UNIT III - THE NETWORK LAYER

The network layer is concerned with getting packets from the source all the way to the destination.

3.1. NETWORK LAYER DESIGN ISSUES

Store-and-Forward Packet Switching

The major components of the system are the carrier's equipment (routers connected by transmission lines), shown inside the shaded oval, and the customers' equipment, shown outside the oval. Host H1 is directly connected to one of the carrier's routers, A, by a leased line. In contrast, H2 is on a LAN with a router, F, owned and operated by the customer. This router also has a leased line to the carrier's equipment. We have shown F as being outside the oval because it does not belong to the carrier, but in terms of construction, software, and protocols, it is probably no different from the carrier's routers.

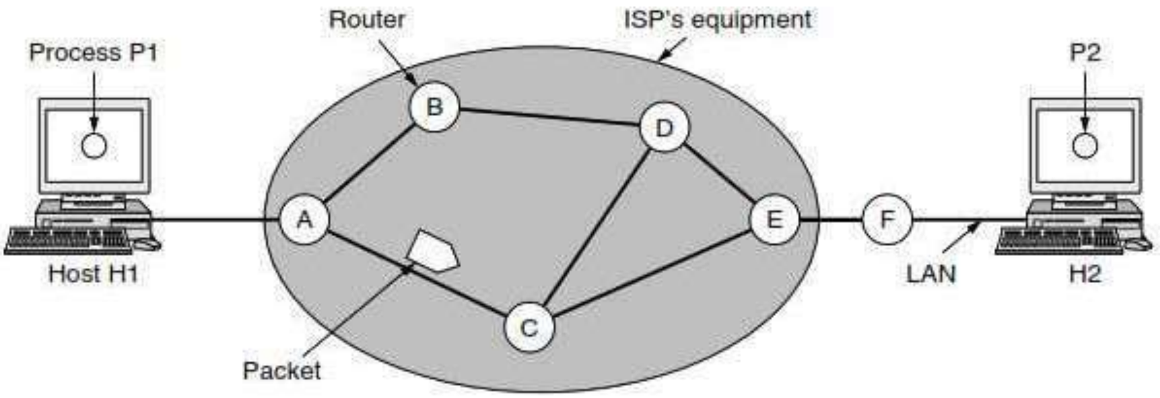


Figure 5-1. The environment of the network layer protocols.

This equipment is used as follows. A host with a packet to send transmits it to the nearest router, either on its own LAN or over a point-to-point link to the carrier. The packet is stored there until it has fully arrived so the checksum can be verified. Then it is forwarded to the next router along the path until it reaches the destination host, where it is delivered. This mechanism is store-and-forward packet switching

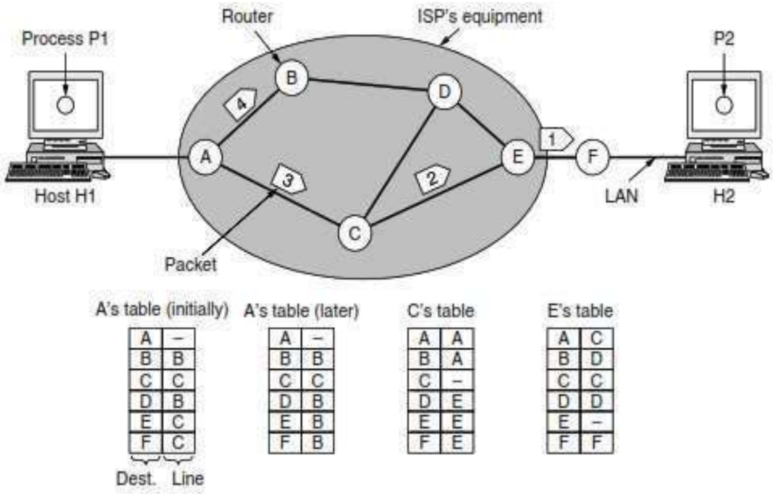
Services Provided to the Transport Layer

The network layer provides services to the transport layer at the network layer/transport layer interface. An important question is what kind of services the network layer provides to the transport layer. The network layer services have been designed with the following goals in mind.

- The services should be independent of the router technology.
- The transport layer should be shielded from the number, type, and topology of the routers present.
- The network addresses made available to the transport layer should use a uniform numbering plan, even across LANs and WANs.

Implementation of Connectionless Service

If connectionless service is offered, packets are injected into the subnet individually and routed independently of each other. No advance setup is needed. In this context, the packets are frequently called datagrams (in analogy with telegrams) and the subnet is called a datagram subnet. If connection oriented service is used, a path from the source router to the destination router must be established before any data packets can be sent. This connection is called a VC (virtual circuit), in analogy with the physical circuits set up by the telephone system, and the subnet is called a virtual-circuit subnet. In this section we will examine datagram subnets; in the next one we will examine virtualcircuit subnets.



A's table (initially)

A	–
B	B
C	C
D	B
E	C
F	C

A's table (later)

A	–
B	B
C	C
D	B
E	B
F	B

C's table

A	A
B	A
C	–
D	E
E	E
F	E

E's table

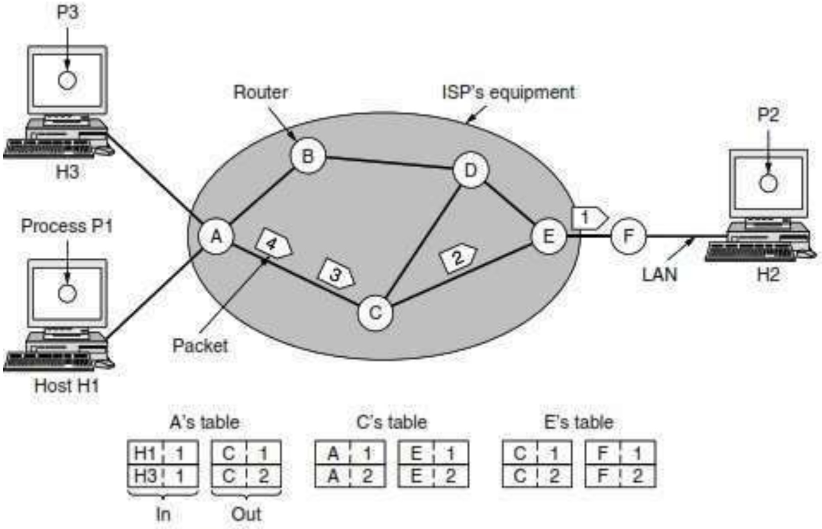
A	C
B	D
C	C
D	D
E	–
F	F

Figure 5-2. Routing within a datagram network.

A has only two outgoing lines—to B and C—so every incoming packet must be sent to one of these routers, even if the ultimate destination is some other router. A's initial routing table is shown in the figure under the label "initially." As they arrived at A, packets 1, 2, and 3 were stored briefly (to verify their checksums). Then each was forwarded to C according to A's table. Packet 1 was then forwarded to E and then to F. When it got to F, it was encapsulated in a data link layer frame and sent to H2 over the LAN. Packets 2 and 3 follow the same route. However, something different happened to packet 4. When it got to Ait was sent to router B, even though it is also destined for F. For some reason, A decided to send packet 4 via a different route than that of the first three. Perhaps it learned of a traffic jam somewhere along the ACE path and updated its routing table, as shown under the label "later." The algorithm that manages the tables and makes the routing decisions is called the routing algorithm.

Implementation of Connection-Oriented Service

For connection-oriented service, we need a virtual-circuit network. Let us see how that works. The idea behind virtual circuits is to avoid having to choose a new route for every packet sent, as in Fig. 5-2. Instead, when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated. With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.



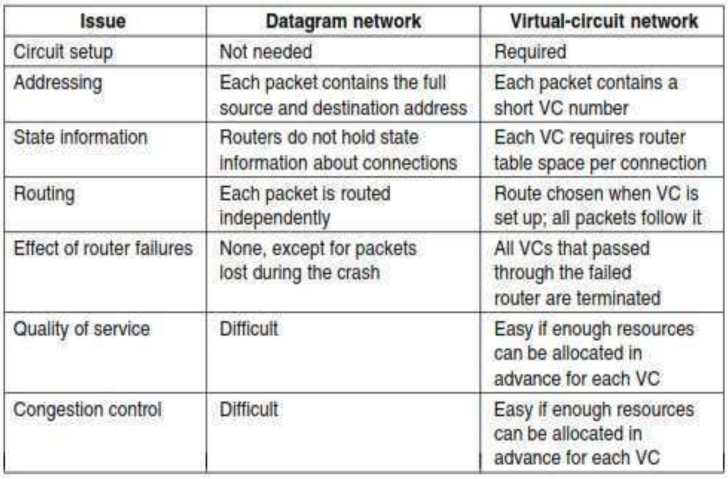
A's table In	A's table Out	C's table In	C's table Out	E's table In	E's table Out
H1 1	C 1	A 1	E 1	C 1	F 1
H3 1	C 2	A 2	E 2	C 2	F 2

Figure 5-3. Routing within a virtual-circuit network.

Host H1 has established connection 1 with host H2. It is remembered as the first entry in each of the routing tables. The first line of A's table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1. Similarly, the first entry at C routes the packet to E, also with connection identifier 1. Now let us consider what happens if H3 also wants to establish a connection to H2. It chooses connection identifier and tells the subnet to establish the virtual circuit.

Comparison of Virtual-Circuit and Datagram Networks

Both virtual circuits and datagrams have their supporters and their detractors. We will now attempt to summarize both sets of arguments.



Issue	Datagram network	Virtual-circuit network
Circuit setup	Not needed	Required
Addressing	Each packet contains the full source and destination address	Each packet contains a short VC number
State information	Routers do not hold state information about connections	Each VC requires router table space per connection
Routing	Each packet is routed independently	Route chosen when VC is set up; all packets follow it
Effect of router failures	None, except for packets lost during the crash	All VCs that passed through the failed router are terminated
Quality of service	Difficult	Easy if enough resources can be allocated in advance for each VC
Congestion control	Difficult	Easy if enough resources can be allocated in advance for each VC

Figure 5-4. Comparison of datagram and virtual-circuit networks.

3.1.2 NETWORK LAYER PERFORMANCE

- The performance of a network can be measured in terms of

  Delay, Throughput and Packet loss.

- Congestion control is an issue that can improve the performance.

DELAY

- A packet from its source to its destination, encounters delays.
- The delays in a network can be divided into four types:

Transmission delay, Propagation delay, Processing delay and Queuing delay.

Transmission Delay

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

  Delaytr = (Packet length) / (Transmission rate)

- Example :

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

Propagation Delay

- Propagation delay is the time it takes for a bit to travel from point A to point B in the transmission media.
- The propagation delay for a packet-switched network depends on the propagation delay of each network (LAN or WAN).
- The propagation delay depends on the propagation speed of the media, which is

$3X10^8$ meters/second in a vacuum and normally much less in a wired medium.

- It also depends on the distance of the link.

- The Propagation delay is calculated using the formula

Delaypg = (Distance) / (Propagation speed)

- Example

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

Processing Delay

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

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

Delaypr = Time required to process a packet in a router or a destination host

Queuing Delay

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

Delayqu = The time a packet waits in input and output queues in a router

Total Delay

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

Total delay = (n + 1) (Delaytr + Delaypg + Delaypr) + (n) (Delayqu)

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

THROUGHPUT

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

Throughput = minimum{TR1 , TR2, . . . TRn}

- Example:

Let us assume that we have three links, each with a different transmission rate. The data can flow at the rate of 200 kbps in Link1, 100 kbps in Link2 and 150kbps in Link3.

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

PACKET LOSS

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

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

3.2. ROUTING ALGORITHMS

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. If the subnet uses datagrams internally, this decision must be made anew for every arriving data packet since the best route may have changed since last time.

If the subnet uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up. Thereafter, data packets just follow the previously established route. The latter case is sometimes called session routing

Routing algorithms can be grouped into two major classes:

Non Adaptive

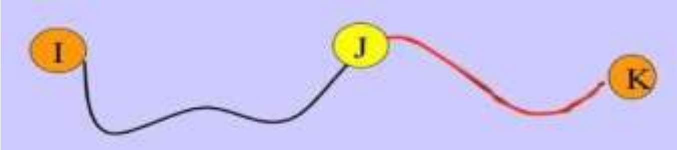
- Adaptive.

Non Adaptive algorithms do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J (for all I and J) is computed in advance, off- line, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing.

Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well. Adaptive algorithms differ in where they get their information (e.g., locally, from adjacent routers, or from all routers), when they change the routes

The Optimality Principle

It states that if the router J is on the optimal path from router I to router K, then the optimal path from J to K also falls along the same route.



Shortest Path Algorithm

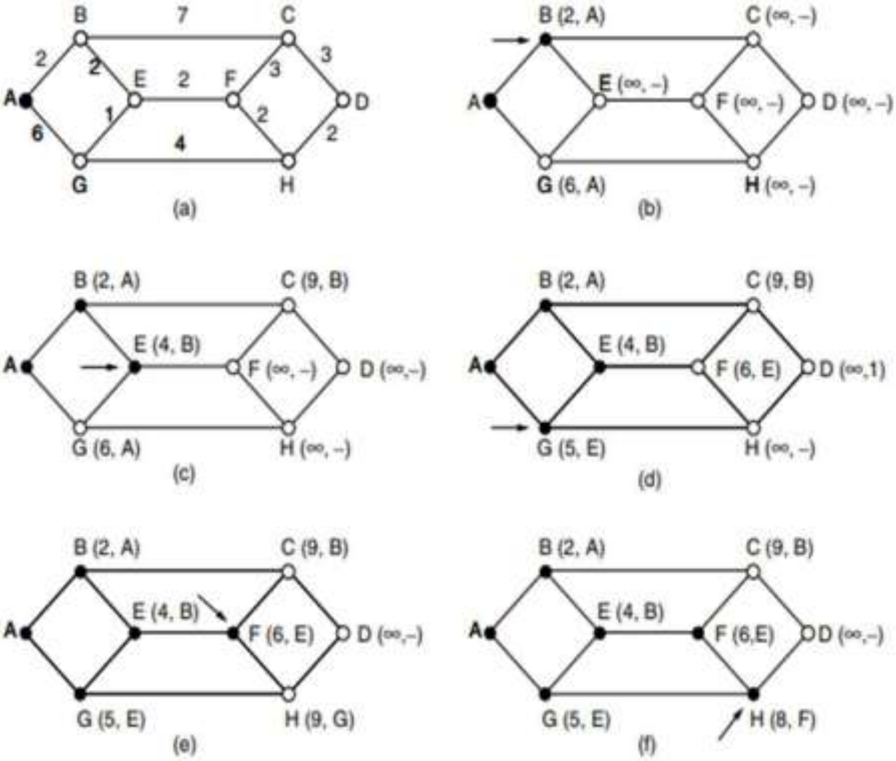


Figure 5-7. The first six steps used in computing the shortest path from *A* to *D*. The arrows indicate the working node.

- Let us begin our study of routing algorithms with a simple technique for computing optimal paths given a complete picture of the network.
- To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.
- In the general case, the labels on the edges could be computed as a function of the distance, bandwidth, average traffic, communication cost, measured delay, and other factors. By changing the weighting function, the algorithm would then compute the ''shortest'' path measured according to any one of a number of criteria or to a combination of criteria.

- Several algorithms for computing the shortest path between two nodes of a graph are known. This one is due to Dijkstra (1959) and finds the shortest paths between a source and all destinations in the network. Each node is labeled (in parentheses) with its distance from the source node along the best known path.
- We want to find the shortest path from A to D. We start out by marking node A as permanent, indicated by a filled-in circle. Then we examine, in turn, each of the nodes adjacent to A (the working node), relabeling each one with the distance to A. If the network had more than one shortest path from A to D and we wanted to find all of them, we would need to remember all of the probe nodes that could reach a node with the same distance.
- Having examined each of the nodes adjacent to A, we examine all the tentatively labeled nodes in the whole graph and make the one with the smallest label permanent, as shown in Fig. 5-7(b). This one becomes the new working node. We now start at B and examine all nodes adjacent to it. If the sum of the label on B and the distance from B to the node being considered is less than the label on that node, we have a shorter path, so the node is relabeled.
- After all the nodes adjacent to the working node have been inspected and the tentative labels changed if possible, the entire graph is searched for the tentatively labeled node with the smallest value. This node is made permanent and becomes the working node for the next round. Figure 5-7 shows the first six steps of the algorithm.

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. One such measure is to have a hop counter contained in the header of each packet that is decremented at each hop, with the packet being discarded when the counter reaches zero.

A better technique for damming the flood is to have routers keep track of which packets have been flooded, to avoid sending them out a second time. One way to achieve this goal is to have the source router put a sequence number in each packet it receives from its hosts. Each router then needs a list per source router telling which sequence numbers originating at that source have already been seen. If an incoming packet is on the list, it is not flooded.

Distance Vector Routing

Computer networks generally use dynamic routing algorithms that are more complex than flooding, but more efficient because they find shortest paths for the current topology. Two dynamic algorithms in particular, distance vector routing and link state routing, are the most popular. In this section, we will look at the former algorithm.

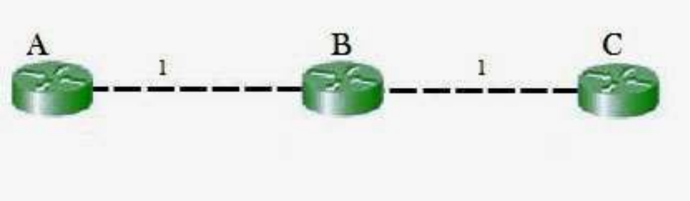
A distance vector routing algorithm operates by having each router maintain a table (i.e., a vector) giving the best known distance to each destination and which link to use to get there. These tables are updated by exchanging information with the neighbors. Eventually, every router knows the best link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

The distance vector routing algorithm is sometimes called by other names, most commonly the distributed BellmanFord routing algorithm, after the researchers who developed it (Bellman, 1957; and Ford and Fulkerson, 1962). It was the original ARPANET routing algorithm and was also used in the Internet under the name RIP.

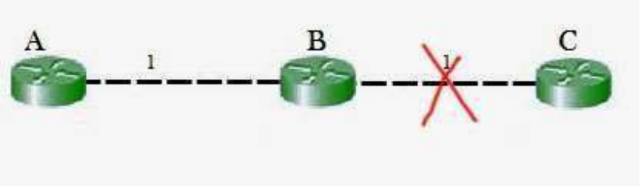
In distance vector routing, each router maintains a routing table indexed by, and containing one entry for each router in the network. This entry has two parts: the preferred outgoing line to use for that destination and an estimate of the distance to that destination.

The Count-to-Infinity Problem

The main issue with Distance Vector Routing (DVR) protocols is Routing Loops since Bellman-Ford Algorithm cannot prevent loops. This routing loop in the DVR network causes the Count to Infinity Problem. Routing loops usually occur when an interface goes down or two routers send updates at the same time.



So in this example, the Bellman-Ford algorithm will converge for each router, they will have entries for each other. B will know that it can get to C at a cost of 1, and A will know that it can get to C via B at a cost of 2.



If the link between B and C is disconnected, then B will know that it can no longer get to C via that link and will remove it from its table. Before it can send any updates it’s possible that it will receive an update from A which will be advertising that it can get to C at a cost of 2. B can get to A at a cost of 1, so it will update a route to C via A at a cost of 3. A will then receive updates from B later and update its cost to 4. They will then go on feeding each other bad information toward infinity which is called as Count to Infinity problem.

3.2.1 UNICAST ROUTING ALGORITHMS

- There are three main classes of routing protocols:

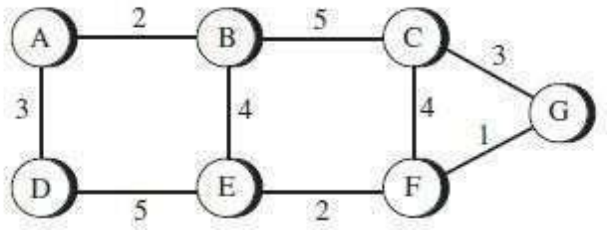
1) Distance Vector Routing Algorithm – Routing Information Protocol 2) Link State Routing Algorithm – Open Shortest Path First Protocol 3) Path-Vector Routing Algorithm - Border Gateway Protocol

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

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

NETWORK AS A GRAPH

- The Figure below shows a graph representing a network.



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

UNICAST ROUTING ALGORITHMS

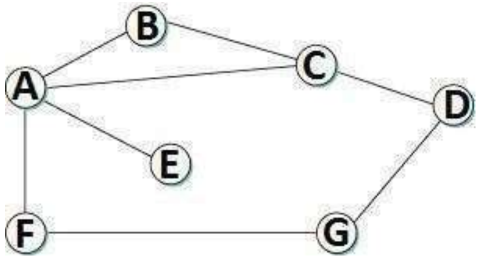
- There are three main classes of routing protocols:

4) Distance Vector Routing Algorithm – Routing Information Protocol
5) Link State Routing Algorithm – Open Shortest Path First Protocol 6) Path-Vector Routing Algorithm - Border Gateway Protocol

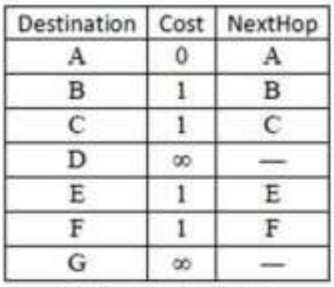
DISTANCE VECTOR ROUTING (DSR)
ROUTING INFORMATION PROTOCOL (RIP)
BELLMAN - FORD ALGORITHM

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

Initial State

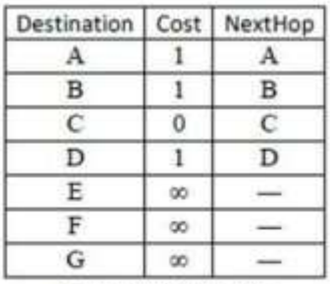


- In given network, cost of each link is 1 hop.
- Each node sets a distance of 1 (hop) to its immediate neighbor and cost to itself as 0.
- Distance for non-neighbors is marked as unreachable with value ∞ (infinity).
- For node A, nodes B, C, E and F are reachable, whereas nodes D and G are unreachable.



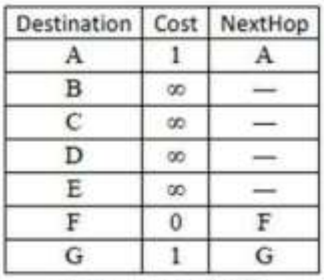
Destination	Cost	NextHop
A	0	A
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

Node A's initial table



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

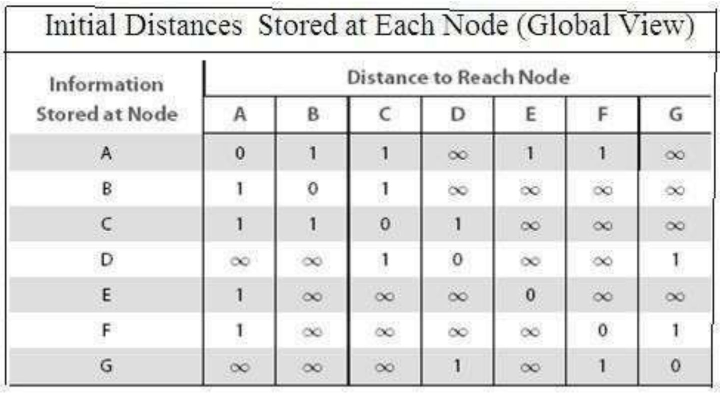
Node C's initial table



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

Node F's initial table

 The initial table for all the nodes are given below



Initial Distances Stored at Each Node (Global View)

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

 Each node sends its initial table (distance vector) to neighbors and receives their estimate.

 Node A sends its table to nodes B, C, E & F and receives tables from nodes B, C, E & F. Each node updates its routing table by comparing with each of its neighbor's table For each destination, Total Cost is computed as:

 Total Cost = Cost (Node to Neighbor) + Cost (Neighbor to Destination) If Total Cost < Cost then

 Cost = Total Cost and NextHop = Neighbor

 Node A learns from C's table to reach node D and from F's table to reach node G. Total Cost to reach node D via C = Cost (A to C) + Cost(C to D)

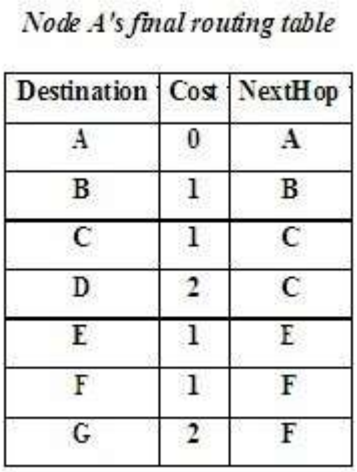
Cost = 1 + 1 = 2.

 Since 2 < ∞, entry for destination D in A's table is changed to (D, 2, C)

 Total Cost to reach node G via F = Cost(A to F) + Cost(F to G) = 1 + 1 = 2

 Since 2 < ∞, entry for destination G in A's table is changed to (G, 2, F)

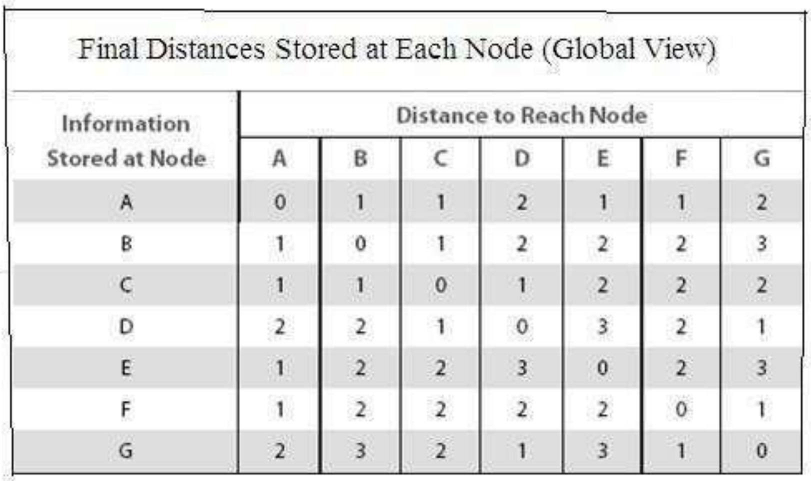
 Each node builds complete routing table after few exchanges amongst its neighbors.



Node A's final routing table

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

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



Final Distances Stored at Each Node (Global View)

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

Updation of Routing Tables

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

Periodic Update

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

Triggered Update

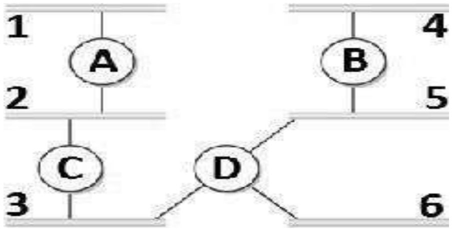
In this case, whenever a node notices a link failure or receives an update from one of its neighbors that causes it to change one of the routes in its routing table.

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

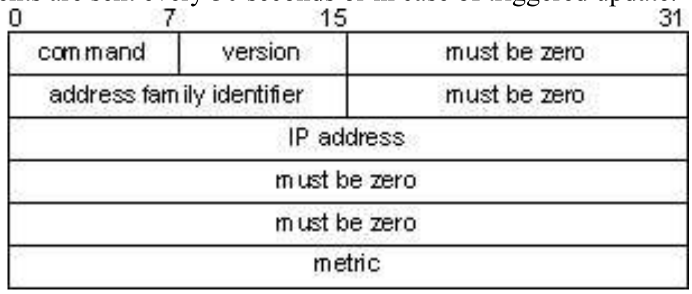
ROUTING INFORMATION PROTOCOL (RIP)

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

Example



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



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

□ Command - It indicates the packet type.

Value 1 represents a request packet. Value 2 represents a response packet.

□ Version - It indicates the RIP version number. For RIPv1, the value is 0x01.

□ Address Family Identifier - When the value is 2, it represents the IP protocol.

□ IP Address - It indicates the destination IP address of the route. It can be the addresses of only the natural network segment.

□ Metric - It indicates the hop count of a route to its destination.

- Each node knows state of link to its neighbors and cost.
- Nodes create an update packet called link-state packet (LSP) that contains:

□ ID of the node

□ List of neighbors for that node and associated cost

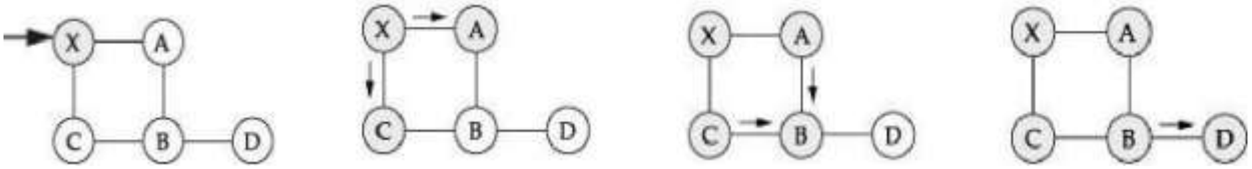
□ 64-bit Sequence number

□ Time to live

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

Reliable Flooding

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



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

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

Route Calculation

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

DIJKSTRA'S SHORTEST PATH ALGORITHM (FORWARD SEARCH ALGORITHM)

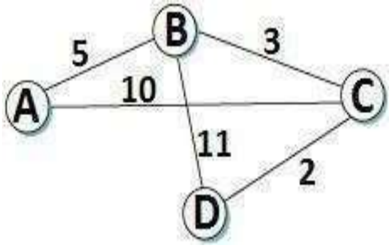
1. Each host maintains two lists, known as Tentative and Confirmed 2.

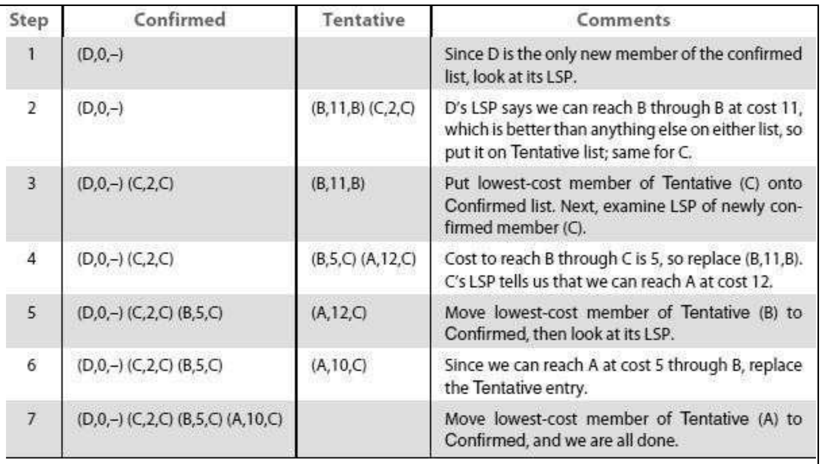
   Initialize the Confirmed list with an entry for the Node (Cost = 0).

3. Node just added to Confirmed list is called Next. Its LSP is examined.
4. For each neighbor of Next, calculate cost to reach each neighbor as Cost (Node to Next) + Cost (Next to Neighbor).
   - a. If Neighbor is neither in Confirmed nor in Tentative list, then add (Neighbor, Cost, NextHop) to Tentative list.
   - b. If Neighbor is in Tentative list, and Cost is less than existing cost, then replace the entry with (Neighbor, Cost, NextHop).

5. If Tentative list is empty then Stop, otherwise move least cost entry from Tentative list to Confirmed list. Go to Step 2.

Example :



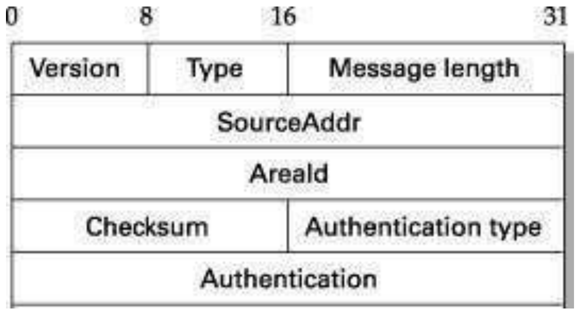


Step	Confirmed	Tentative	Comments
1	(D,0,–)		Since D is the only new member of the confirmed list, look at its LSP.
2	(D,0,–)	(B,11,B) (C,2,C)	D's LSP says we can reach B through B at cost 11, which is better than anything else on either list, so put it on Tentative list; same for C.
3	(D,0,–) (C,2,C)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

OPEN SHORTEST PATH FIRST PROTOCOL (OSPF)

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

Link State Packet Format



□ Version — represents the current version, i.e., 2.

□ Type — represents the type (1–5) of OSPF message.

Type 1 - “hello” message, Type 2 - request, Type 3 – send , Type 4 - acknowledge the receipt of link state messages , Type 5 - reserved

- □ SourceAddr — identifies the sender
- □ AreaId — 32-bit identifier of the area in which the node is located
- □ Checksum — 16-bit internet checksum
- □ Authentication type — 1 (simple password), 2 (cryptographic authentication).
- □ Authentication — contains password or cryptographic checksum

Difference Between Distance-Vector And Link-State Algorithms

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

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

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

Spanning Trees

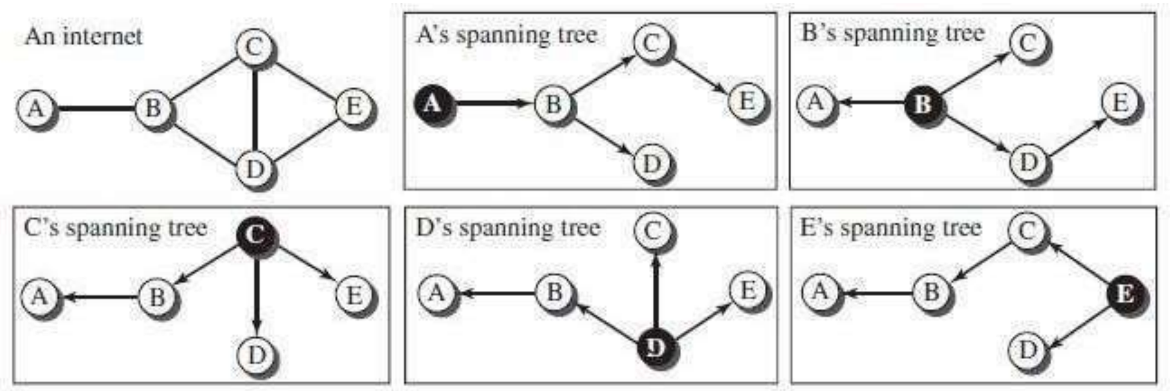
- In path-vector routing, the path from a source to all destinations is determined by the best spanning tree.
- The best spanning tree is not the least-cost tree.
- It is the tree determined by the source when it imposes its own policy.
- If there is more than one route to a destination, the source can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

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

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

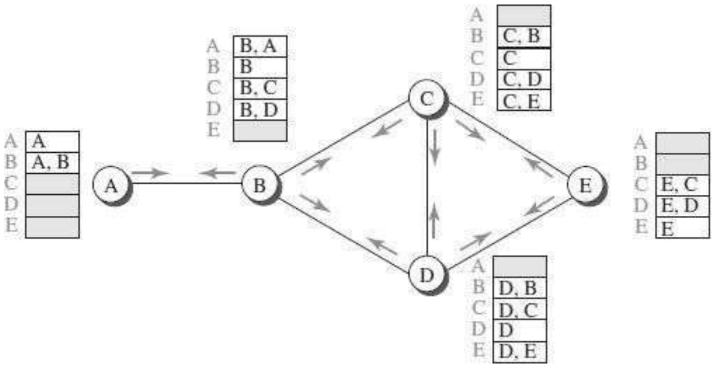
Example:

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



Path Vectors made at booting time

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



A	A
B	A, B
C	
D	
E	

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

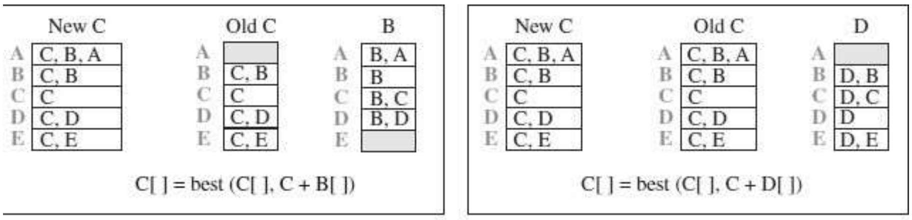
A	
B	C, B
C	C
D	C, D
E	C, E

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

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

Updating Path Vectors

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



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

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

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

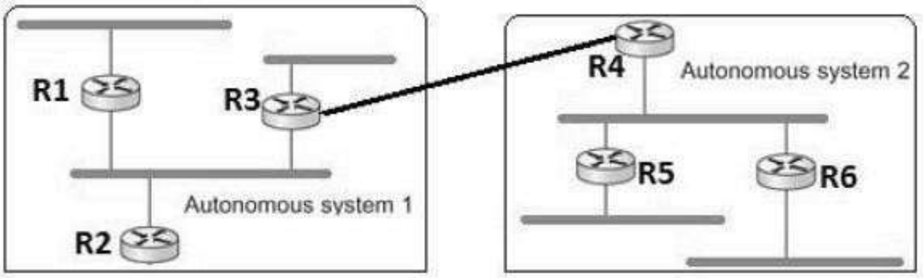
	New C	Old C	D
A	C, B, A	C, B, A	
B	C, B	C, B	D, B
C	C	C	D, C
D	C, D	C, D	D
E	C, E	C, E	D, E

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

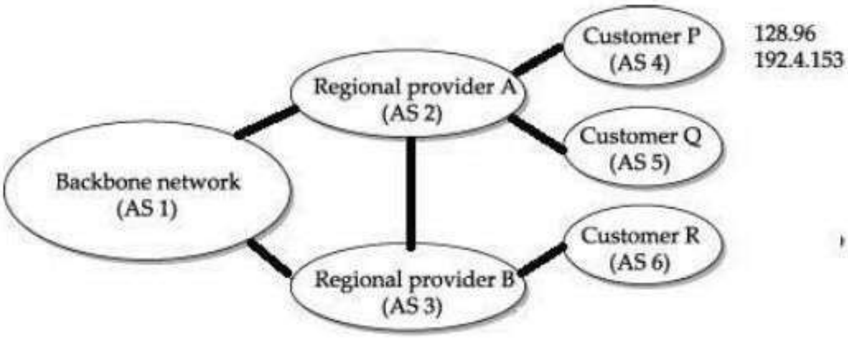
Event 2: C receives a copy of D's vector

BORDER GATEWAY PROTOCOL (BGP)

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

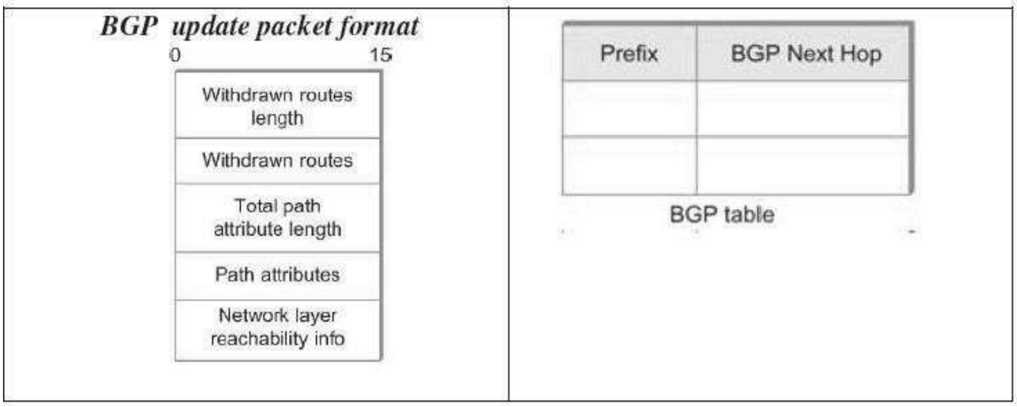


- Each AS have a border router (gateway), by which packets enter and leave that AS. In above figure, R3 and R4 are border routers.
- One of the router in each autonomous system is designated as BGP speaker.
- BGP Speaker exchange reachability information with other BGP speakers, known as external BGP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



- If there are multiple routes to a destination, BGP speaker chooses one based on policy.

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



iBGP - interior BGP

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

3.3. CONGESTION CONTROL ALGORITHM

When too many packets are present in the network it causes packet delay and loss of packet which degrades the performance of the system. This situation is called congestion.

The network layer and transport layer share the responsibility for handling congestions. One of the most effective ways to control congestion is trying to reduce the load that transport layer is placing on the network.

Congestion Control refers to techniques and mechanisms that can either prevent congestion, before it happens, or remove congestion, after it has happened.

In a network vulnerable to congestion, the more the total amount of sent packets reaches the maximum capacity, the more the total amount of delivered packets decreases. This can result in a situation where nearly no packets get delivered successfully. The goal of congestion control is to keep network capacity at its maximum level. The following factors responsible for congestion:

- Slow network links
- Shortage of buffer space
- Slow processors

Congestion Control in computer networks covers several mechanisms getting the performance - capacity desirable one and thus prevent the network from congestion. The generic term Congestion Control covers preventive as well as reactive mechanisms.

Congestion Control can be realized by working on either side of the equation, thus by decreasing demand or by increasing available resources. To control congestion in computer networks, it is important to take a look on the architecture and implementation of the network, because any design decision affects the congestion control strategy of the network and thus the demand or the available resources in the previous equation. Those design decisions are called Policies.

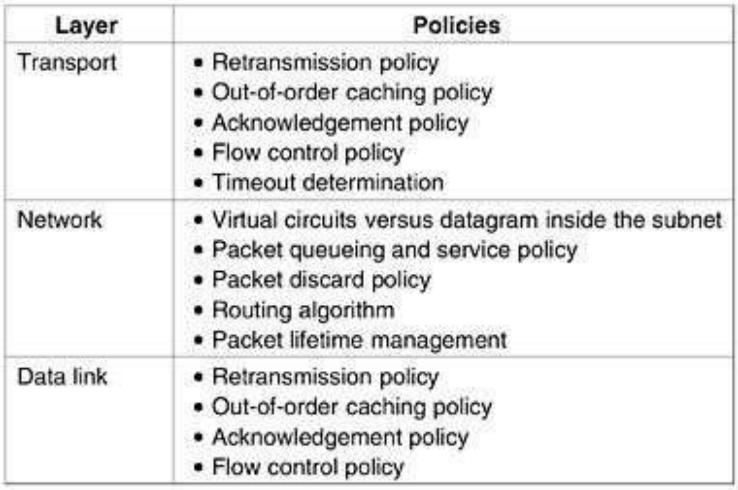
One important policy is the connection mechanism. There are two fundamental distinct network types, connection-oriented and connectionless networks. In connection-oriented network, all communication of a connection between two endpoints is routed over the same gateways. In connectionless networks, single packets are routed independently. Therefore, packets belonging to a single connection can be routed differently.

Admission Control as described in below is part of the connection-oriented networks concept, but it is conceptually not a part of connectionless networks. However, state of the art network architectures combines the robustness of the cheaper connectionless networks with the service quality advantages of connection-oriented networks. The Quality of Service (QoS) research area deals with this topic. Congestion Control and resource allocation are two sides of the same coin.

In a congested network,

where Total Demand > Available Resources is true.

It is often desirable to allocate the rare resources fair, so that every participant gets an equal share of the networks capacity. Packet queuing and service policies affect this subject. We will introduce Fair Queuing below as a mechanism to ensure fairness for every source. Under the side effect of greater overhead, another concept ensures fairness for every source-destination pair. Packet drop policies are related to the issue of dropping packets when buffer space is to short to queue more incoming packets. Examples for this are Load Shedding and Active Queue Management (AQM).



Layer	Policies
Transport	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	• Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	• Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Policies that affects congestion control

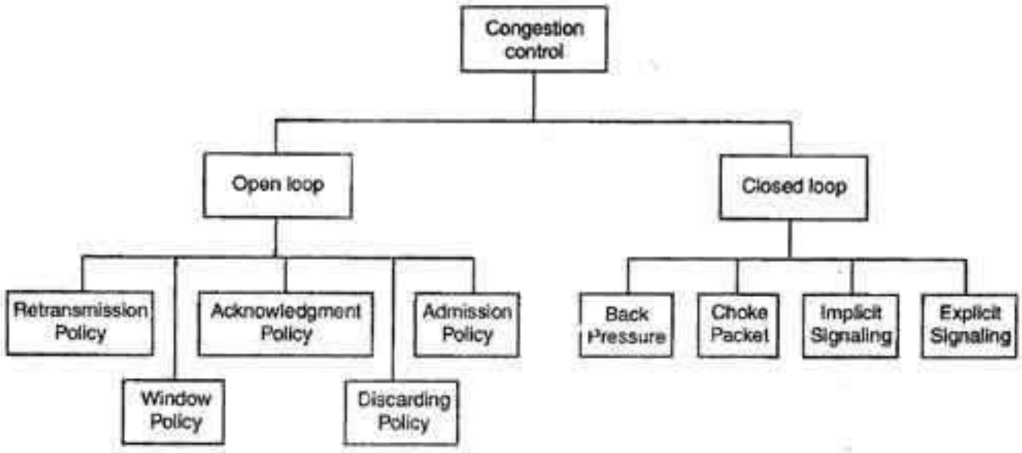
Table summarizes policies that affect congestion control mechanisms on different network layers. In this topic we only consider policies from the network and the transport layer.

Congestion can be controlled by increasing available resources. This becomes necessary when network resources are congested over a long period of time. Stronger resources have to be build up, but that is a manual task and not subject to Congestion Control algorithms implemented in network software or hardware.

Three examples to increase available resources:

1. Dial-up links can be added during high usage.
2. Power increases on satellite links to increase their bandwidth.
3. Network paths can be split, so that extra traffic is sent via routes that may not be considered as optimal under low load. This solution is not yet researched well and therefore is not used in practice today. This approach breaks with the concept of connection-oriented networks, because data is no longer routed over the same network path.

Here focus is on Congestion Control algorithms that decrease demand or service can be classified in various ways. Congestion control mechanisms are divided into two categories, one category prevents the congestion from happening and the other category removes congestion after it has taken place.



Types of Congestion Control Methods

These two categories are:

1. Open loop
2. Closed loop

Open Loop Congestion Control

- In this method, policies are used to prevent the congestion before it happens.
- Congestion control is handled either by the source or by the destination.
- 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.
- However, retransmission in general may increase the congestion in the network. But we need to implement good retransmission policy to prevent congestion.
- 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 over Go-back-n window as in Go-back-n method, when timer for a packet times out, several packets are resent, although some may have arrived safely at the receiver. Thus, this duplication may make congestion worse.
- Selective reject method 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.

- Acknowledgments also add to the traffic load on the network. Thus, by sending fewer acknowledgements we can reduce load on the network.
- To implement it, several approaches can be used:

  1. A receiver may send an acknowledgement only if it has a packet to be sent.
  2. A receiver may send an acknowledgement when a timer expires.
  3. A receiver may also decide to acknowledge only N packets at a time.

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

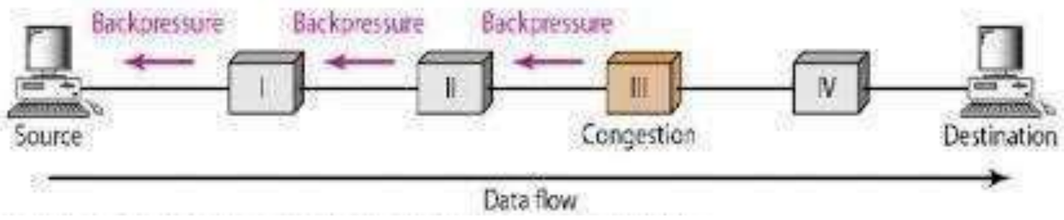
- An admission policy, which is a quality-of-service mechanism, can also prevent congestion in virtual circuit networks.
- Switches in a flow first check the resource requirement of a flow before admitting it to the network.
- A router can deny establishing a virtual circuit connection if there is congestion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

Closed Loop Congestion Control

- Closed loop congestion control mechanisms try to remove the congestion after it happens.
- The various methods used for closed loop congestion control are:

Backpressure

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

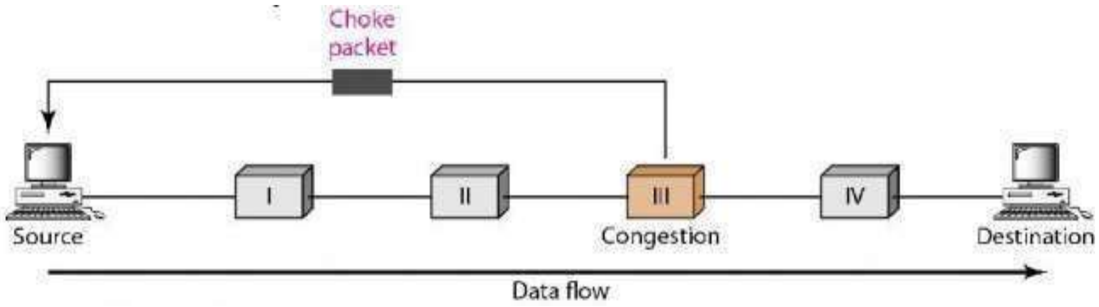


- The backpressure technique can be applied only to virtual circuit networks. In such virtual circuit each node knows the upstream node from which a data flow is coming.
- In this method of congestion control, the congested node stops receiving data from the immediate upstream node or nodes.
- This may cause the upstream node on 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 turns 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 i.e. the intermediate nodes through which the packet has traveled are not warned.



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. The receiver in this case uses policies such as slowing down the acknowledgements to remove the congestion.

Traffic Shaping

All network nodes like hosts or gateways, can be sending agents, because they are able to send data to other network nodes. In networks based on virtual circuits, the transmission rates and quality are negotiated with receiving network nodes when a connection is initialized. A flow specification is the result of this negotiation. Sending agents have to adhere to the flow specification and shape their traffic accordingly. In packet switched networks the volume of traffic is changing quickly over time. This makes periodic renegotiations necessary in those networks. Quality of service negotiation is called Traffic Policing. To control the sending rate of agents and prevent the receiving agents from data overflow, one of the most established mechanisms on the network layer is the Leaky Bucket algorithm.

Leaky Bucket Algorithm

The Leaky Bucket algorithm is a Traffic Shaping solution, categorized by as an open loop approach. Just imagine a leaky bucket that has a hole on its ground. As you can see in figure 6.6 given below.

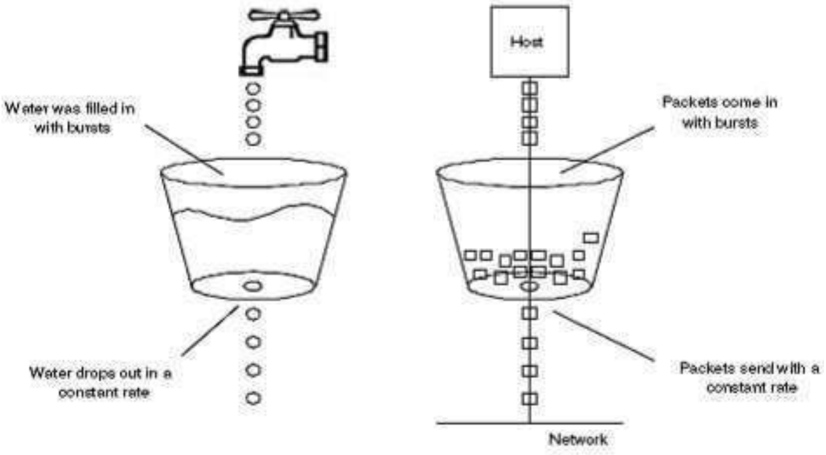

Water was filled in with bursts
Water drops out in a constant rate
Host
Packets come in with bursts
Packets send with a constant rate
Network

Figure 6.6: Leaky Bucket Algorithm

If the bucket is empty, no water drops out. If there is some water inside, water drops out with a constant rate. If the bucket is full and additional water is filled in, the water overflows the bucket. This can be implemented as follows. On a sender, a time-dependent queue ensures that an equal amount of data units is send per time interval. On the one hand, data can be put fast into the queue until it is full. On the other hand, data always leave the queue at the same rate. If the queue is full, no more packets can be stored in it and packet loss occurs. If the queue is empty, no data leaves the queue. The Leaky Bucket algorithm can be implemented for packets or a constant amount of bytes, send within each time interval. Using this algorithm, transmission peaks are flattened to a constant transmission rate on a sending agent.

Token Bucket Algorithm

A more flexible approach to control the sending rate on agents is the Token Bucket algorithm, also categorized as an open-loop approach. This algorithm allows bursts for short transmission while making sure that no data is lost. In contrast to the Leaky Bucket algorithm, not the data that is to be send but tokens are queued in a time-depended queue. One token is needed to send a single portion of data.

Implementations contain a token counter that is incremented on every time interval, so that the counter grows over time up until a maximum counter value is reached. The token counter is decremented by one for every data portion sent. When the token counter is zero, no data can be transmitted. An example would be a token counter with a maximum of 50 tokens. When no data is sent for a longer period of time, the counter will reach and stay at its maximum. In this situation, when an application sends a series of 100 data portions through the network interface, 50 portions can be send immediately because there are 50 tokens available. After that, the remaining 50 portions are stored and transmitted one by one on each time interval the token counter gets incremented.

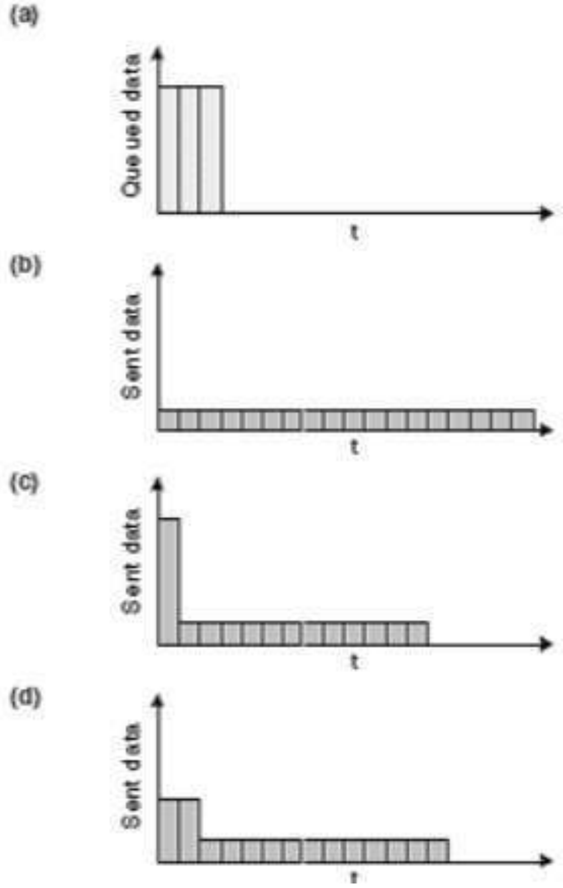


Figure 6.7: Traffic Shaping with Leaky and Token Bucket Algorithms

Token Bucket's characteristic influence on the traffic shape is displayed on diagram (c) in figure 6.7, where the sample incoming burst displayed on diagram (a) is processed. First, there are enough buckets available to allow a burst over a while. After that, the traffic is sent with a constant lower rate, because new tokens get available at a constant rate of time.

Token Bucket as well as Leaky Bucket algorithms can be implemented for packets or constant byte amounts. In the case of Token Bucket with a constant amount of bytes for each token, it is possible to store fractions of tokens for later transmissions, when they have not be fully consumed by sending packets. For example when a 100 byte token is used to send a 50 byte packet, it is possible to store the rest of the token (50 bytes) for later transmissions.

Load Shedding

An open-loop approach to control congestion on routers is the Load Shedding mechanism. Load Shedding is a way to instruct a router to drop packets from its queue, if it is congested. Note that the Load Shedding mechanism only reacts on situations, where the router is already congested and no more packets can be stored in the queue. Other queue management solutions like Active Queue Management (AQM) to monitor the queue state and take actions to prevent the router from congestion. When dropping packets, the router has to decide which packets should be dropped. There are two concepts for dropping packets. On the one hand the router can drop new incoming packets. This approach can be optimal for file transmissions. Consider a router that has packets 7, 8 and 9 of a 10 packets long file in its full queue. Now packet 10 arrives. Assuming a receiving host drops packets that arrive out of sequential order, the worst decision would be to drop packet 7, as it would lead to retransmission of packets 7 to 10. Dropping packet 10 would be optimal, as it would be the only packet that would have to be retransmitted. On the other hand for some multimedia and real-time applications, the value of new packets is higher than the value of old packets. In those cases, the router can be instructed to drop old packets. The concept of dropping new packets is called wine, dropping old packets is called milk. Wine is implemented as tail-drop on queues, milk is implemented as head-drop.

Fair Queuing

As they say Congestion Control and resource allocation are two sides of the same coin. To enforce hosts and routers to implement and use congestion control and avoidance algorithms described here, an open-loop mechanisms called Fair Queuing was introduced. Imagine hosts and routers connected to a router that reduces their data rates according to congestion control algorithms. A connected misbehaving sending host that does not reduce its data rates according to congestion control algorithms will overflow the router and use more resources than well behaving hosts. To resolve this issue, a FIFO queue is introduced on every incoming line as shown in figure 6.8.

Packets to deliver are chosen from all queues by a round robin algorithm. This ensures, that hosts or routers that misbehave do not use more resources and get a poorer service than they would, if they behaved well. Packets from misbehaving hosts will be dropped when they overflow their incoming FIFO queue on a router.

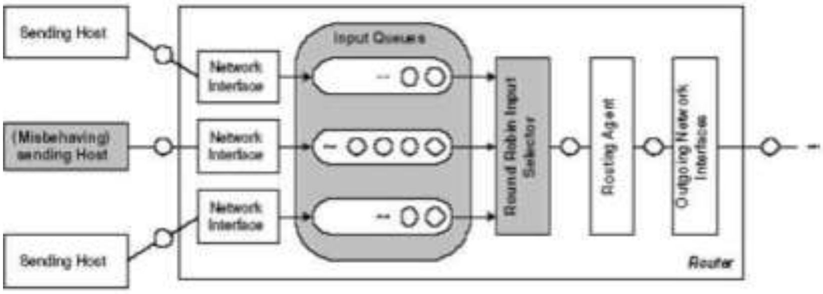


Figure 6.8: Fair Queuing Example

Admission Control

Generally used in closed-loop, Congestion Avoidance mechanism is Admission Control. Admission Control manages the establishment of connections between connection endpoints in connection-oriented networks. When a connection is established, the connected endpoints can rely on the expected service quality, as congestion cannot occur. On the Internet, Admission Control cannot be applied as it is working connectionless.

In connection-oriented networks, the transmissions belonging to a connection are always routed over the same network nodes. When a host tries to establish a connection, the virtual circuit has to be build up by all involved network nodes that have the chance to decide, if they can handle an additional connection or not. Answering this kind of question is not trivial, since it requires the knowledge of the current load on nodes as well as the expected additional load that results from the additional connection. Clients must declare their demand of service quality to aid this decision on the involved network nodes. For clients, this approach seems to be a bit restrictive. However, it is a very stable solution. In networks where connection establishment is only possible when the network load allows it, clients get a guarantee that the requested quality of service is fulfilled by the network. In such networks, there is no congestion during normal operation. A problem is to ensure that all participants follow the policies they advertised after the connections are established. If they use more service than they requested, the network can be congested. In connection-oriented networks, the network nodes have to save information about every connection, so it is possible for them to check, if clients do not exceed their contracts. If not, they should decrease their service to the badbehaving clients accordingly.

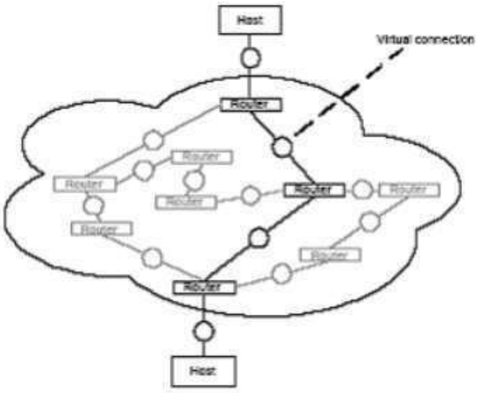


Figure 6.9: Admission control in computer networks

Examples of connection-oriented networks using admission control to avoid congestion are telephone networks or A synchronous Transfer Mode (ATM) networks. As Figure 6.9 shows admission control in computer networks.

3.4 IP ADDRESSING

INTRODUCTION

Computer needs to communicate with another computer somewhere else in the world. Usually, computers communicate through the Internet. The packet transmitted by the sending computer may pass through several LANs or WANs before reaching the destination computer. For this level of communication, we need a global addressing scheme; we called this logical addressing or IP address.

IPv4 Address Format

Internet Protocol being a layer-3 protocol (OSI) takes data Segments from layer-4 (Transport) and divides it into packets. IP packet encapsulates data unit received from above layer and add to its own header information. The encapsulated data is referred to as IP Payload. IP header contains all the necessary information to deliver the packet at the other end.

Version − Version no. of Internet Protocol used (e.g. IPv4).

IHL − Internet Header Length; Length of entire IP header.

Type of service: Low Delay, High Throughput, Reliability (8 bits)

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’.

Fragment Offset − This offset tells the exact position of the fragment in the original IP Packet.

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.

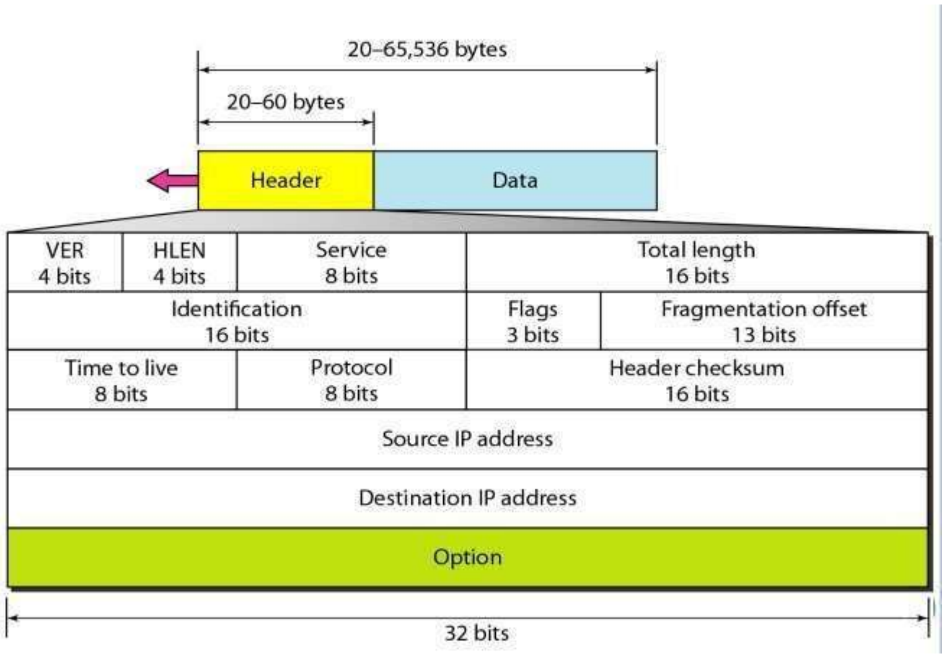
Protocol − Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol. For example protocol number of ICMP is 1, TCP is 6 and UDP is 17.

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.

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.



IPv4 ADDRESSES

An 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. IPv4 addresses are unique. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address at the same time.

Address Space

A protocol such as IPv4 that defines addresses has an address space.

IPv4 uses 32-bit addresses, which means that the address space is 232 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. But the actual number is much less because of the restrictions imposed on the addresses.

2.2. IPv4 Address Notations

There are two prevalent notations to show an IPv4 address:

a. Binary notation and

b. Dotted decimal notation.

a. 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

b. 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 3.12 shows an IPv4 address in both binary and dotted-decimal notation. Note that because each byte (octet) is 8 bits, each number in dotted-decimal notation is a value ranging from 0 to 255.

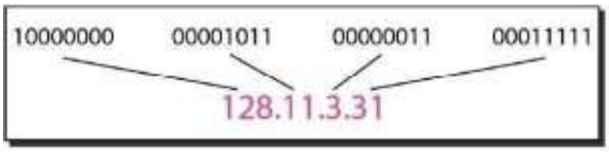


Figure 3.12 Dotted-decimal notation and binary notation for an IPv4 address

Example: Change the following IPv4 addresses from binary notation to dotted-decimal notation.

a. 10000001 00001011 00001011 1101111

b. 11000001 10000011 00011011 1111111 Solution

We replace each group of 8 bits with its equivalent decimal number (see Appendix B) and add dots for separation. a. 29.11.11.239

b.193.131.27.255

Example: Change the following IPv4 addresses from dotted-decimal notation to binary notation. a. 11.56.45.78

b. 21.34.7.82

Solution

We replace each decimal number with its binary equivalent.

a. 01101111 00111000 00101101 1001110

b. 11011101 00100010 00000111 1010010

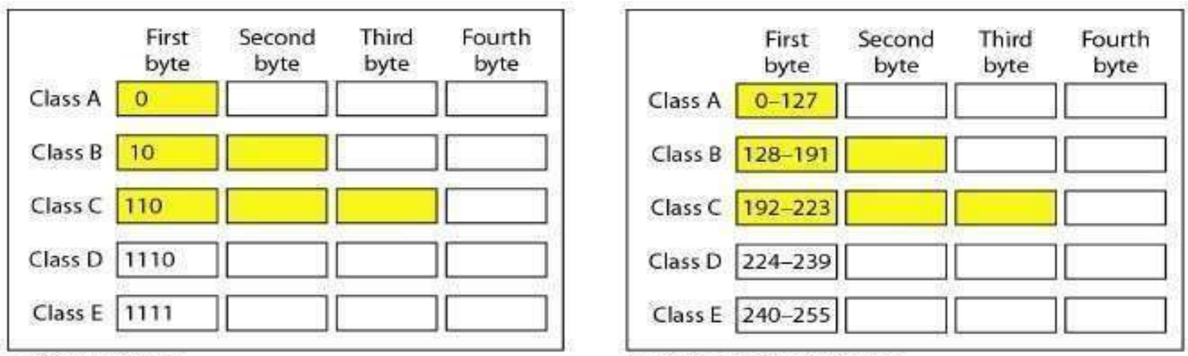
Types of IPv4 Addressing Schemes

There are two types of IPv4 addressing schemes:

- 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.
- 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. Both methods are shown in below figure



	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

Example: Find the class of each address.

a. 00000001 00001011 00001011 1101111

b. 11000001 10000011 00011011 1111111

c. 14.23.120.8

d. 52.5.15.111

Solution

a. The first bit is 0. This is a class A address.

b. The first 2 bits are 1; the third bit is 0. This is a class C address.

c. The first byte is 14 (between 0 and 127); the class is A.

d. The first byte is 252 (between 240 and 255); the class is E.

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 19.1. Table 19.1 Number of blocks and block size in Classful IPv4 addressing

Class	Number of Blocks	Block Size	Application
A	2^7 =128	2^{24} =16,777,216	Unicast

B	2^{14} =16,384	2^{16} =65,536	Unicast
C	2^{21} =2,097,152	2^{8} =256	Unicast
D	1	2^{28} =268,435,456	Multicast
E	1	2^{28} =268,435,456	Reserved

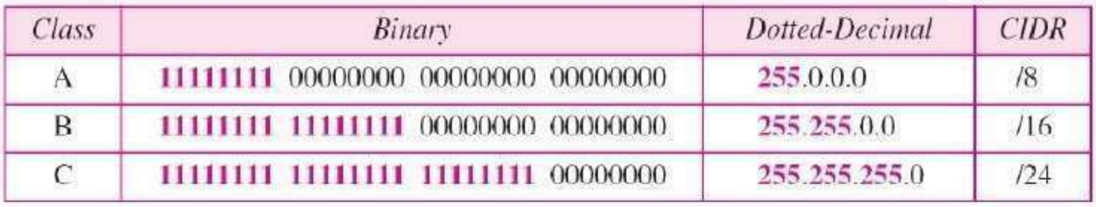
- Class A addresses were designed for large organizations with a large number of attached hosts or routers.
- Class B addresses were designed for midsize organizations with tens of thousands of attached hosts or routers.
- Class C addresses were designed for small organizations with a small number of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

Limitations of Classful Addressing:

- A block in class A address is too large for almost any organization. This means most of the addresses in class A were wasted and were not used.
- A block in class B is also very large, probably too large for many of the organizations that received a class B block.
- A block in class C is probably too small for many organizations.
- Class D addresses were designed for multicasting. Each address in this class is used to define one group of hosts on the Internet. The Internet authorities wrongly predicted a need for 268,435,456 groups. This never happened and many addresses were wasted here too.
- And lastly, the class E addresses were reserved for future use; only a few were used, resulting in another waste of addresses.

Netid and Hostid

- In classful addressing, an IP address in class A, B, or C is divided into netid and hostid.
- These parts are of varying lengths, depending on the class of the address. Figure 19.2 shows some netid and hostid bytes.
- 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.



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

Table 19.2 Default masks for classful addressing

Mask

A mask (also called the default mask) is a 32-bit number made of contiguous 1s followed by contiguous 0s. The masks for classes A, B, and C are shown in Table 19.2. The concept does not apply to classes D and E.

- The mask can help us to find the netid and the hostid. For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid.
- The last column of Table 19.2 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 Interdomain Routing (CIDR) notation.

Address Depletion Problem

The fast growth of the Internet led to the near depletion of the available addresses in classful addressing scheme. Yet the number of devices on the Internet is much less than the 232 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 that has alleviated the problem is the idea of classless addressing.
- Classful addressing, which is almost obsolete, is replaced with 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. For example, a household may be given only two addresses; a large organization may be given thousands of addresses. An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.
- The Internet authorities impose three restrictions on classless address blocks:

  o The addresses in a block must be contiguous, one after another. o The
  number of addresses in a block must be a power of 2 (1, 2, 4, 8, ...). o The
  first address must be evenly divisible by the number of addresses.

A better way to define a block of addresses is to select any address in the block and the mask. As we discussed before, 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. It is very convenient to give just the value of n preceded by a slash (CIDR notation).

- In 1Pv4 addressing, a block of addresses can be defined as x.y.z.t/n in which x.y.z.t defines one of the addresses and the /n defines the mask.
- The address and the /n notation completely define the whole block (the first address, the last address, and the number of addresses).

First Address: The first address in the block can be found by setting the 32 - n rightmost bits in the binary notation of the address to 0s.

Example

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

What is the first address in the block?

Solution

The binary representation of the given address is 11001101 00010000 00100101 00100111. If we set 32 - 28 rightmost bits to 0, we get 11001101 0001000 00100101 0010000 or 205.16.37.32. This is actually the block shown in Figure 19.3.

Last Address: The last address in the block can be found by setting the 32 - n rightmost bits in the binary notation of the address to 1s.

Example:

Find the last address for the block in Example 19.6.

Solution

The binary representation of the given address is 11001101 00010000 00100101 00100111. If we set 32 - 28 rightmost bits to 1, we get 11001101 00010000 00100101 0010 1111 or 205.16.37.47. This is actually the block shown in Figure 19.3.

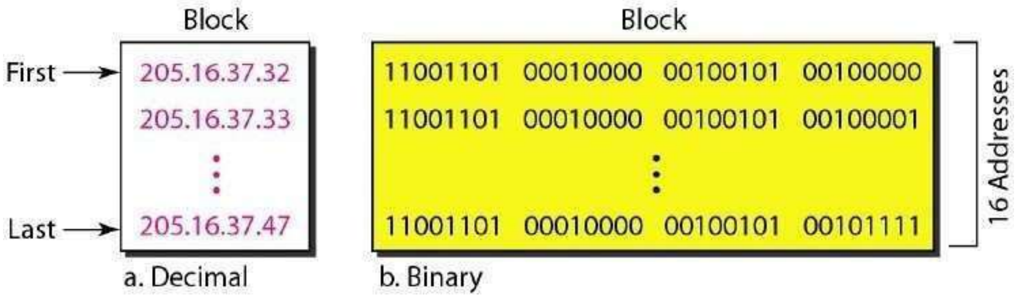


Figure 19.3 A block of16 addresses granted to a small organization

Number of Addresses: The number of addresses in the block is the difference between the last and first address. It can easily be found using the formula 232- n.

Example: Find the number of addresses in Example 19.6.

Solution

The value of n is 28, which means that number of addresses is 232- 28 or 16.

Example

Another way to find the first address, the last address, and the number of addresses is to represent the mask as a 32bit binary (or 8-digit hexadecimal) number. This is particularly useful when we are writing a program to find these pieces of information. In Example 19.5 the /28 can be represented as 11111111 11111111 11111111 11110000 (twenty-eight 1s and four 0s). Find

a. The first address

b. The last address

c. The number of addresses

Solution

a. The first address can be found by ANDing the given addresses with the mask. ANDing here is done bit by bit. The result of ANDing 2 bits is 1 if both bits are 1s; the result is 0 otherwise.

Address: 11001101 00010000 00100101 00100111
Mask: 11111111 11111111 11111111 11110000
First address: 11001101 00010000 00100101 0100000

b. The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit. The result of ORing 2 bits is 0 if both bits are 0s; the result is 1 otherwise. The complement of a number is found by changing each 1 to 0 and each 0 to 1.

Address: 11001101 00010000 00100101 00100111
Mask complement: 00000000 00000000 00000000 00001111
Last address: 11001101 00010000 00100101 00101111

c. The number of addresses can be found by complementing the mask, interpreting it as a decimal number, and adding 1 to it.

Mask complement: 000000000 00000000 00000000 00001111
Number of addresses: 15 + 1 =16

Network Addresses

A very important concept in IP addressing is the network address. When an organization is given a block of addresses, the organization is free to allocate the addresses to the devices that need to be connected to the Internet.

- The first address in the class, however, is normally (not always) treated as a special address. The first address is called the network address and defines the organization network.
- It defines the organization itself to the rest of the world. Usually the first address is the one that is used by routers to direct the message sent to the organization from the outside.

Figure 19.4 shows an organization that is granted a 16-address block. The organization network is connected to the Internet via a router. The router has two addresses. One belongs to the granted block; the other belongs to the network that is at the other side of the router. We call the second address x.y.z.t/n because we do not know anything about the network it is connected to at the other side. All messages destined for addresses in the organization block (205.16.37.32 to 205.16.37.47) are sent, directly or indirectly, to x.y.z.t/n. We say directly or indirectly because we do not know the structure of the network to which the other side of the router is connected.

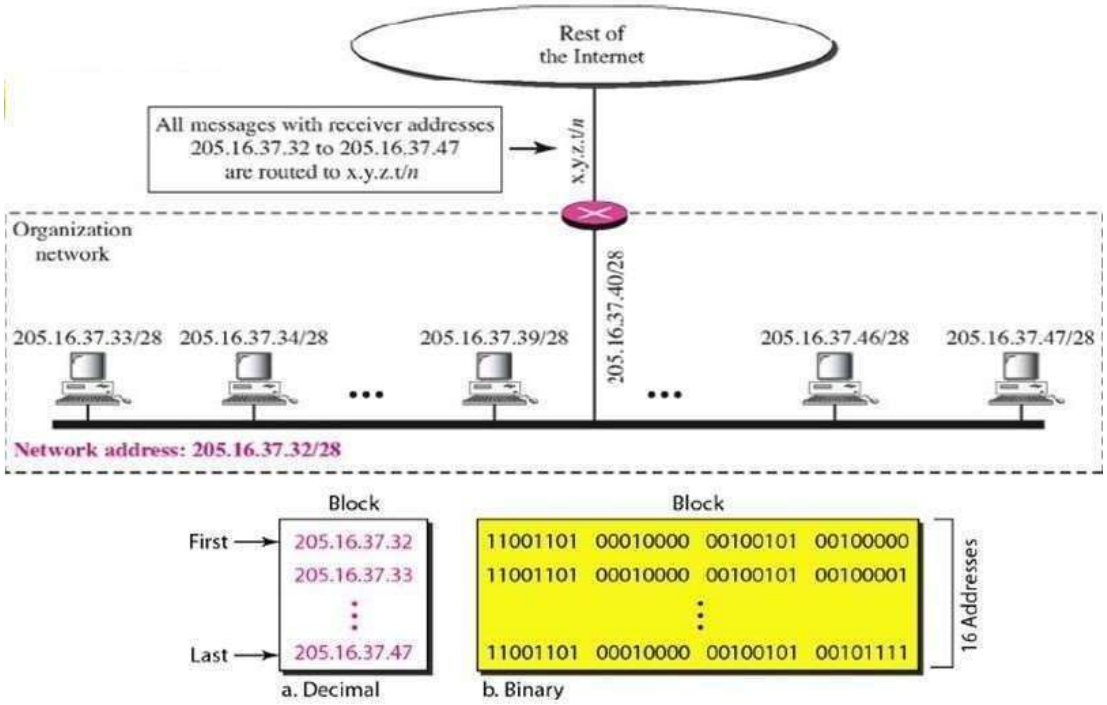


Figure 19.4 A network configuration for the block 205.16.37.32/28

The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.

Hierarchy

IP addresses, like other addresses or identifiers we encounter these days, have levels of hierarchy.

Two-Level Hierarchy: No Subnetting

An IP address can define only two levels of hierarchy when not subnetted.

- The n leftmost bits of the address x.y.z.t/n define the network (organization network).
- The 32 – n rightmost bits define the particular host (computer or router) to the network.
- The two common terms are prefix and suffix.
- The part of the address that defines the network is called the prefix; the part that defines the host is called the suffix. Figure 19.5 shows the hierarchical structure of an IPv4 address.

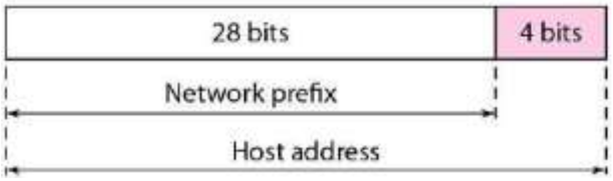


Figure 19.5 Two levels of hierarchy in an IPv4 address

- The prefix is common to all addresses in the network; the suffix changes from one device to another.
- Each address in the block can be considered as a two-level hierarchical structure: the leftmost n bits (prefix) define the network; the rightmost 32 - n bits define the host.

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. The rest of the world still sees the organization as one entity; however, internally there are several subnets. All messages are sent to the router address that connects the organization to the rest of the Internet; the router routes the message to the appropriate subnets. The organization, however, needs to create small sub blocks of addresses, each assigned to specific subnets. The organization has its own mask; each subnet must also have its own.

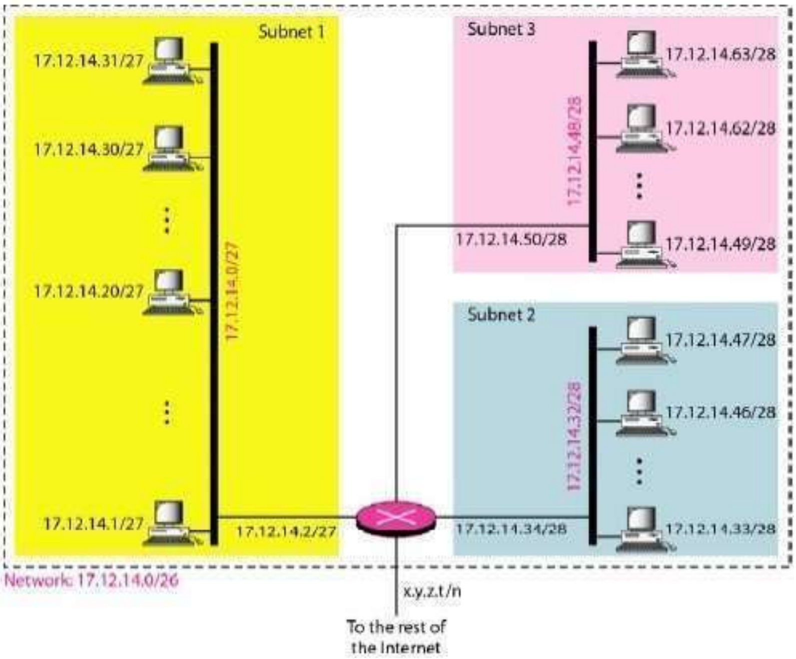


Figure 19.6 Configuration and addresses in a subnetted network

Example, suppose an organization is given the block 17.12.40.0/26, which contains 64 addresses. The organization has three offices and needs to divide the addresses into three sub blocks of 32, 16, and 16 addresses. We can find the new masks by using the following arguments:

1. Suppose the mask for the first 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. Figure 19.6 shows one configuration for the above scenario.

Let us check to see if we can find the subnet addresses from one of the addresses in the subnet.

a. In subnet 1, the address 17.12.14.29/27 can give us the subnet address if we use the mask /27 because

Host: 00010001 00001100 00001110 00011101 Mask:
/27
Subnet: 00010001 00001100 00001110 00000000 ...... (17.12.14.0)

b. In subnet 2, the address 17.12.14.45/28 can give us the subnet address if we use the mask /28 because

Host: 00010001 00001100 00001110 00101101 Mask:
/28
Subnet: 00010001 00001100 00001110 00100000 ...... (17.12.14.32)

c. In subnet 3, the address 17.12.14.50/28 can give us the subnet address if we use the mask /28 because

Host: 00010001 00001100 00001110 00110010 Mask:
/28
Subnet: 00010001 00001100 00001110 00110000 ...... (17.12.14.48)

Note that applying the mask of the network, /26, to any of the addresses gives us the network address 17.12.14.0/26. We can say that through subnetting, we have three levels of hierarchy. Note that in our example, the subnet prefix length can differ for the subnets as shown in Figure 19.7.

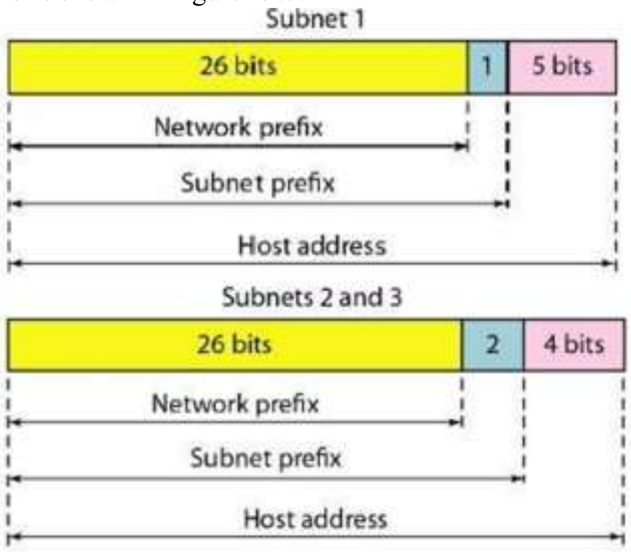


Figure 19.7 Three-level hierarchy in an IPv4 address

More Levels of Hierarchy

The structure of classless addressing does not restrict the number of hierarchical levels. An organization can divide the granted block of addresses into sub blocks. Each sub block can in turn be divided into smaller sub blocks. And so on. One example of this is seen in the ISPs.

- A national ISP can divide a granted large block into smaller blocks and assign each of them to a regional ISP.
- A regional ISP can divide the block received from the national ISP into smaller blocks and assign each one to a local ISP.
- A local ISP can divide the block received from the regional ISP into smaller blocks and assign each one to a different organization.
- Finally, an organization can divide the received block and make several subnets out of it.

Address Allocation

- The next issue in classless addressing is address allocation. How are the blocks allocated? The ultimate responsibility of address allocation is given to a global authority called the Internet Corporation for Assigned Names and Addresses (ICANN).
- However, ICANN does not normally allocate addresses to individual organizations. It assigns a large block of addresses to an ISP. Each ISP, in turn, divides its assigned block into smaller sub blocks and grants the sub blocks to its customers.
- In other words, an ISP receives one large block to be distributed to its Internet users.

o This is called address aggregation: many blocks of addresses are aggregated in one block and granted to one ISP.

Example

An ISP is granted a block of addresses starting with 190.100.0.0/16 (65,536 addresses). The ISP needs to distribute these addresses to three groups of customers as follows:

a. The first group has 64 customers; each needs 256 addresses.
b. The second group has 128 customers; each needs 128 addresses.
c. The third group has 128 customers; each needs 64 addresses.

Design the sub blocks and find out how many addresses are still available after these allocations. Solution

Figure 19.8 shows the situation.

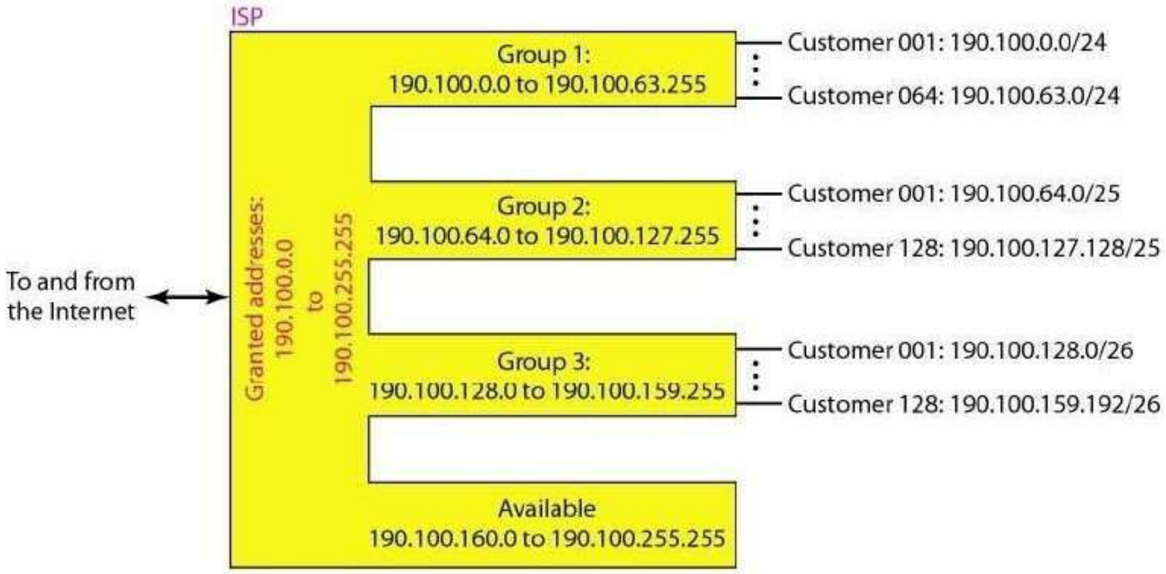

ISP
Group 1:
190.100.0.0 to 190.100.63.255
Customer 001: 190.100.0.0/24
Customer 064: 190.100.63.0/24
Group 2:
190.100.64.0 to 190.100.127.255
Customer 001: 190.100.64.0/25
Customer 128: 190.100.127.128/25
Group 3:
190.100.128.0 to 190.100.159.255
Customer 001: 190.100.128.0/26
Customer 128: 190.100.159.192/26
Available
190.100.160.0 to 190.100.255.255
Granted addresses:
190.100.0.0
to
190.100.255.255
To and from
the Internet

Figure 19.8 An example of address allocation and distribution by an ISP

1. Group 1

For this group, each customer needs 256 addresses. This means that 8 (log2256) bits are needed to define each host. The prefix length is then 32 - 8 =24. The addresses are

1st Customer:	190.100.0.0/24	190.100.0.255/24
2nd Customer:	190.100.1.0/24	190.100.1.255/24

64th Customer:	190.100.63.0/24	190.100.63.255/24
Total =64 X 256 =16,384		

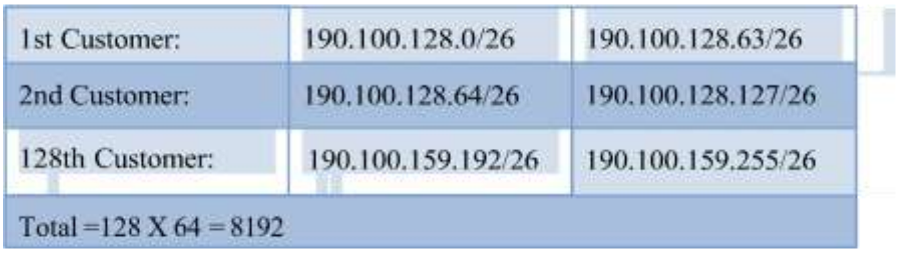
2. Group2

For this group, each customer needs 128 addresses. This means that 7 (log2 128) bits are needed to define each host. The prefix length is then 32 - 7 =25. The addresses are:

1st Customer:	190.100.64.0/25	190.100.64.127/25
2nd Customer:	190.100.64.128/25	190.100.64.255/25
128th Customer:	190.100.127.128/25	190.100.127.255/25
Total =128 X 128 = 16,384		

3. Group3

For this group, each customer needs 64 addresses. This means that 6 (log2 64) bits are needed to each host. The prefix length is then 32 - 6 =26. The addresses are



1st Customer:	190.100.128.0/26	190.100.128.63/26
2nd Customer:	190.100.128.64/26	190.100.128.127/26
128th Customer:	190.100.159.192/26	190.100.159.255/26
Total =128 X 64 = 8192		

To understand what Variable Length Subnet Masking (VLSM) is, let's go through a business example. Suppose a company has bought the IP address range 37.1.1.0/24 (256 addresses). The company has 5 offices, as shown in figure 1 below:

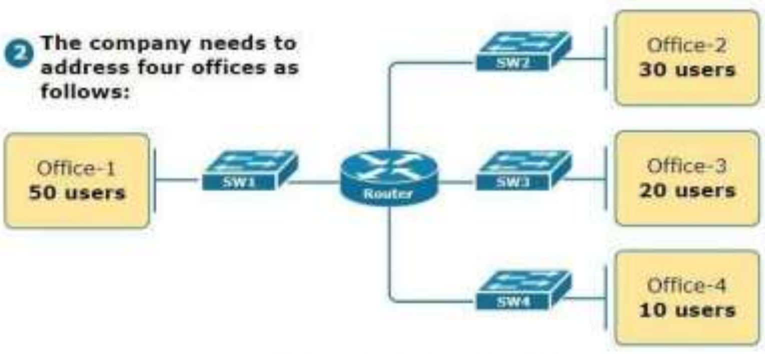


Figure 1. Subnetting Business Requirements

We are tasked to subnet the 37.1.1.0/24 IP address block and assign each office an IP subnet. Let's see what the two ways to do this are.

What is FLSM?

The first approach to this task is to divide the 256-address block into four equal-sized subnets. This technique is called Fixed Lenght Subnet Mask (FLSM). The benefit of this approach is that all subnets have the same subnet mask, which makes the process very straightforward and less prone to errors. Figure 2 below illustrates this example.

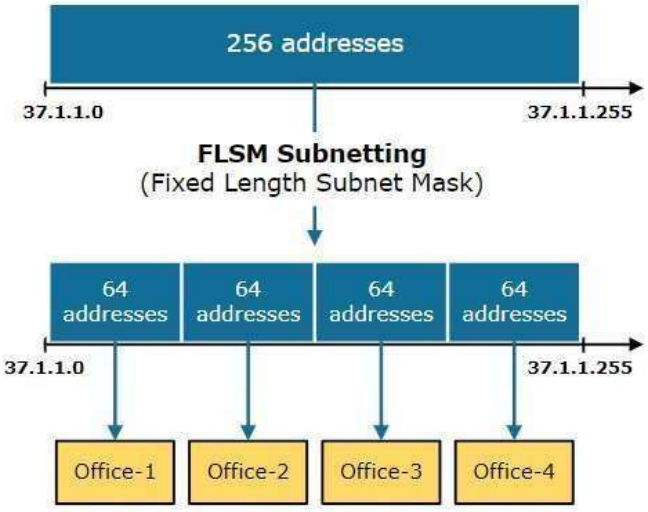


Figure 2. What is FLSM?

However, this method results in a significant waste of IP addresses. For example, office-4 has only 10 users, but we assign a subnet with 64 IP addresses. Hence, 54 addresses sit unused. From the company's point of view, this is a bad use of resources.

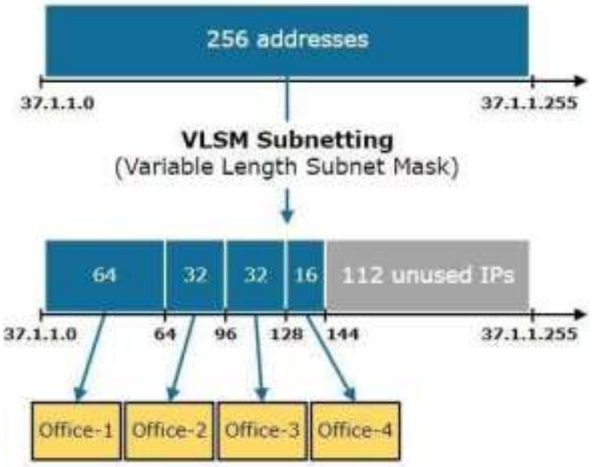


Figure 3: What is VLSM?

VLSM stands for Variable Length Subnet Mask. VLSM is a subnetting technique that allows network admins to allocate IP addresses more efficiently using different subnet masks for different network segments. It provides greater flexibility in assigning IP addresses by creating subnets of varying sizes based on the specific needs and number of hosts in each subnet. This technique helps reduce the waste of IP addresses and better uses the available IP space.

Notice that using VLSM, we are left with 112 free IP addresses that we can allocate to another location in the future. Compared to the FLSM approach, this method is much more efficient. The main idea is that VLSM allows us to divide an IP address space into subnets of varying sizes based on the specific requirements of each office. This helps to minimize the waste of IP addresses, as each subnet is assigned only the necessary number of IPs instead of using fixed-size subnets that may be too large or too small for the purpose.

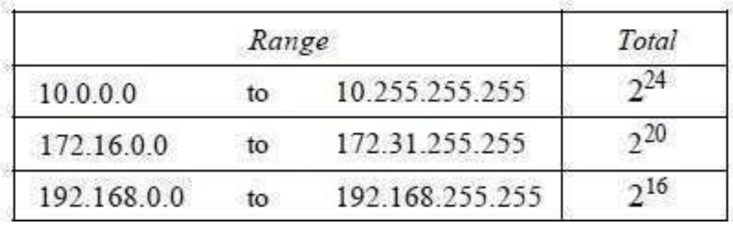
Network Address Translation (NAT)

The number of home users and small businesses that want to use the Internet is ever increasing. In the beginning, a user was connected to the Internet with a dial-up line, which means that she was connected for a specific period of time. An ISP with a block of addresses could dynamically assign an address to this user. An address was given to a user when it was needed. But the situation is different today. Home users and small businesses can be connected by an ADSL line or cable modem. In addition, many are not happy with one address; many have created small networks with several hosts and need an IP address for each host. With the shortage of addresses, this is a serious problem.

A quick solution to this problem is called network address translation (NAT).

- NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally. The traffic inside can use the large set; the traffic outside, the small set.□
- To separate the addresses used inside the home or business and the ones used for the Internet, the Internet authorities have reserved three sets of addresses as private addresses, shown in Table 19.3.□

Table 19.3 Addresses for private networks



Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

- Any organization can use an address out of this set without permission from the Internet authorities. Everyone knows that these reserved addresses are for private networks.□
- They are unique inside the organization, but they are not unique globally. No router will forward a packet that has one of these addresses as the destination address.□
- The site must have only one single connection to the global Internet through a router that runs the NAT software.□

Figure 19.9 shows a simple implementation of NAT. As Figure 19.9 shows, the private network uses private addresses. The router that connects the network to the global address uses one private address and one global address. The private network is transparent to the rest of the Internet; the rest of the Internet sees only the NAT router with the address 200.24.5.8.

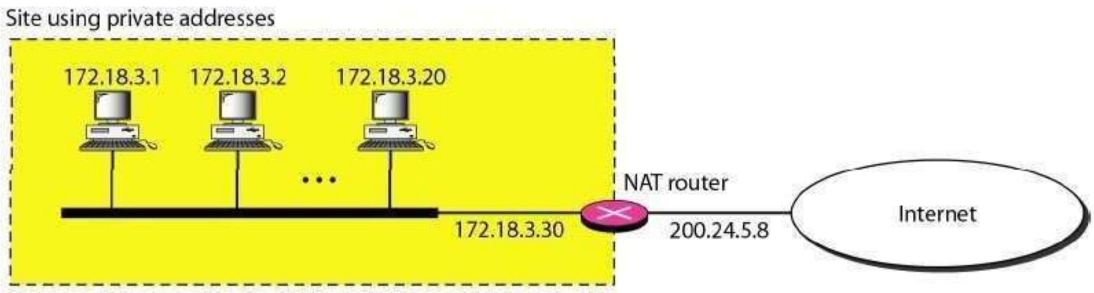


Figure 19.9 A NAT implementation

Address Translation

- All the outgoing packets go through the NAT router, which replaces the source address in the packet with the global NAT address.
- All incoming packets also pass through the NAT router, which replaces the destination address in the packet (the NAT router global address) with the appropriate private address. Figure 19.10 shows an example of address translation.

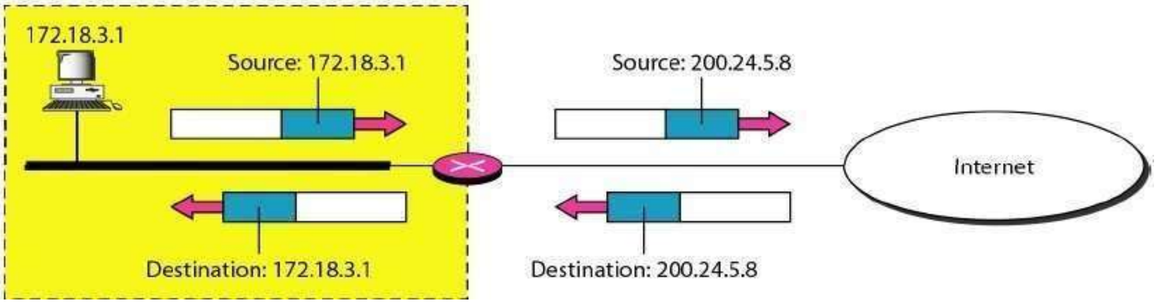


Figure 19.10 Addresses in a NAT

IPv6 ADDRESSES

Despite all short-term solutions, such as classless addressing and NAT, address depletion is still a long-term problem for the Internet. This and other problems in the IP protocol itself, such as lack of accommodation for real-time audio and video transmission, and encryption and authentication of data for some applications, 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. 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, as shown in Figure 19.13.

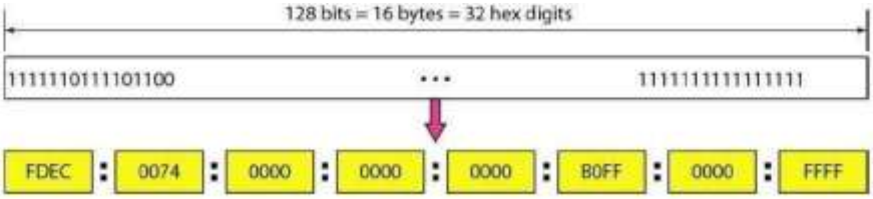


Figure 19.13 IPv6 address in binary and hexadecimal colon notation

Abbreviation

Although 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. Only the leading zeros can be dropped, not the trailing zeros (see Figure 19.14).

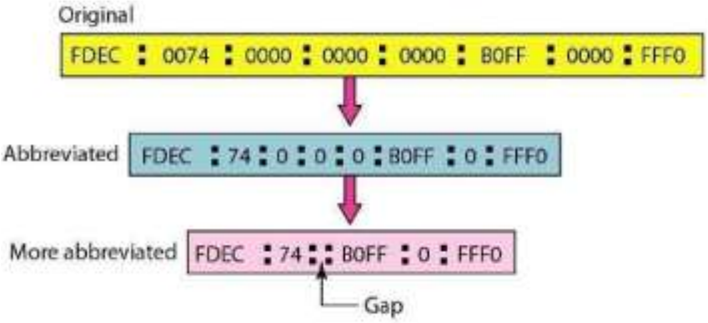

Original
FDEC : 0074 : 0000 : 0000 : 0000 : B0FF : 0000 : FFF0
Abbreviated
FDEC : 74 : 0 : 0 : 0 : B0FF : 0 : FFF0
More abbreviated
FDEC : 74 :: B0FF : 0 : FFF0
Gap

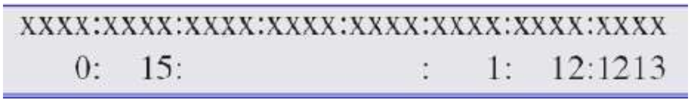
Figure 19.14 Abbreviated IPv6 addresses

Using this form of abbreviation, 0074 can be written as 74, 000F as F, and 0000 as 0. Note that 3210 cannot be abbreviated. Further abbreviations are possible if there are consecutive sections consisting of zeros only. We can remove the zeros altogether and replace them with a double semicolon. Note that this type of abbreviation is allowed only once per address. If there are two runs of zero sections, only one of them can be abbreviated. Reexpansion of the abbreviated address is very simple: Align the unabbreviated portions and insert zeros to get the original expanded address.

Example: Expand the address 0:15::1:12:1213 to its original.

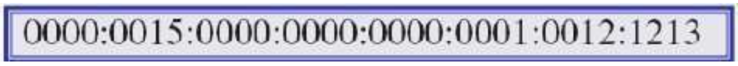
Solution

We first need to align the left side of the double colon to the left of the original pattern and the right side of the double colon to the right of the original pattern to find now many 0s we need to replace the double colon.



```
XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX
   0:  15:                 :   1:  12:1213
```

This means that the original address is



0000:0015:0000:0000:0000:0001:0012:1213

Routing Information Protocol (RIP)

Routing Information Protocol or RIP is one of the first routing protocols to be created. RIP is used in both Local Area Networks (LANs) and Wide Area Networks (WANs), and also runs on the Application layer of the OSI model. There are multiple versions of RIP including RIPv1 and RIPv2. The original version or RIPv1 determines network paths based on the IP destination and the hop count of the journey.

RIPv1 interacts with the network by broadcasting its IP table to all routers connected to the network. RIPv2 is a little more sophisticated than this and sends its routing table on to a multicast address. RIPv2 also uses authentication to keep data more secure and chooses a subnet mask and gateway for future traffic. The main limitation of RIP is that it has a maximum hop count of 15 which makes it unsuitable for larger networks.

Pros:

- Historical Significance: RIP is one of the oldest and widely recognized routing protocols.□
- Operational Simplicity: It’s relatively straightforward to understand and implement.□
- Application Layer Operation: Operates on the application layer, making it easy to manage and configure.□
- Multicast Capability (RIPv2): RIPv2 can multicast its routing table, providing a more efficient way to communicate with other routers than broadcasting.□
- Enhanced Security (RIPv2): RIPv2 offers authentication measures to enhance data security.□

Cons:

- Maximum Hop Count: RIP’s maximum hop count of 15 restricts its use in larger networks.□
- Lack of Scalability: Due to its hop count limitation, it is not suited for modern expansive networks.□
- Broadcaster (RIPv1): RIPv1’s method of broadcasting its entire table can lead to increased traffic and potential inefficiencies.□
- Limited Route Metric: RIP uses hop count as its sole metric, which may not always represent the best path in complex networks.□
- Slower Convergence: RIP can be slower to adapt to network changes, leading to potential temporary routing loops.□

Open Shortest Path First (OSPF)

Open Shortest Path First or OSPF protocol is a link-state IGP that was tailor-made for IP networks using the Shortest Path First (SPF) algorithm. The SPF routing algorithm is used to calculate the shortest path spanning-tree to ensure efficient data transmission of packets. OSPF routers maintain databases detailing information about the surrounding topology of the network. This database is filled with data taken from Link State Advertisements (LSAs) sent by other routers. LSAs are packets that detail information about how many resources a given path would take.

OSPF also uses the Dijkstra algorithm to recalculate network paths when the topology changes. This protocol is also relatively secure as it can authenticate protocol changes to keep data secure. It is used by many organizations because it’s scalable to large environments. Topology changes are tracked and OSPF can recalculate compromised packet routes if a previously-used route has been blocked.

Pros:

- Efficient Routing: Utilizes the Shortest Path First (SPF) algorithm to ensure optimal data packet transmission.□
- Detailed Network Insight: OSPF routers maintain a database on the network's topology, offering a detailed perspective on its structure.□
- Dynamic Adaptability: Employs the Dijkstra algorithm to dynamically adjust to network topology changes, ensuring continuity in data transmission.□
- Security Features: Offers protocol change authentication to maintain data security, ensuring that only authorized updates are made.□
- Highly Scalable: Suitable for both small and large-scale network environments, making it versatile for various organizational sizes.□

Cons:

- Complex Configuration: Given its many features, OSPF can be complex to set up and maintain.□
- Higher Overhead: Maintaining detailed databases and frequently recalculating routes can generate more network overhead.□
- Sensitive to Topology Changes: While OSPF can adapt to changes, frequent topology alterations can cause performance dips as it recalculates routes.□
- Resource Intensive: OSPF routers require more memory and CPU resources due to their database maintenance and route recalculations.□
- Potential for Large LSDB: In very large networks, the Link State Database (LSDB) can grow significantly, necessitating careful design and segmenting.□

Border Gateway Protocol (BGP)

Border Gateway Protocol or BGP is the routing protocol of the internet that is classified as a distance path vector protocol. BGP was designed to replace EGP with a decentralized approach to routing. The BGP Best Path Selection Algorithm is used to select the best routes for data packet transfers. If you don't have any custom settings then BGP will select routes with the shortest path to the destination.

However many administrators choose to change routing decisions to criteria in line with their needs. The best routing path selection algorithm can be customized by changing the BGP cost community attribute. BGP can make routing decisions based Factors such as weight, local preference, locally generated, AS_Path length, origin type, multi-exit discriminator, eBGP over iBGP, IGP metric, router ID, cluster list and neighbor IP address.

BGP only sends updated router table data when something changes. As a result, there is no auto-discovery of topology changes which means that the user has to configure BGP manually. In terms of security, BGP protocol can be authenticated so that only approved routers can exchange data with each other.

Pros:

- Internet Backbone: As the primary routing protocol of the internet, BGP plays a pivotal role in global data exchanges.□

- Decentralized Design: Unlike its predecessor EGP, BGP's decentralized nature ensures more robust and adaptable network operations.□
- Customizable Path Selection: BGP's Best Path Selection Algorithm can be tailored to meet unique network demands by adjusting attributes.□
- Efficient Updates: Only transmitting updates when there's a change, BGP reduces unnecessary network traffic.□
- Granular Routing Decisions: Administrators have a plethora of factors like weight, AS_Path length, and IGP metric to inform routing decisions, allowing for a high degree of routing precision.□
- Authentication: BGP provides security measures allowing only authorized routers to participate in data exchanges, enhancing the security of routing updates.□

Cons:

- Complex Configuration: BGP requires meticulous manual configuration since it doesn't auto-discover topology changes.□
- Potential Instability: Mistakes or malicious actions in BGP configurations can inadvertently or intentionally divert internet traffic, potentially leading to large-scale outages.□
- Scalability Concerns: As the internet grows, BGP's scalability, in its current form, might pose challenges.□
- Vulnerabilities: Despite authentication measures, BGP is historically susceptible to certain security issues, like prefix hijacking.□
- Learning Curve: Given its complexity and significance, mastering BGP can be challenging for many network administrators.□
- Convergence Time: BGP can sometimes take longer to converge after a network change compared to some other protocols.□

DHCP ARP and ICMP

DHCP (Dynamic Host Configuration Protocol): DHCP is a protocol that automatically assigns IP addresses to devices on a network. When a device connects to a network, it sends a DHCP request, and the DHCP server responds with an IP address, subnet mask, default gateway, and other network configuration information.

ARP (Address Resolution Protocol): ARP is a protocol that maps IP addresses to MAC addresses. When a device wants to send a packet to another device on the same network, it uses ARP to find the MAC address associated with the destination IP address.

ICMP (Internet Control Message Protocol): ICMP is a protocol that is used for network diagnostics and troubleshooting. ICMP messages are typically generated by network devices when errors occur, and they are used to provide feedback to the sender about the status of the network. These protocols work together to enable communication between devices on an IP network. DNS provides a way to translate human-readable domain names into machine-readable IP addresses, DHCP simplifies the process of IP address assignment, ARP is used to resolve IP addresses to MAC addresses for communication on the local network, and ICMP provides feedback about the status of the network. Together, these protocols form the backbone of IP networking and make it possible for devices to communicate with one another over the internet.

DHCP Process:

□ The dynamic host configuration protocol is used to simplify the installation and maintenance of networked computers.

□ DHCP is derived from an earlier protocol called BOOTP.

□ Ethernet addresses are configured into network by manufacturer and they are unique.

□ IP addresses must be unique on a given internetwork but also must reflect the structure of the internetwork

□ Most host Operating Systems provide a way to manually configure the IP information for the host

□ Drawbacks of manual configuration :

1. A lot of work to configure all the hosts in a large network
2. Configuration process is error-prune

□ It is necessary to ensure that every host gets the correct network number and that no two hosts receive the same IP address.

□ For these reasons, automated configuration methods are required.

□ The primary method uses a protocol known as the Dynamic Host Configuration Protocol (DHCP).

□ The main goal of DHCP is to minimize the amount of manual configuration required for a host.

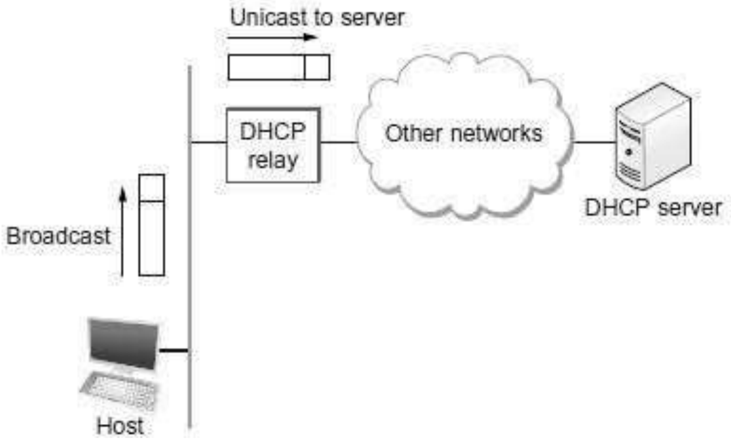
□ If a new computer is connected to a network, DHCP can provide it with all the necessary information for full system integration into the network.

□ DHCP is based on a client/server model.

□ DHCP clients send a request to a DHCP server to which the server responds with an IP address □ DHCP server is responsible for providing configuration information to hosts.

□ There is at least one DHCP server for an administrative domain.

□ The DHCP server can function just as a centralized repository for host configuration information. □ The DHCP server maintains a pool of available addresses that it hands out to hosts on demand.



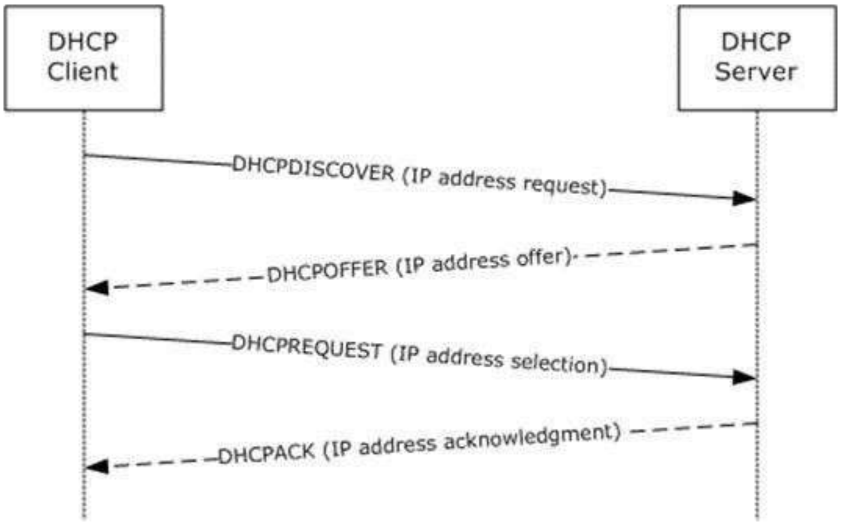
□ A newly booted or attached host sends a DHCPDISCOVER message to a special IP address (255.255.255.255., which is an IP broadcast address.

 This means it will be received by all hosts and routers on that network.

 DHCP uses the concept of a relay agent. There is at least one relay agent on each network.

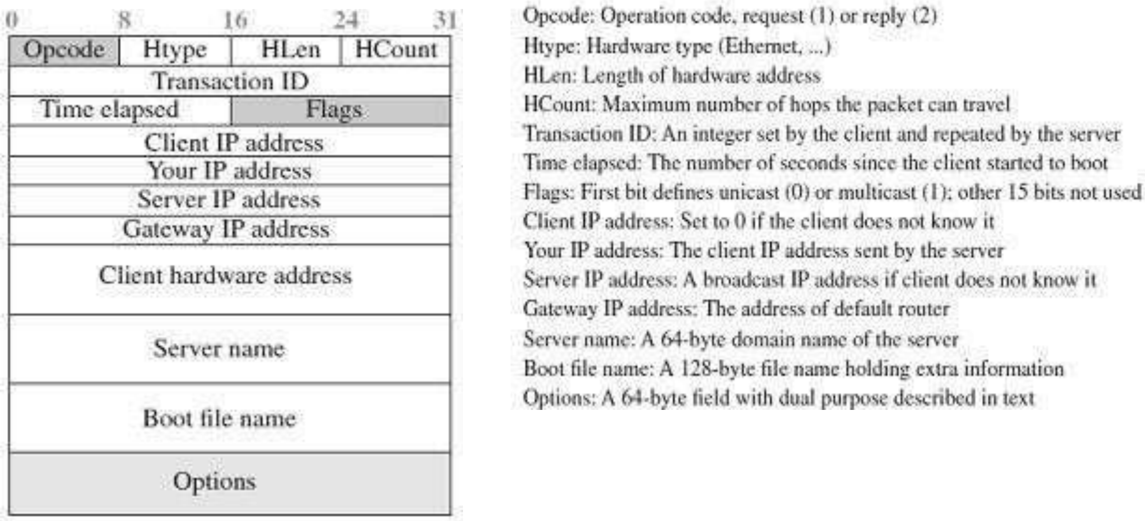
 DHCP relay agent is configured with the IP address of the DHCP server.

 When a relay agent receives a DHCPDISCOVER message, it unicasts it to the DHCP server and awaits the response, which it will then send back to the requesting client.


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

DHCP Message Format

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



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

Exercises:

1. Write the Network, broadcast address, and a valid host range for question a through question a. 192.168.100.25/30

   b. . 192.168.100.37/28

   c. 192.168.100.66/27

   d. 192.168.100.17/29

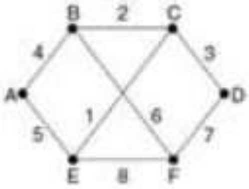
   e. 192.168.100.99/26

f. 192.168.100.99/25

2. What is the broadcast address of 192.168.192.10/29?
3. What is the subnet for host ID 10.16.3.65/23?
4. A large number of consecutive IP addresses are available starting at 192.168.0.0. Suppose that four organizations, A, B, C, and D, request 2000, 4000, 8000, and 4000 addresses, respectively, and in that order. For each of these, give the first IP address assigned, the last IP address assigned, and the mask in the x.x.x.x/s notation.
5. Complete the following based on the decimal IP address.

Decimal IP Address	Address Class	Number of		Number of Subnets (2^x)	Number of Hosts ($2^x – 2$)
		Subnet Bits	Host Bits		
10.25.66.154/23					
172.31.254.12/24					
192.168.20.123/28					
63.24.89.21/18					
128.1.1.254/20					
208.100.54.209/30					

6. Consider the network shown below. Distance vector routing is used. Calculate C's new routing table?


B
2
C
4
3
A
D
1
6
5
7
E
8
F