

Explicit Signaling

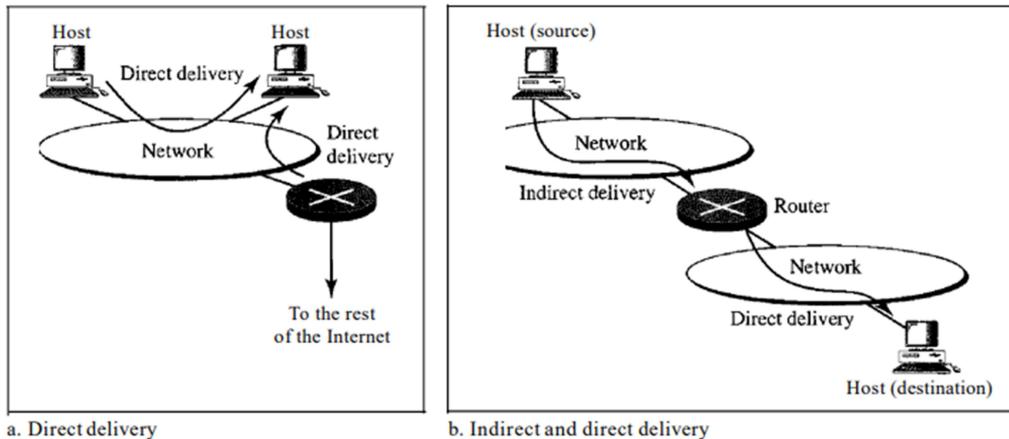
- In this method, the congested nodes explicitly send a signal to the source or destination to inform about the congestion.
- Explicit signaling is different from the choke packet method. In choke packed method, a separate packet is used for this purpose whereas in explicit signaling method, the signal is included in the packets that carry data .
- Explicit signaling can occur in either the forward direction or the backward direction .
- In backward signaling, a bit is set in a packet moving in the direction opposite to the congestion. This bit warns the source about the congestion and informs the source to slow down.
- In forward signaling, a bit is set in a packet moving in the direction of congestion. This bit warns the destination about the congestion.

Host to Host Delivery

Delivery refers to the way a packet is handled by the underlying networks under the control of the network layer. Forwarding refers to the way a packet is delivered to the next station. Routing refers to the way routing tables are created to help in forwarding. Routing protocols are used to continuously update the routing tables that are consulted for forwarding and routing.

Direct Versus Indirect Delivery

Figure *Direct and indirect delivery*



The delivery of a packet to its final destination is accomplished by using two different methods of delivery, direct and indirect, as shown in Figure below.

Direct Delivery

In a direct delivery, the final destination of the packet is a host connected to the same physical network as the deliverer. Direct delivery occurs when the source and destination of the packet are located on the same physical network or when the delivery is between the last router and the destination host. The sender can easily determine if the delivery is direct. It can extract the network address of the destination (using the mask) and compare this address with the addresses of the networks to which it is connected. If a match is found, the delivery is direct.

Indirect Delivery

If the destination host is not on the same network as the deliverer, the packet is delivered indirectly. In an indirect delivery, the packet goes from router to router until it reaches the one connected to the same physical network as its final destination. Note that a delivery always

involves one direct delivery but zero or more indirect deliveries. Note also that the last delivery is always a direct delivery.

FORWARDING

Forwarding means to place the packet in its route to its destination. Forwarding requires a host or a router to have a routing table. When a host has a packet to send or when a router has received a packet to be forwarded, it looks at this table to find the route to the final destination.

Routing Table A host or a router has a routing table with an entry for each destination, or a combination of destinations, to route IP packets. The routing table can be either static or dynamic.

Static Routing Table A static routing table contains information entered manually. The administrator enters the route for each destination into the table. When a table is created, it cannot update automatically when there is a change in the Internet. The table must be manually altered by the administrator.

A static routing table can be used in a small internet that does not change very often, or in an experimental internet for troubleshooting. It is poor strategy to use a static routing table in a big internet such as the Internet.

Dynamic Routing Table A dynamic routing table is updated periodically by using one of the dynamic routing protocols such as RIP, OSPF, or BGP. Whenever there is a change in the Internet, such as a shutdown of a router or breaking of a link, the dynamic routing protocols update all the tables in the routers (and eventually in the host) automatically. The routers in a big internet such as the Internet need to be updated dynamically for efficient delivery of the IP packets

Format

Common fields in a routing table

Mask	Network address	Next-hop address	Interlace	Flags	Reference count	Use

A routing table has a minimum of four columns. However, some of today's routers have even more columns. We should be aware that the number of columns is vendor-dependent, and not all columns can be found in all routers. Figure below shows some common fields in today's routers.

- **Mask.** This field defines the mask applied for the entry.
- **Network address.** This field defines the network address to which the packet is finally delivered. In the case of host-specific routing, this field defines the address of the destination host.
- **Next-hop address.** This field defines the address of the next-hop router to which the packet is delivered.
- **Interface.** This field shows the name of the interface.
- **Flags.** This field defines up to five flags. Flags are on/off switches that signify either presence or absence. The five flags are U (up), G (gateway), H (host-specific), D (added by redirection), and M (modified by redirection).
- **Reference count.** This field gives the number of users of this route at the moment. For example, if five people at the same time are connecting to the same host from this router,

the value of this column is 5.

- **Use.** This field shows the number of packets transmitted through this router for the corresponding destination

Internetworking

The physical and data link layers of a network operate locally. These two layers are jointly responsible for data delivery on the network from one node to the next, as shown in Figure below. This internetwork is made of five networks: four LANs and one WAN. If host A needs to send a data packet to host D, the packet needs to go first from A to R1 (a switch or router), then from R1 to R3, and finally from R3 to host D. We say that the data packet passes through three links. In each link, two physical and two data link layers are involved

Need for Network Layer

The network layer is responsible for host-to-host delivery and for routing the packets through the routers or switches. Figure below shows the same internetwork with a network layer added

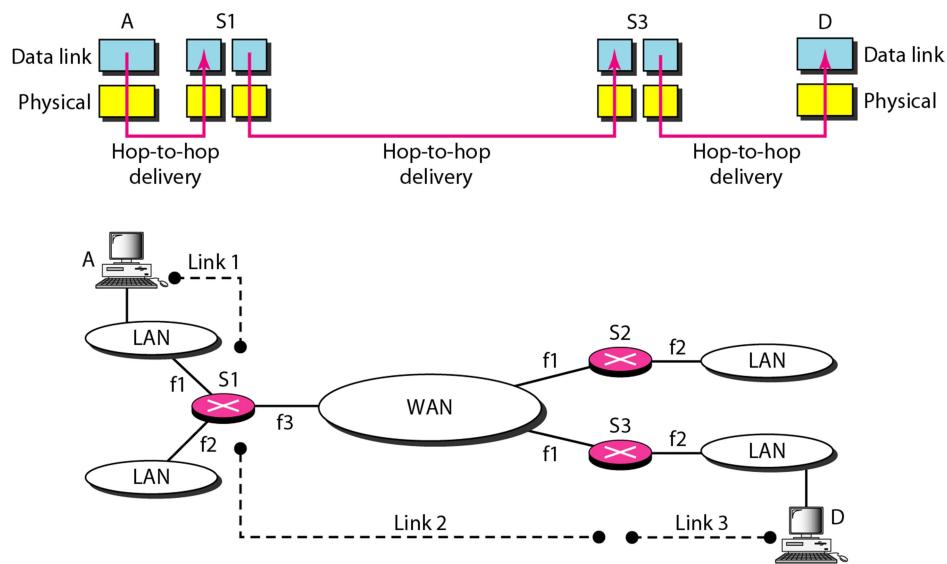
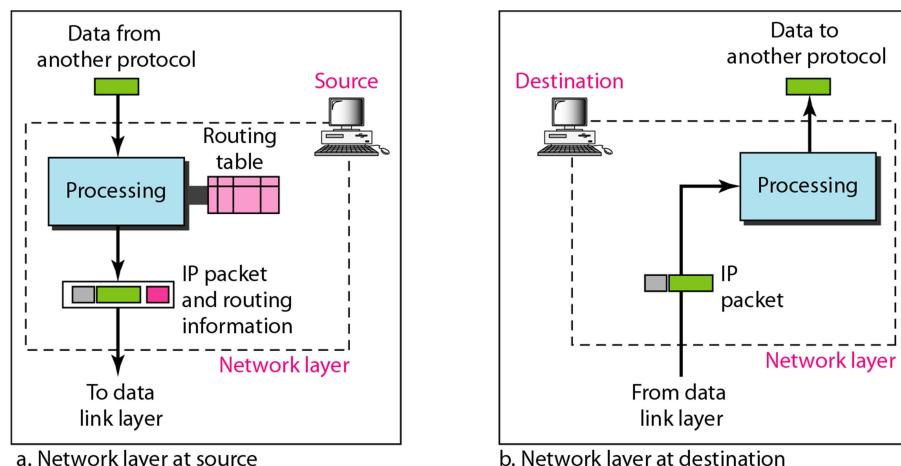
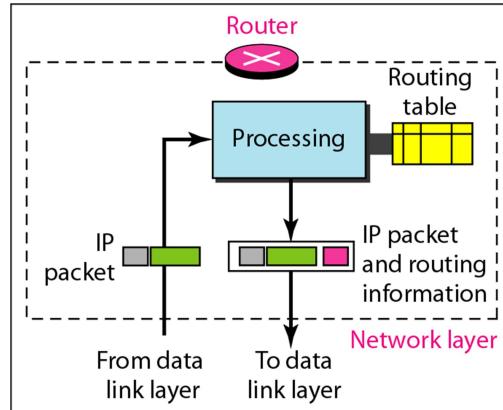


Figure below shows the general idea of the functionality of the network layer at a source, at a router, and at the destination.



The network layer at the source is responsible for creating a packet from the data coming from another protocol (such as a transport layer protocol or a routing protocol). The header of the packet contains, among other information, the logical addresses of the source and destination. The network layer is responsible for checking its routing table to find the routing information (such as the outgoing interface of the packet or the physical address of the next node). If the packet is too large, the packet is fragmented

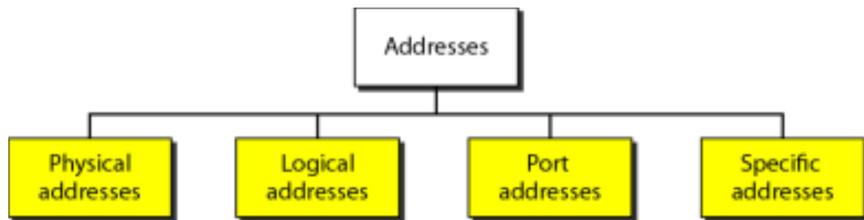


c. Network layer at a router

The network layer at the switch or router is responsible for routing the packet. When a packet arrives, the router or switch consults its routing table and finds the interface from which the packet must be sent. The packet, after some changes in the header, with the routing information is passed to the data link layer again. The network layer at the destination is responsible for address verification; it makes sure that the destination address on the packet is the same as the address of the host. If the packet is a fragment, the network layer waits until all fragments have arrived, and then reassembles them and delivers the reassembled packet to the transport layer.

Addressing

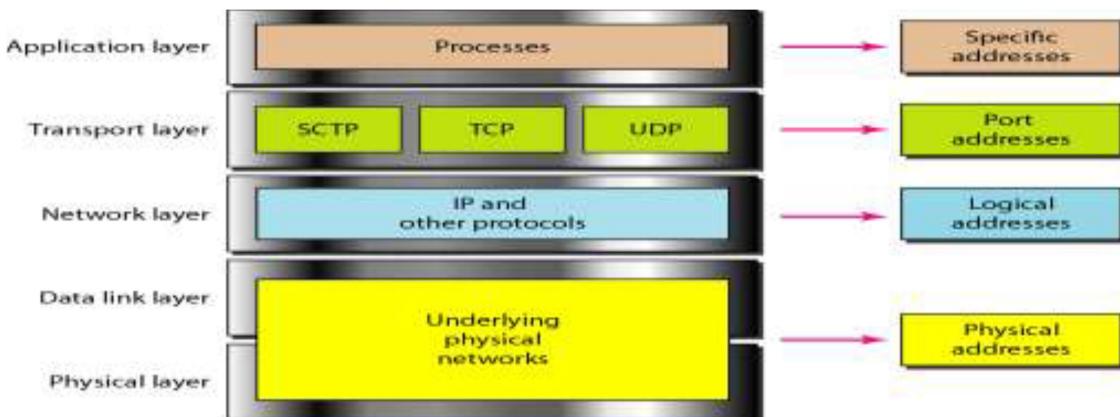
Four levels of addresses are used in an internet employing the *TCP/IP* protocols: physical (link) addresses, logical (IP) addresses, port addresses, and specific addresses.



Physical Addresses:

The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN. It is included in the frame used by the data link layer. It is the lowest level address. The physical addresses have authority over the network (LAN or WAN). The size and format of these addresses vary depending on the network.

Relationship of layers and addresses in TCP/IP:



For example, Ethernet uses a 6-byte (48-bit) physical address that is imprinted on the network interface card (NIC). Most local-area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below:

07:01:02:01:2C:4B

This is a 6-byte (12 hexadecimal digits) physical address.

Logical Addresses:

Logical addresses are necessary for universal communications that are independent of underlying physical networks. Physical addresses are not adequate in an internetwork environment where different networks can have different address formats. A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network. The logical addresses are designed for this purpose. A logical address in the Internet is currently a 32-bit address that can uniquely define a host connected to the Internet.

Port Addresses:

The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. A system that sends nothing but data from one computer to another is not complete. Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process.

For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes. In other words, they need addresses. In the TCP/IP architecture, the label assigned to a process is called a port address. A port address in TCP/IP is 16 bits in length.

753

A 16-bit port address represented as one single number.

Specific Addresses:

Some applications have user-friendly addresses that are designed for that specific address. Examples include the e-mail address (for example, forouzan@fhda.edu) and the Universal Resource Locator (URL) (for example, www.mhhe.com).

Difference between logical, physical, port addresses

Through logical address the system identify a network (source to destination). after identifying the network physical address is used to identify the host on that network. The port address is used to identify the particular application running on the destination machine.

Logical Address: An IP address of the system is called logical address. This address is the combination of Net ID and Host ID. This address is used by network layer to identify a particular network (source to destination) among the networks. This address can be changed by changing the host position on the network. So it is called logical address.

Physical address: Each system having a NIC(Network Interface Card) through which two systems physically connected with each other with cables. The address of the NIC is called Physical address or mac address. This is specified by the manufacturer company of the card. This address is used by data link layer.

Port Address: There are many application running on the computer. Each application run with a port no.(logically) on the computer. This port no. for application is decided by the Kernal of the OS. This port no. is called port address.

IPv4 Addressing

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device. IPv4 addresses are unique.

Address Space

- An IPv4 address is 32 bits long.
- The address space of IPv4 is 2^{32} (4,294,967,296)

A protocol such as IPv4 that defines addresses has an address space. An address space is the total number of addresses used by the protocol. If a protocol uses N bits to define an address, the address space is 2^N because each bit can have two different values (0 or 1) and N bits can have 2^N values. IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than 4 billion). This means that, theoretically, if there were no restrictions, more than 4 billion devices could be connected to the Internet. We will see shortly that the actual number is much less because of the restrictions imposed on the addresses.

Notations

There are two prevalent notations to show an IPv4 address: binary notation and dotted decimal notation.

Binary Notation In binary notation

The IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte. So it is common to hear an IPv4 address referred to as a 32-bit address or a 4-byte address. The following is an example of an IPv4 address in binary notation:

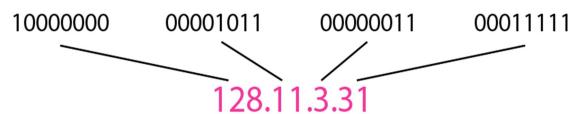
01110101 10010101 00011101 00000010

Dotted-Decimal Notation

To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. The following is the dotted-decimal notation of the above address:

117.149.29.2

Figure below shows an IPv4 address in both binary and dotted-decimal notation



Types of Addressing:

- Classful Addressing
- Classless Addressing

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.

We can find the class of an address when given the address in binary notation or dotted-decimal notation. If the address is given in binary notation, the first few bits can immediately tell us the class of the address. If the address is given in decimal-dotted notation, the first byte defines the class.

	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

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

Table 19.1 Number of blocks and block size in classful IPv4 addressing

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

1) In Classful addressing it divides IP address into network ID and host ID for example Class A :- has first octet as network ID and last three octet as Host ID

Class B :- has first two as network ID and last three two as Host ID

Class C :- has first three as network ID and last octet as Host ID

- 2) Here class A,B and C are used unicast where as class D is used for multicasting and class E for research
- 3) Disadvantage is that it limit the number of network that can be provided to the network
- 4) Example RIP (Routing Information Protocol) protocol uses classful addressing
- 5) Same Subnet mask is used in complete network

Class	Binary	Dotted-Decimal	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

Net ID and Host ID

Classless Addressing

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. 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.
- 1) It allows us to use variable length subnet mask so also known as VLSM (Variable Length Subnet Mask)
 - 2) Different subnet mask used in same network.
 - 3) In this there is no boundary on host id and network id
 - 4) Classless Addressing also known as CIDR(classless interdomain routing)
 - 5) There is no default subnet mask in classless routing.
 - 6) Example: BGP(Border Gateway Protocol),RIPv2

IPv6 Addresses

IPv6 addresses consist of 128 bits, instead of 32 bits, and include a scope field that identifies the type of application suitable for the address. IPv6 does not support broadcast addresses, but instead uses multicast addresses for broadcast. In addition, IPv6 defines a new type of address called anycast

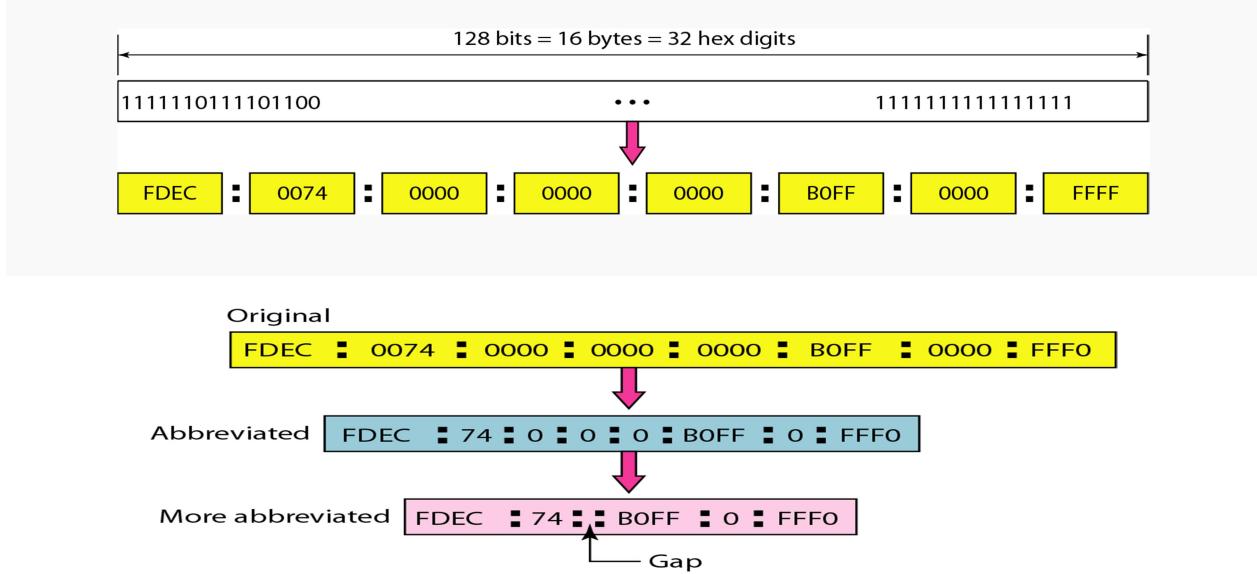
Structure

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

Hexadecimal Colon Notation

IPv6 specifies hexadecimal colon notation. In this notation, 128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal

digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon.



Address Space

IPv6 has a much larger address space; 2^{128} addresses are available. The designers of IPv6 divided the address into several categories. A few leftmost bits, called the type prefix, in each address define its category.

Unicast address

A **unicast address** defines a single computer. The packet sent to a unicast address must be delivered to that specific computer. IPv6 defines two types of unicast addresses: geographically based and provider-based.

Multicast Addresses

Multicast addresses are used to define a group of hosts instead of just one. A packet sent to a multicast address must be delivered to each member of the group.

Anycast Addresses

IPv6 also defines anycast addresses. An anycast address, like a multicast address, also defines a group of nodes. 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).

Reserved Addresses

Another category in the address space is the reserved address. These addresses start with eight 0s (type prefix is 00000000).

Local Addresses These addresses are used when an organization wants to use IPv6 protocol without being connected to the global Internet. In other words, they provide addressing for private networks.

IPV4 PROTOCOL

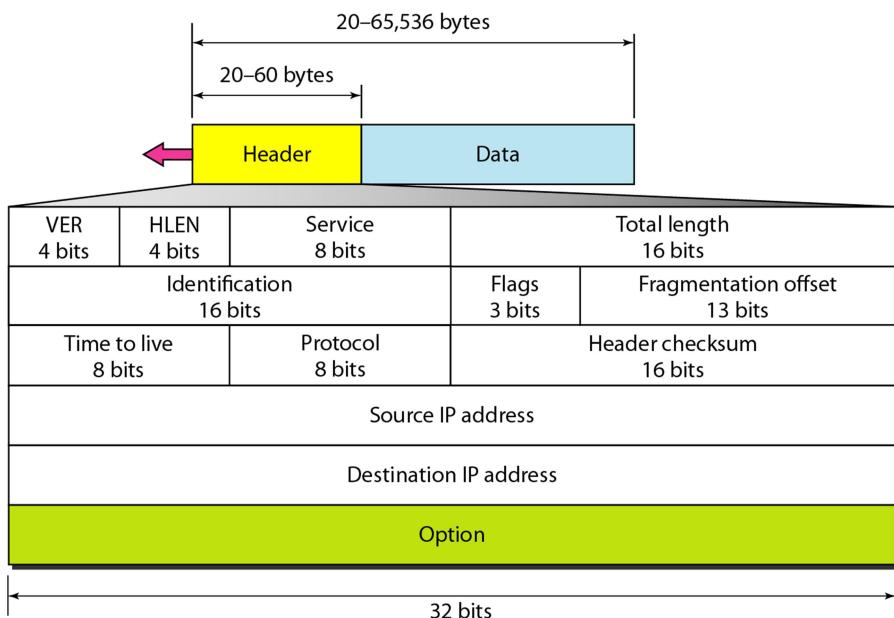
The Internet Protocol version 4 (IPv4) is the delivery mechanism used by the TCP/IP protocols. IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service. The term best-effort means that IPv4 provides no error control or flow control (except for error detection on the header). IPv4 assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees.

If reliability is important, IPv4 must be paired with a reliable protocol such as TCP. An example of a more commonly understood best-effort delivery service is the post office. The post office does its best to deliver the mail but does not always succeed. If an unregistered letter is lost, it is up to the sender or would-be recipient to discover the loss and rectify the problem. The post office itself does not keep track of every letter and cannot notify a sender of loss or damage.

IPv4 is also a connectionless protocol for a packet-switching network that uses the datagram approach. This means that each datagram is handled independently, and each datagram can follow a different route to the destination. This implies that datagrams sent by the same source to the same destination could arrive out of order. Also, some could be lost or corrupted during transmission. Again, IPv4 relies on a higher-level protocol to take care of all these problems.

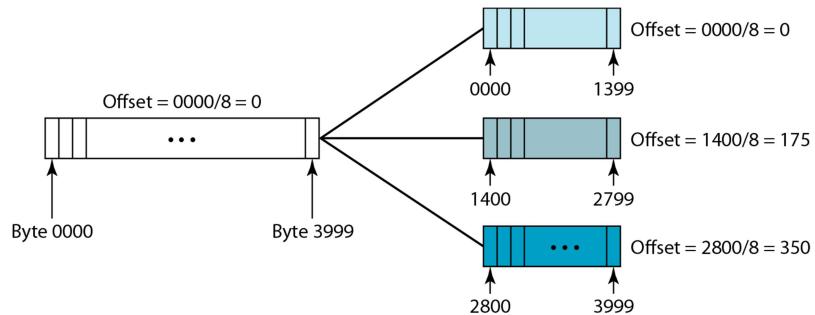
IP Datagram

Packets in the IPv4 layer are called datagrams. Figure below shows the IPv4 datagram format.

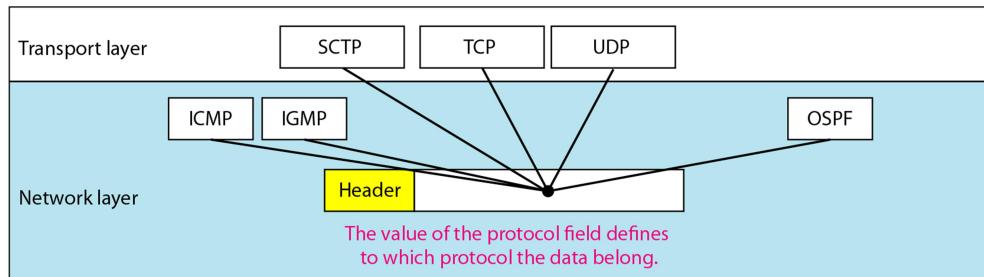


- **Version** – Version no. of Internet Protocol used (e.g. IPv4).
- **IHL** – Internet Header Length; Length of entire IP header.
- **DSCP** – Differentiated Services Code Point; this is Type of Service .It define the level of service a packet receives in the network.
- **ECN** – Explicit Congestion Notification; It carries information about the congestion seen in the route.
- **Total Length** – Length of entire IP Packet (including IP header and IP Payload).
- **Identification** – If IP packet is fragmented during the transmission, all the fragments contain same identification number. to identify original IP packet they belong to.

- **Flags** – As required by the network resources, if IP Packet is too large to handle, these ‘flags’ tells if they can be fragmented or not. In this 3-bit flag, the MSB is always set to ‘0’.
- **Time to Live** – To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded.
- **Fragment Offset** – This offset tells the exact position of the fragment in the original IP Packet. This 13-bit field shows the relative position of this fragment with respect to the whole datagram. It is the offset of the data in the original datagram measured in units of 8 bytes. Figure below shows a datagram with a data size of 4000 bytes fragmented into three fragments



- **Protocol** – Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol. For example protocol number of ICMP is 1, TCP is 6 and UDP is 17.



- **Header Checksum** – This field is used to keep checksum value of entire header which is then used to check if the packet is received error-free.

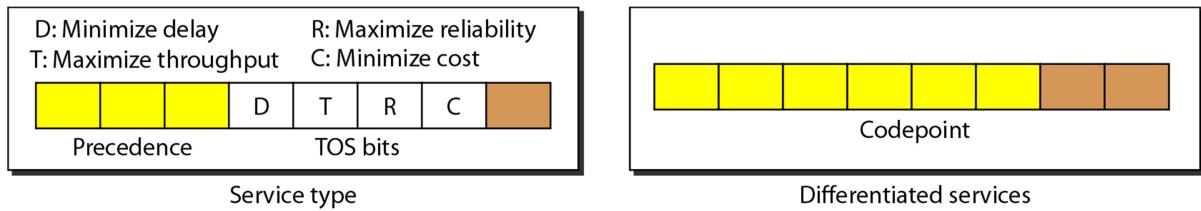
4	5	0	28								
1			0 0 0 0								
4	17		0 0 0 0								
10.12.14.5											
12.6.7.9											
4, 5, and 0	→	4	5	0	0						
28	→	0	0	1	C						
1	→	0	0	0	1						
0 and 0	→	0	0	0	0						
4 and 17	→	0	4	1	1						
0	→	0	0	0	0						
10.12	→	0	A	0	C						
14.5	→	0	E	0	5						
12.6	→	0	C	0	6						
7.9	→	0	7	0	9						
Sum	→	7	4	4	E						
Checksum	→	8	B	B	1						

The implementation of the checksum in the IPv4 packet follows the same principles. 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** – 32-bit address of the Sender (or source) of the packet.
- **Destination Address** – 32-bit address of the Receiver (or destination) of the packet.
- **Options** – This is optional field, which is used if the value of IHL is greater than 5. These options may contain values for options such as Security, Record Route, Time Stamp, etc.

Services:

IETF has changed the interpretation and name of this 8-bit field. This field, previously called service type, is now called differentiated services. We show both interpretations in Figure below



1. Service Type

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

a. Precedence is a 3-bit subfield ranging from 0 (000 in binary) to 7 (111 in binary). The precedence defines the priority of the datagram in issues such as congestion. If a router is congested and needs to discard some datagrams, those datagrams with lowest precedence are discarded first. Some datagrams in the Internet are more important than others. For example, a datagram used for network management is much more urgent and important than a datagram containing optional information for a group.

b. TOS bits is a 4-bit subfield with each bit having a special meaning. Although a bit can be either 0 or 1, one and only one of the bits can have the value of 1 in each datagram. The bit patterns and their interpretations are given in Table 20.1. With only 1 bit set at a time, we can have five different types of services.

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

2. **Differentiated Services** In this interpretation, the first 6 bits make up the codepoint subfield, and the last 2 bits are not used. The codepoint subfield can be used in two different ways.
- When the 3 rightmost bits are Os, the 3 leftmost bits are interpreted the same as the precedence bits in the service type interpretation. In other words, it is compatible with the old interpretation
 - When the 3 rightmost bits are not all Os, the 6 bits define 64 services based on the priority assignment by the Internet or local authorities according to Table 20.3. The first category contains 32 service types; the second and the third each contain 16. The first category (numbers 0, 2, 4, ..., 62) is assigned by the Internet authorities (IETF). The second category (3, 7, 11, 15, ..., 63) can be used by local authorities (organizations). The third category (1, 5, 9, ..., 61) is temporary and can be used for experimental purposes. Note that the numbers are not contiguous. If they were, the first category would range from 0 to 31, the second from 32 to 47, and the third from 48 to 63. This would be incompatible with the TOS interpretation because XXXOOO (which includes 0, 8, 16, 24, 32, 40, 48, and 56) would fall into all three categories.

Table 20.3 *Values for codepoints*

<i>Category</i>	<i>Codepoint</i>	<i>Assigning Authority</i>
1	XXXXXX0	Internet
2	XXXXX11	Local
3	XXXXOI	Temporary or experimental

IPv6 PROTOCOL

IPv4 has some deficiencies (listed below) that make it unsuitable for the fast-growing Internet.

- Despite all short-term solutions, such as subnetting, classless addressing, and NAT, address depletion is still a long-term problem in the Internet.
- The Internet must accommodate real-time audio and video transmission. This type of transmission requires minimum delay strategies and reservation of resources not provided in the IPv4 design.
- The Internet must accommodate encryption and authentication of data for some applications. No encryption or authentication is provided by IPv4.

In IPv6, the Internet protocol was extensively modified to accommodate the unforeseen growth of the Internet. The format and the length of the IP address were changed along with the packet format. Related protocols, such as ICMP, were also modified. Other protocols in the network layer, such as ARP, RARP, and IGMP, were either deleted or included in the ICMPv6 protocol

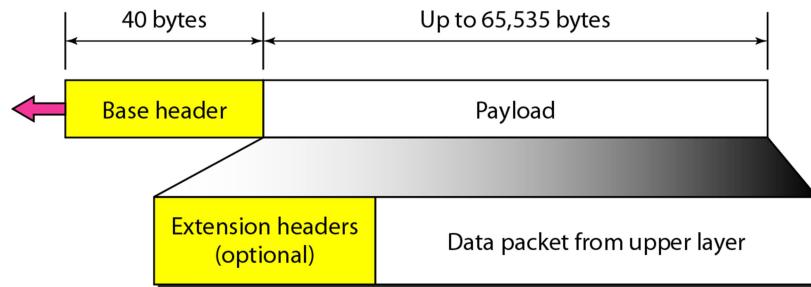
Advantages

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

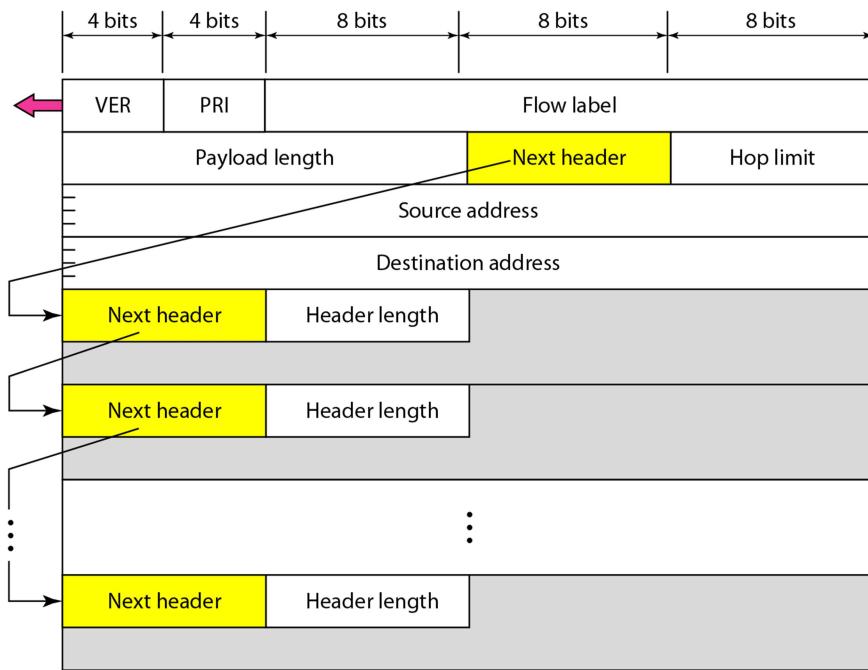
- Larger address space.** An IPv6 address is 128 bits long. Compared with the 32-bit address of IPv4, this is a huge (296) increase in the address space.
- Better header format.** IPv6 uses a new header format in which options are separated from the base header and inserted, when needed, between the base header and the upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by routers.
- 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.
- **Support for resource allocation.** In IPv6, the type-of-service field has been removed, but a mechanism (called low label) has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as real-time audio and video.
- **Support for more security.** The encryption and authentication options in IPv6 provide confidentiality and integrity of the packet.

IP DATAGRAM FORMAT



BASE HEADER FORMAT



- **Version** – 4-bit version number of Internet Protocol = 6.
- **Traffic class** – 8-bit traffic class field. It helps routers to handle the traffic based on priority of the packet. If congestion occurs on router then packets with least priority will be discarded.
- **Flow label** – 20-bit field. Flow Label field is used by source to label the packets belonging to the same flow in order to request special handling by intermediate IPv6 routers
- **Payload length** – It is a 16-bit (unsigned integer) field, indicates total size of the payload which tells routers about amount of information a particular packet contains in its payload.

- **Next header** – Next Header indicates type of extension header (if present) immediately following the IPv6 header. It is an 8-bit field defining the header that follows the base header in the datagram.

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

- **Hop limit** – Hop Limit field is same as TTL in IPv4 packets. It indicates the maximum number of intermediate nodes IPv6 packet is allowed to travel. Its value gets decremented by one, by each node that forwards the packet and packet is discarded if value decrements to 0.
- **Source address** – 128 bits. The address of the initial sender of the packet.
- **Destination address** – 128 bits. The address of the intended recipient of the packet.

IPv6 introduces the concept of (optional) **extension headers**. These headers can be supplied to provide extra information, but encoded in an efficient way. Six kinds of extension headers are defined at present, as listed in Fig

Ext. Header	Description
Hop-by-Hop Options	Examined by all devices on the path
Destination Options (with routing options)	Examined by destination of the packet
Routing Header	Methods to take routing decision
Fragment Header	Contains parameters of fragmented datagram done by source
Authentication Header	verify authenticity
Encapsulating Security Payload	Carries Encrypted data

Comparison between IPv4 and IPv6 Headers

<i>Comparison</i>
1. The header length field is eliminated in IPv6 because the length of the header is fixed in this version.
2. The service type field is eliminated in IPv6. The priority and flow label fields together take over the function of the service type field.
3. The total length field is eliminated in IPv6 and replaced by the payload length field.
4. The identification, flag, and offset fields are eliminated from the base header in IPv6. They are included in the fragmentation extension header.
5. The TTL field is called hop limit in IPv6.
6. The protocol field is replaced by the next header field.
7. The header checksum is eliminated because the checksum is provided by upper-layer protocols; it is therefore not needed at this level.
8. The option fields in IPv4 are implemented as extension headers in IPv6.

ADDRESS MAPPING (ARP)

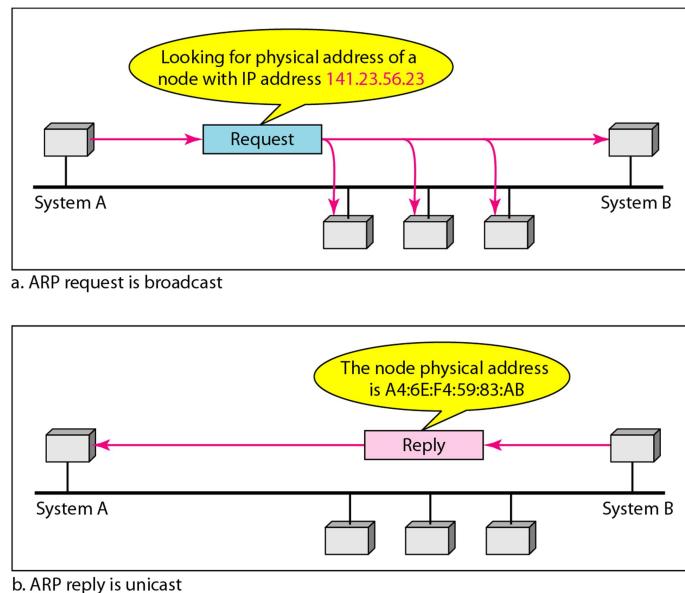
An internet is made of a combination of physical networks connected by internetworking devices such as routers. A packet starting from a source host may pass through several different physical networks before finally reaching the destination host. The hosts and routers are recognized at the network level by their logical (IP) addresses. However, packets pass through physical networks to reach these hosts and routers. At the physical level, the hosts and routers are recognized by their physical addresses.

This means that delivery of a packet to a host or a router requires two levels of addressing: logical and physical. We need to be able to map a logical address to its corresponding physical address and vice versa. These can be done by using either static or dynamic mapping.

Mapping Logical to Physical Address: ARP

Anytime a host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver. The logical (IP) address is obtained from the DNS (see Chapter 25) if the sender is the host or it is found in a routing table if the sender is a router. But the IP datagram must be encapsulated in a frame to be able to pass through the physical network. This means that the sender needs the physical address of the receiver. The host or the router sends an ARP query packet. The packet includes the physical and IP addresses of the sender and the IP address of the receiver. Because the sender does not know the physical address of the receiver, the query is broadcast over the network

Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet. The response packet contains the recipient's IP and physical addresses. The packet is unicast directly to the inquirer by using the physical address received in the query packet.

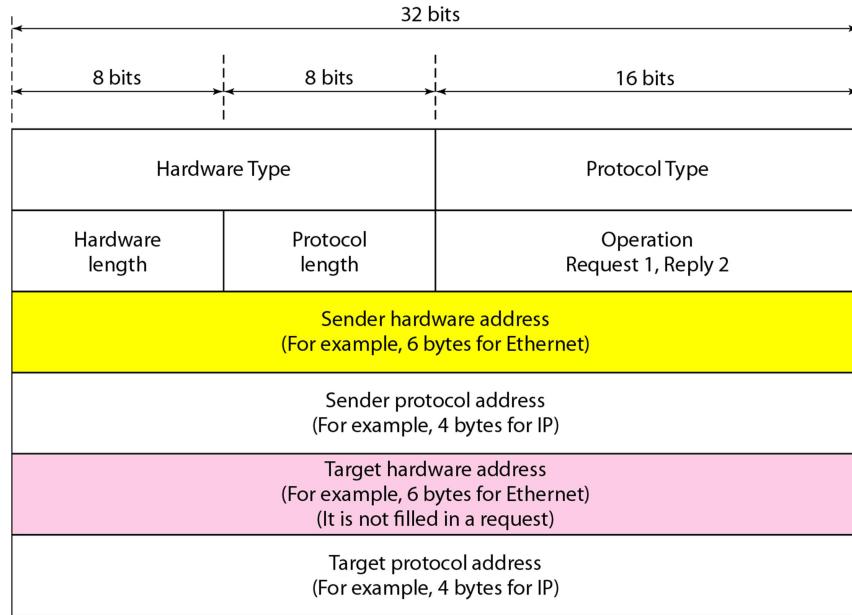


In Figure, the system on the left (A) has a packet that needs to be delivered to another system (B) with IP address 141.23.56.23. System A needs to pass the packet to its data link layer for the actual delivery, but it does not know the physical address of the recipient. It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of 141.23.56.23.

This packet is received by every system on the physical network, but only system B will answer it, as shown in above figure. System B sends an ARP reply packet that includes its physical address. Now system A can send all the packets it has for this destination by using the physical address it received.

Packet Format

Figure below shows the format of an ARP packet

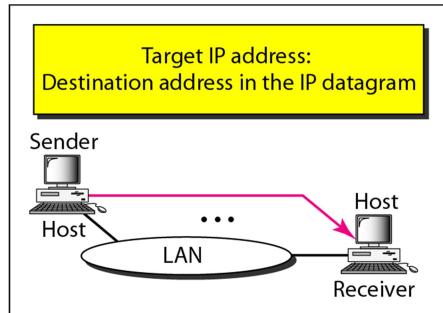


The fields are as follows:

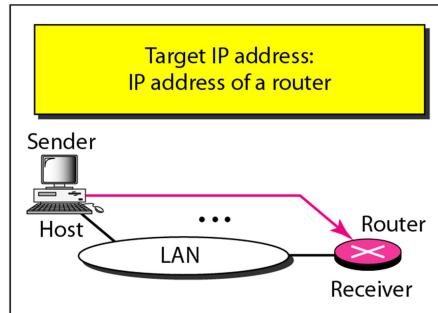
- Hardware type. This is a 16-bit field defining the type of the network on which ARP is running. Each LAN has been assigned an integer based on its type. For example, Ethernet is given type 1. ARP can be used on any physical network.
- Protocol type. This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is 080016, ARP can be used with any higher-level protocol.
- Hardware length. This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.
- Protocol length. This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.
- Operation. This is a 16-bit field defining the type of packet. Two packet types are defined: ARP request (1) and ARP reply (2).
- Sender hardware address. This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.
- Sender protocol address. This is a variable-length field defining the logical (for example, IP) address of the sender. For the IP protocol, this field is 4 bytes long.
- Target hardware address. This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long. For an ARP request message, this field is all 0s because the sender does not know the physical address of the target.
- Target protocol address. This is a variable-length field defining the logical (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long

Four different cases

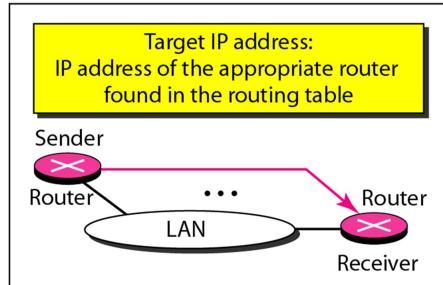
The following are four different cases in which the services of ARP can be used



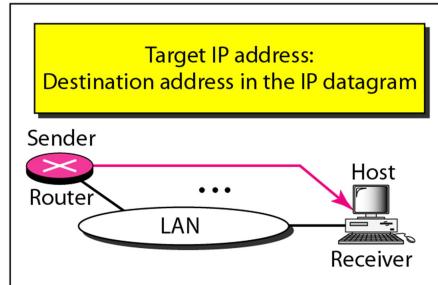
Case 1. A host has a packet to send to another host on the same network.



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



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



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

1. The sender is a host and wants to send a packet to another host on the same network. In this case, the logical address that must be mapped to a physical address is the destination IP address in the datagram header
2. The sender is a host and wants to send a packet to another host on another network. In this case, the host looks at its routing table and finds the IP address of the next hop (router) for this destination. If it does not have a routing table; it looks for the IP address of the default router. The IP address of the router becomes the logical address that must be mapped to a physical address.
3. The sender is a router that has received a datagram destined for a host on another network. It checks its routing table and finds the IP address of the next router. The IP address of the next router becomes the logical address that must be mapped to a physical address.
4. The sender is a router that has received a datagram destined for a host on the same network. The destination IP address of the datagram becomes the logical address that must be mapped to a physical address.

Internet Control Message Protocol (ICMP)

The IP provides unreliable and connectionless datagram delivery. It was designed this way to make efficient use of network resources. The IP protocol is a best-effort delivery service that delivers a datagram from its original source to its final destination. However, it has two deficiencies: lack of error control and lack of assistance mechanisms.

The IP protocol has no error-reporting or error-correcting mechanism. What happens if something goes wrong? What happens if a router must discard a datagram because it cannot find a router to the final destination, or because the time-to-live field has a zero value? What happens if the final destination host must discard all fragments of a datagram because it has not received all fragments within a predetermined time limit? These are examples of situations where an error has occurred and the IP protocol has no built-in mechanism to notify the original host.

The IP protocol also lacks a mechanism for host and management queries. A host sometimes needs to determine if a router or another host is alive. And sometimes a network administrator needs information from another host or router.

The Internet Control Message Protocol (ICMP) has been designed to compensate for the above two deficiencies. It is a companion to the IP protocol.

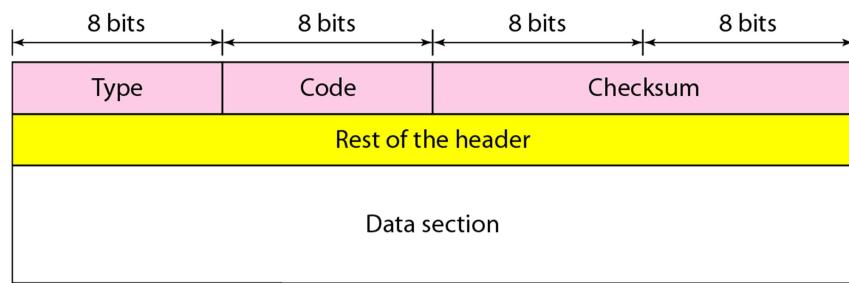
Types of Messages

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. For example, nodes can discover their neighbors. Also, hosts can discover and learn about routers on their network, and routers can help a node redirect its messages.

Message Format



An ICMP message has an 8-byte header and a variable-size data section. Although the general format of the header is different for each message type, the first 4 bytes are common to all. As above Figure shows, the first field, ICMP type, defines the type of the message. The code field specifies the reason for the particular message type. The last common field is the checksum field.

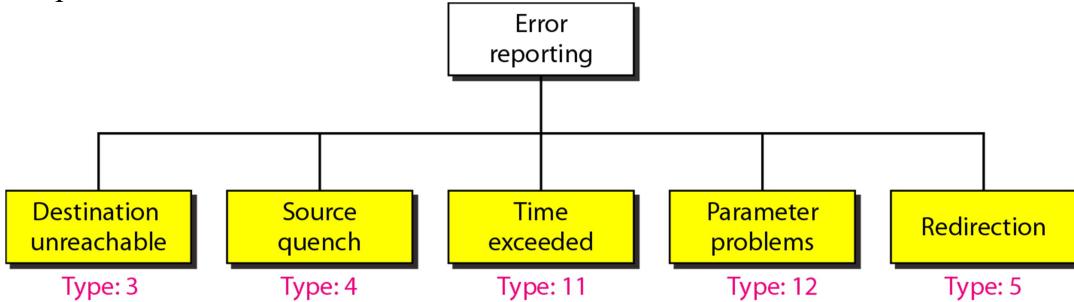
The rest of the header is specific for each message type.

The **data section** in error messages carries information for finding the original packet that had the error. In query messages, the data section carries extra information based on the type of the query.

Error reporting messages

One of the main responsibilities of ICMP is to report errors. Error messages are always sent to the original source because the only information available in the datagram about the route is the Source and destination IP addresses. ICMP uses the source IP address to send the error message to the source (originator) of the datagram.

Five types of errors are handled: destination unreachable, source quench, time exceeded, parameter problems, and redirection



Destination Unreachable

When a router cannot route a datagram or a host cannot deliver a datagram, the datagram is discarded and the router or the host sends a destination-unreachable message back to the source host that initiated the datagram. Note that destination-unreachable messages can be created by either a router or the destination host.

Source Quench

The lack of flow control can create congestion in routers or the destination host. A router or a host has a limited-size queue (buffer) for incoming datagrams waiting to be forwarded (in the case of a router) or to be processed (in the case of a host). If the datagrams are received much faster than they can be forwarded or processed, the queue may overflow. In this case, the router or the host has no choice but to discard some of the datagrams. The source-quench message in ICMP was designed to add a kind of flow control to the IP. When a router or host discards a datagram due to congestion, it sends a source-quench message to the sender of the datagram. This message has two purposes. First, it informs the source that the datagram has been discarded. Second, it warns the source that there is congestion somewhere in the path and that the source should slow down (quench) the sending process.

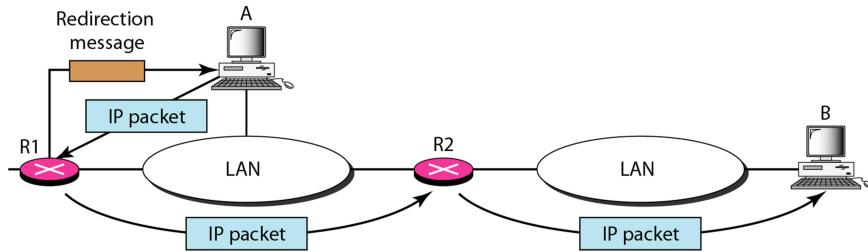
Time Exceeded

The time-exceeded message is generated in two cases: routers use routing tables to find the next hop (next router) that must receive the packet. If there are errors in one or more routing tables, a packet can travel in a loop or a cycle, going from one router to the next or visiting a series of routers endlessly, each datagram contains a field called time to live that controls this situation. When a datagram visits a router, the value of this field is decremented by 1. When the time-to-live value reaches 0, after decrementing, the router discards the datagram. However, when the datagram is discarded, a time-exceeded message must be sent by the router to the original source. Second, a time-exceeded message is also generated when not all fragments that make up a message arrive at the destination host within a certain time limit.

Parameter Problem

Any ambiguity in the header part of a datagram can create serious problems as the data gram travels through the Internet. If a router or the destination host discovers an ambiguous or missing value in any field of the datagram, it discards the datagram and sends a parameter-problem message back to the source.

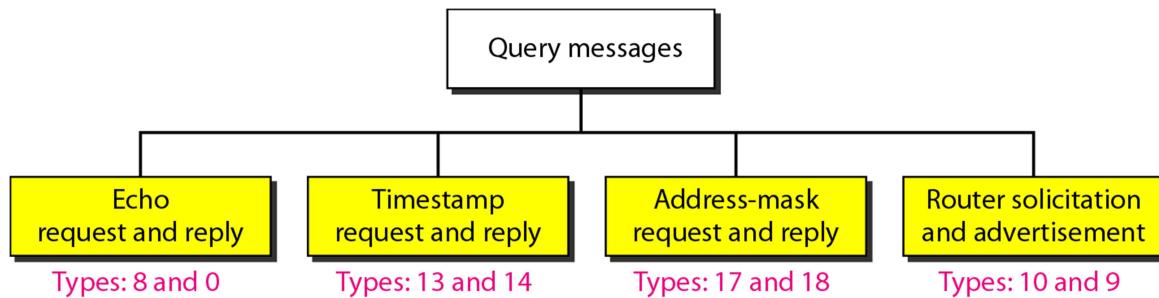
Redirection



When a router needs to send a packet destined for another network, it must know the IP address of the next appropriate router. The same is true if the sender is a host. Both routers and hosts, then, must have a routing table to find the address of the router or the next router. Routers take part in the routing update process and are supposed to be updated constantly. Routing is dynamic.

Query messages

In addition to error reporting, ICMP can diagnose some network problems. This is accomplished through the query messages, a group of four different pairs of messages, as shown in Figure below. In this type of ICMP message, a node sends a message that is answered in a specific format by the destination node. A query message is encapsulated in an IP packet, which in turn is encapsulated in a data link layer frame. However, in this case, no bytes of the original IP are included in the message, as shown in Figure



Echo Request and Reply

The echo-request and echo-reply messages are designed for diagnostic purposes. Network managers and users utilize this pair of messages to identify network problems. The combination of echo-request and echo-reply messages determines whether two systems (hosts or routers) can communicate with each other. The echo-request and echo-reply messages can be used to determine if there is communication at the IP level.

Timestamp Request and Reply

Two machines (hosts or routers) can use the timestamp request and timestamp reply messages to determine the round-trip time needed for an IP datagram to travel between them. It can also be used to synchronize the clocks in two machines.

Address-Mask Request and Reply

A host may know its IP address, but it may not know the corresponding mask. For example, a host may know its IP address as 159.31.17.24, but it may not know that the corresponding mask is /24. To obtain its mask, a host sends an address-mask-request message to a router on the LAN. If the host knows the address of the router, it sends the request directly to the router. If it does not know, it broadcasts the message. The router receiving the address-mask-request message

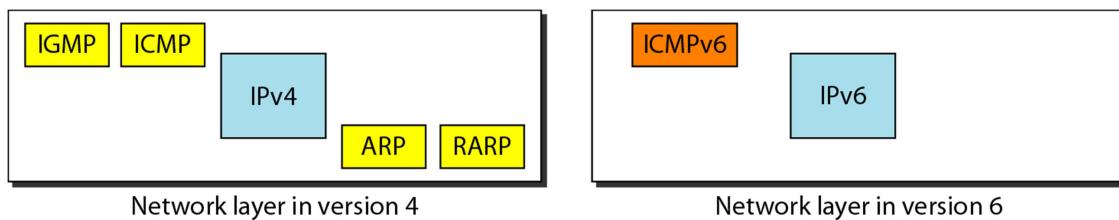
responds with an address-mask-reply message, providing the necessary mask for the host. This can be applied to its full IP address to get its subnet address.

Router Solicitation and Advertisement

As we discussed in the redirection message section, a host that wants to send data to a host on another network needs to know the address of routers connected to its own network. Also, the host must know if the routers are alive and functioning. The router-solicitation and router-advertisement messages can help in this situation. A host can broadcast (or multicast) a router-solicitation message. The router or routers that receive the solicitation message broadcast their routing information using the router-advertisement message. A router can also periodically send router-advertisement messages even if no host has solicited. Note that when a router sends out an advertisement, it announces not only its own presence but also the presence of all routers on the network of which it is aware.

ICMPv6

Another protocol that has been modified in version 6 of the TCPIIP protocol suite is ICMP (ICMPv6). This new version follows the same strategy and purposes of version 4. ICMPv4 has been modified to make it more suitable for IPv6. In addition, some protocols that were independent in version 4 are now part of Internetworking Control Message Protocol (ICMPv6). Figure below compares the network layer of version 4 to version 6



The ARP and IGMP protocols in version 4 are combined in ICMPv6. The RARP protocol is dropped from the suite because it was rarely used and BOOTP has the same functionality. Just as in ICMPv4, we divide the ICMP messages into two categories. However, each category has more types of messages than before.

Error Reporting

As we saw in version 4, one of the main responsibilities of ICMP is to report errors. Five types of errors are handled: destination unreachable, packet too big, time exceeded parameter problems, and redirection. ICMPv6 forms an error packet, which is then encapsulated in an IP datagram. This is delivered to the original source of the failed datagram. Table below compares the error-reporting messages of ICMPv4 with ICMPv6. The source-quench message is eliminated in version 6 because the priority and the flow label fields allow the router to control congestion and discard the least important messages. In this version, there is no need to inform the sender to slow down. The packet-too-big message is added because fragmentation is the responsibility of the sender in IPv6. If the sender does not make the right packet size decision, the router has no choice but to drop the packet and send an error message to the sender

Comparison of error-reporting messages in ICMPv4 and ICMPv6

Type of Message	Version 4	Version 6
Destination unreachable	Yes	Yes
Source quench	Yes	No
Packet too big	No	Yes
Time exceeded	Yes	Yes
Parameter problem	Yes	Yes
Redirection	Yes	Yes

Query

In addition to error reporting, ICMP can diagnose some network problems. This is accomplished through the query messages. Four different groups of messages have been defined: echo request and reply, router solicitation and advertisement, neighbor solicitation and advertisement, and group membership. Table 21.4 shows a comparison between the query messages in versions 4 and 6.

Comparison of query messages in ICMPv4 and ICMPv6

Type of Message	Version 4	Version 6
Echo request and reply	Yes	Yes
Timestamp request and reply	Yes	No
Address-mask request and reply	Yes	No
Router solicitation and advertisement	Yes	Yes
Neighbor solicitation and advertisement	ARP	Yes
Group membership	IGMP	Yes