MODULE 3: THE NETWORK LAYER

What's Inside a Router?

• The router is used for transferring packets from an incoming-links to the appropriate outgoing-links.

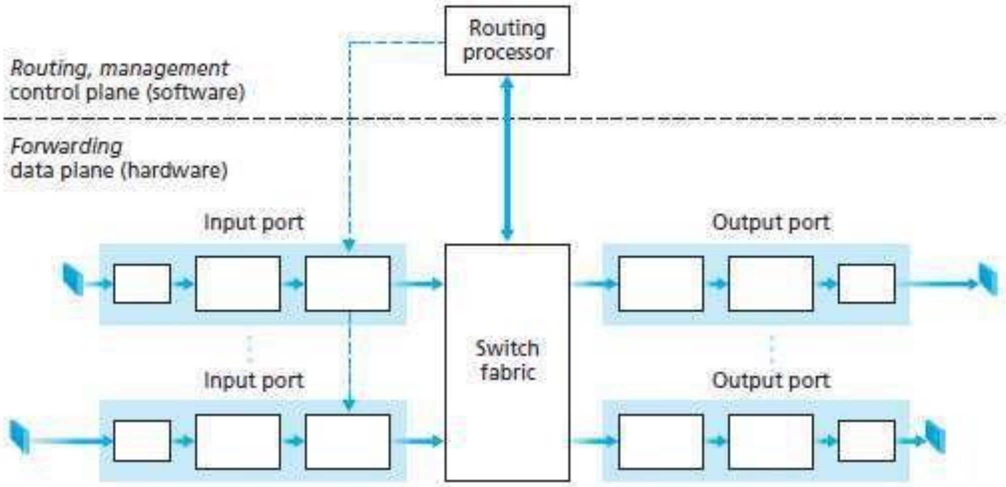


Figure 3.5: Router architecture

• Four components of router (Figure 3.5):

1) Input Ports

• An input-port is used for terminating an incoming physical link at a router (Figure 3.6).
• It is used for interoperating with the link layer at the other side of the incoming-link.
• It is used for lookup function i.e. searching through forwarding-table looking for longest prefix match.
• It contains forwarding-table.
• Forwarding-table is consulted to determine output-port to which arriving packet will be forwarded.
• Control packets are forwarded from an input-port to the routing-processor.
• Many other actions must be taken:
    i) Packet's version number, checksum and time-to-live field must be checked.
    ii) Counters used for network management must be updated.

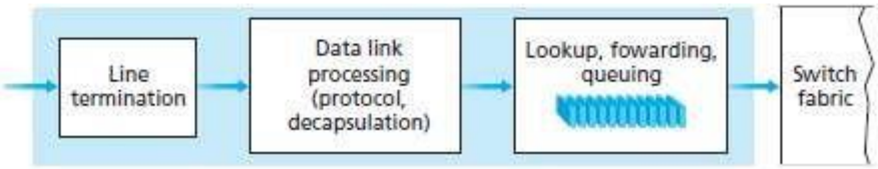


Figure 3.6: Input port processing

2) Switching Fabric

• The switching fabric connects the router's input-ports to its output-ports.
• In fabric, the packets are switched (or forwarded) from an input-port to an output-port.
• In fact, fabric is a network inside of a router.
• A packet may be temporarily blocked if packets from other input-ports are currently using the fabric.
• A blocked packet will be queued at the input-port & then scheduled to send at a later point in time.

3) Output Ports

• An output-port
    → stores packets received from the switching fabric and
    → transmits the packets on the outgoing-link.
• For a bidirectional link, an output-port will typically be paired with the input-port.

4) Routing Processor

• The routing-processor
    → executes the routing protocols

→ maintains routing-tables & attached link state information and
→ computes the forwarding-table.

• It also performs the network management functions.

3.3.1 Switching

• Three types of switching fabrics (Figure 3.7):

1) Switching via memory
2) Switching via a bus and
3) Switching via an interconnection network.

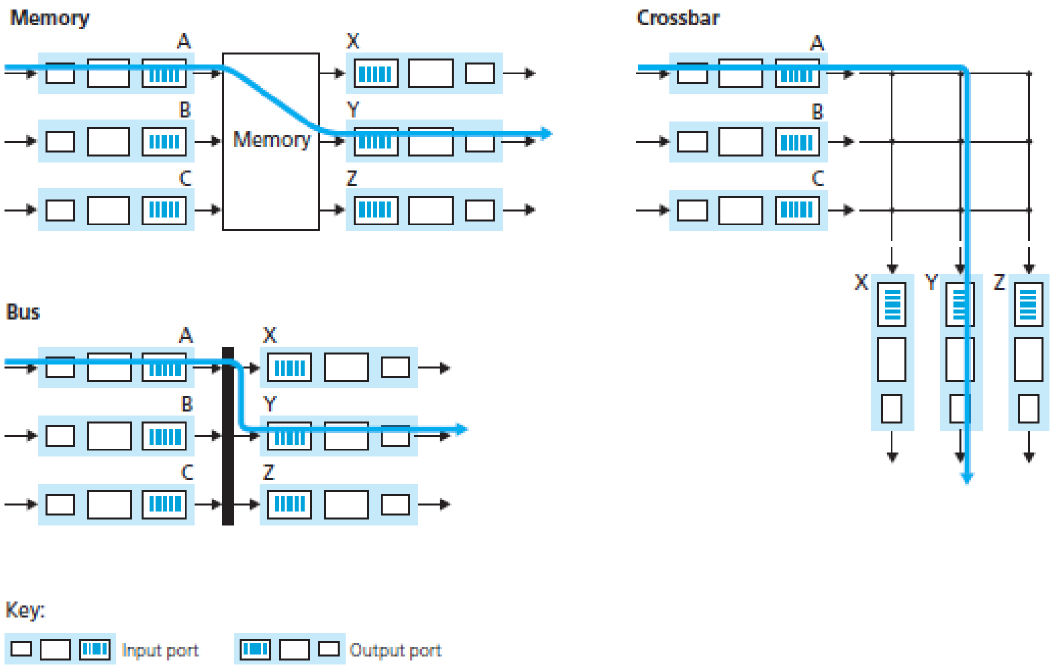


Figure 3.7: Three switching techniques

3.3.1.1 Switching via Memory

• Switching b/w input-ports & output-ports is done under direct control of CPU i.e. routing-processor.
• Input and output-ports work like a traditional I/O devices in a computer.
• Here is how it works (Figure 3.7a):

i) On arrival of a packet, the input-port notifies the routing-processor via an interrupt.
ii) Then, the packet is copied from the input-port to processor-memory.
iii) Finally, the routing-processor
→ extracts the destination-address from the header
→ looks up the appropriate output-port in the forwarding-table and
→ copies the packet into the output-port's buffers.

• Let memory-bandwidth = B packets per second.
Thus, the overall forwarding throughput must be less than B/2.
• Disadvantage:
➢ Multiple packets cannot be forwarded at the same time. This is because
→ only one memory read/write over the shared system bus can be done at a time.

3.3.1.2 Switching via a Bus

• Switching b/w input-ports & output-ports is done without intervention by the routing-processor.
• Here is how it works (Figure 3.7b):

i) The input-port appends a switch-internal label (header) to the packet.
➢ The label indicates the local output-port to which the packet must be transferred.
ii) Then, the packet is received by all output-ports.
➢ But, only the port that matches the label will keep the packet.
iii) Finally, the label is removed at the output-port.

• Disadvantages:
i) Multiple packets cannot be forwarded at the same time. This is because

→ only one packet can cross the bus at a time.

ii) The switching speed of the router is limited to the bus-speed.

3.3.1.3 Switching via an Interconnection Network

• A crossbar switch is an interconnection network.
• The network consists of 2N buses that connect N input-ports to N output-ports.
• Each vertical bus intersects each horizontal bus at a crosspoint.
• The crosspoint can be opened or closed at any time by the switch-controller.
• Here is how it works (Figure 3.7c):

1) To move a packet from port A to port Y, the switch-controller closes the crosspoint at the intersection of buses A and Y.

2) Then, port A sends the packet onto its bus, which is picked up by bus Y.

• Advantage:

➢ Crossbar networks are capable of forwarding multiple packets in parallel.

➢ For ex: A packet from port B can be forwarded to port X at the same time. This is because

→ A-to-Y and B-to-X packets use different input and output buses.

• Disadvantage:

➢ If 2 packets have to use same output-port, then one packet has to wait. This is because

→ only one packet can be sent over any given bus at a time.

3.3.2 Output Processing

• Output-port processing

→ takes the packets stored in the output-port's memory and

→ transmits the packets over the output link (Figure 3.8).

• This includes

→ selecting and dequeueing packets for transmission and

→ performing the linklayer and physical-layer transmission functions.

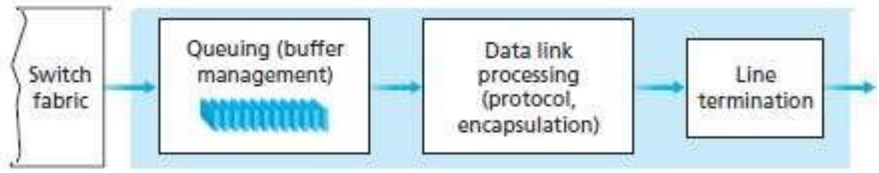


Figure 3.8: Output port processing

3.3.3 Where Does Queueing Occur?

• Packet queues may form at both the input-ports & the output-ports (Figure 3.9).
• As the queues grow large, the router's memory can be exhausted and packet loss will occur.
• The location and extent of queueing will depend on

1) The traffic load

2) The relative speed of the switching fabric and

3) The line speed

• Switching fabric transfer rate R_{switch} is defined as

"The rate at which packets can be moved from input-port to output-port".

• If R_{switch} is N times faster than R_{line}, then only negligible queuing will occur at the input-ports.

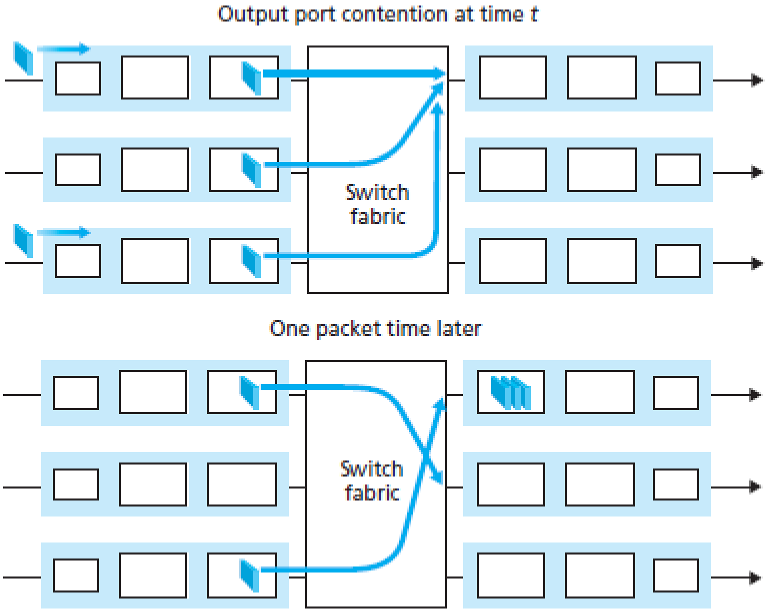


Figure 3.9: Output port queuing

- At output-port, packet-scheduler is used to choose one packet among those queued for transmission.
- The packet-scheduling can be done using
  → first-come-first-served (FCFS) or
  → weighted fair queuing (WFQ).
- Packet scheduling plays a crucial role in providing QoS guarantees.
- If there is less memory to hold an incoming-packet, a decision must be made to either
  1) Drop the arriving packet (a policy known as drop-tail) or
  2) Remove one or more already-queued packets to make room for the newly arrived packet.

3.4 IP: Forwarding & Addressing in the Internet

- IP(Internet Protocol) is main protocol responsible for packetizing, forwarding & delivery of a packet at network-layer.
- It is a connection-less & unreliable protocol.
  i) Connection-less means there is no connection setup b/w the sender and the receiver.
  ii) Unreliable protocol means
    → IP does not make any guarantee about delivery of the data.
    → Packets may get dropped during transmission.
- It provides a best-effort delivery service.
- Best effort means IP does its best to get the packet to its destination, but with no guarantees.
- If reliability is important, IP must be paired with a TCP which is reliable transport-layer protocol.
- IP does not provide following services
  → flow control
  → error control
  → congestion control services.

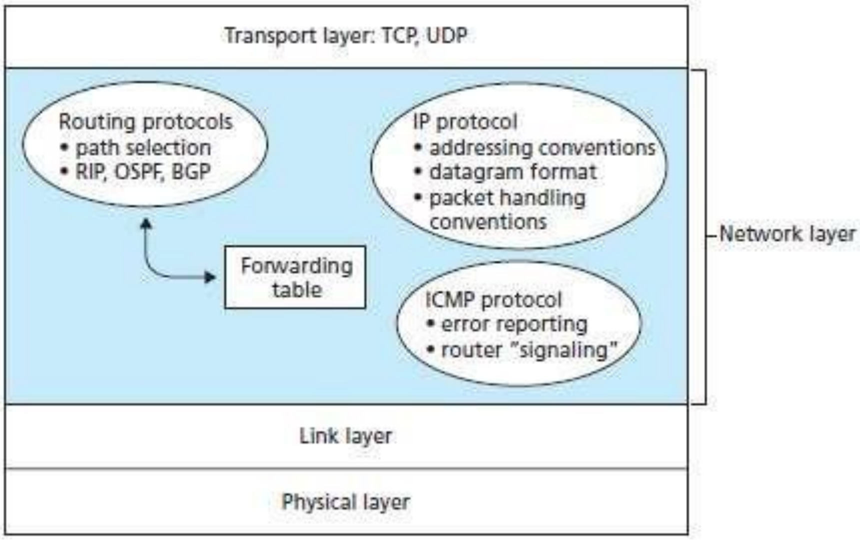


Figure 3.10: A look inside the Internet's network-layer

• Two important components of IP:
    1) Internet addressing and
    2) Forwarding
• There are two versions of IP in use today.
    1) IP version 4 (IPv4) and
    2) IP version 6 (IPv6)
• As shown in Figure 3.10, the network-layer has three major components:
    1) IP protocol
    2) Routing component determines the path a data follows from source to destination
    3) Network-layer is a facility to report errors in datagrams

3.4.1 IPv4 Datagram Format
• IP uses the packets called datagrams.
• A datagram consist of 2 parts: 1) Payload (or Data) 2) Header.

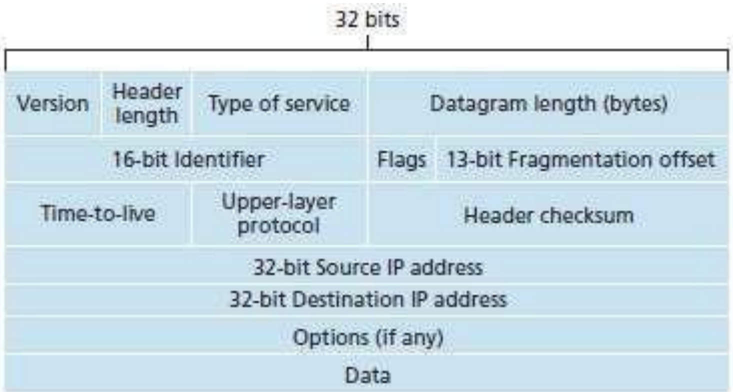


Figure 3.11: IPv4 datagram format

1) Payload (or Data)
• This field contains the data to be delivered to the destination.
2) Header
• Header contains information essential to routing and delivery.
• IP header contains following fields (Figure 3.11):
    1) Version
    ➢ This field specifies version of the IPv4 datagram, i.e. 4.
    2) Header Length
    ➢ This field specifies length of header.
    ➢ Without options field, header-length = 5 bytes.

3) Type of Service (TOS)

➢ This field specifies priority of packet based on parameters such as delay, throughput, reliability & cost.

4) Datagram Length

➢ This field specifies the total length of the datagram (header + data).

➢ Maximum length = 65535 bytes.

5) Identifier, Flags, Fragmentation Offset

➢ These fields are used for fragmentation and reassembly.

➢ Fragmentation occurs when the size of the datagram is larger than the MTU of the network.

i) Identifier: This field uniquely identifies a datagram packet.

ii) Flags: It is a 3-bit field. The first bit is not used.

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

The third bit M is called the more fragment bit.

iii) Fragmentation Offset: This field identifies location of a fragment in a datagram.

6) Time-To-Live (TTL)

➢ This defines lifetime of the datagram (default value 64) in hops.

➢ Each router decrements TTL by 1 before forwarding. If TTL is zero, the datagram is discarded.

7) Protocol

➢ This field specifies upper-layer protocol used to receive the datagram at the destination-host.

➢ For example, TCP=6 and UDP=17.

8) Header Checksum

➢ This field is used to verify integrity of header only.

➢ If the verification process fails, the packet is discarded.

9) Source IP Address & Destination IP Address

➢ These fields contain the addresses of source and destination respectively.

10) Options

➢ This field allows the packet to request special features such as

→ security level

→ route to be taken by packet at each router.

IPv6

- CIDR, subnetting and NAT could not solve address-space exhaustion faced by IPv4.
- IPv6 was evolved to solve this problem.

Changes from IPv4 to IPv6 (Advantages of IPv6)

1) Expanded Addressing Capabilities

➢ IPv6 increases the size of the IP address from 32 to 128 bits (Supports upto 3.4×10^{38} nodes).

➢ In addition to unicast & multicast addresses, IPv6 has an anycast address.

➢ Anycast address allows a datagram to be delivered to only one member of the group.

2) A Streamlined 40-byte Header

➢ A number of IPv4 fields have been dropped or made optional.

➢ The resulting 40-byte fixed-length header allows for faster processing of the IP datagram.

➢ A new encoding of options field allows for more flexible options processing.

3) Flow Labeling & Priority

➢ A flow can be defined as

"Labeling of packets belonging to particular flows for which the sender requests special handling".

➢ For example:

Audio and video transmission may be treated as a flow.

IPv6 Datagram Format

- The format of the IPv6 datagram is shown in Figure 3.18.

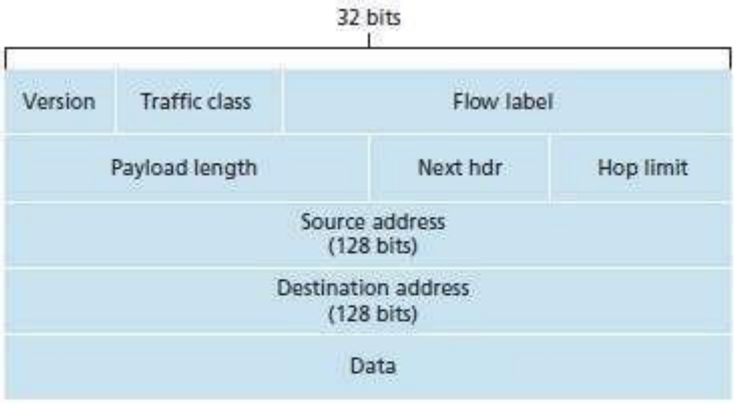


Figure 3.18: IPv6 datagram format

• The following fields are defined in IPv6:

4) Version

➢ This field specifies the IP version, i.e., 6.

5) Traffic Class

➢ This field is similar to the TOS field in IPv4.
➢ This field indicates the priority of the packet.

6) Flow Label

➢ This field is used to provide special handling for a particular flow of data.

7) Payload Length

➢ This field shows the length of the IPv6 payload.

8) Next Header

➢ This field is similar to the options field in IPv4 (Figure 3.19).
➢ This field identifies type of extension header that follows the basic header.

9) Hop Limit

➢ This field is similar to TTL field in IPv4.
➢ This field shows the maximum number of routers the packet can travel.
➢ The contents of this field are decremented by 1 by each router that forwards the datagram.
➢ If the hop limit count reaches 0, the datagram is discarded.

10) Source & Destination Addresses

➢ These fields show the addresses of the source & destination of the packet.

11) Data

➢ This field is the payload portion of the datagram.
➢ When the datagram reaches the destination, the payload will be
→ removed from the IP datagram and
→ passed on to the upper layer protocol (TCP or UDP).

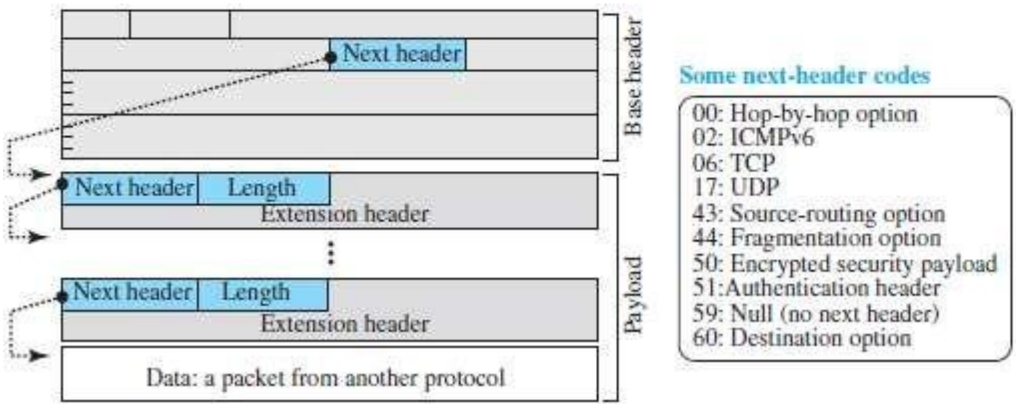


Figure 3.19: Payload in IPv6 datagram

3.4.4.2 IPv4 Fields not present in IPv6

1) Fragmentation/Reassembly

• Fragmentation of the packet is done only by the source, but not by the routers.

The reassembling is done by the destination.
• Fragmentation & reassembly is a time-consuming operation.
• At routers, the fragmentation is not allowed to speed up the processing in the router.
• If packet-size is greater than the MTU of the network, the router
→ drops the packet.
→ sends an error message to inform the source.

2) Header Checksum
• In the Internet layers, the transport-layer and link-layer protocols perform check summing.
• This functionality was redundant in the network-layer.
• So, this functionality was removed to speed up the processing in the router.

3) Options
• In, IPv6, next-header field is similar to the options field in IPv4.
• This field identifies type of extension header that follows the basic header.
• To support extra functionalities, extension headers can be placed b/w base header and payload.

3.4.4.3 Difference between IPv4 & IPv6

	IPv4	IPv6
1	IPv4 addresses are 32 bit length	IPv6 addresses are 128 bit length
2	Fragmentation is done by sender and forwarding routers	Fragmentation is done only by sender
3	Does not identify packet flow for QoS handling	Contains Flow Label field that specifies packet flow for QoS handling
4	Includes Options up to 40 bytes	Extension headers used for optional data
5	Includes a checksum	Does not includes a checksum
6	Address Resolution Protocol (ARP) is available to map IPv4 addresses to MAC addresses	Address Resolution Protocol (ARP) is replaced with Neighbor Discovery Protocol (NDP)
7	Broadcast messages are available	Broadcast messages are not available
8	Manual configuration (Static) of IP addresses or DHCP (Dynamic configuration) is required to configure IP addresses	Auto-configuration of addresses is available
9	IPSec is optional, external	IPSec is required

3.4.4.4 Transitioning from IPv4 to IPv6
• IPv4-capable systems are not capable of handling IPv6 datagrams.
• Two strategies have been devised for transition from IPv4 to IPv6:
1) Dual stack and
2) Tunneling.

3.4.5.5.1 Dual Stack Approach
• IPv6-capable nodes also have a complete IPv4 implementation. Such nodes are referred to as IPv6/IPv4 nodes.
• IPv6/IPv4 node has the ability to send and receive both IPv4 and IPv6 datagrams.
• When interoperating with an IPv4 node, an IPv6/IPv4 node can use IPv4 datagrams.
When interoperating with an IPv6 node, an IPv6/IPv4 node can use IPv6 datagrams.
• IPv6/IPv4 nodes must have both IPv6 and IPv4 addresses.
• IPv6/IPv4 nodes must be able to determine whether another node is IPv6-capable or IPv4-only.
• This problem can be solved using the DNS.
If the node name is resolved to IPv6-capable, then the DNS returns an IPv6 address
Otherwise, the DNS return an IPv4 address.
• If either the sender or the receiver is only IPv4-capable, an IPv4 datagram must be used.
• Two IPv6-capable nodes can send IPv4 datagrams to each other.

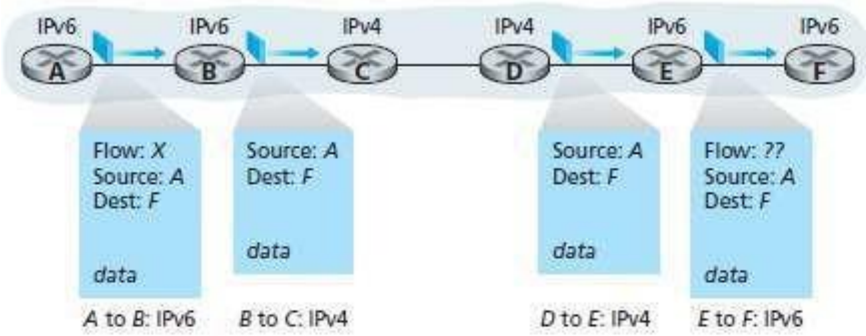


Figure 3.20: A dual-stack approach

- Dual stack is illustrated in Figure 3.20.
- Here is how it works:
  1) Suppose IPv6-capable Node-A wants to send a datagram to IPv6-capable Node-F.
  2) IPv6-capable Node-B creates an IPv4 datagram to send to IPv4-capable Node-C.
  3) At IPv6-capable Node-B, the IPv6 datagram is copied into the data field of the IPv4 datagram and appropriate address mapping can be done.
  4) At IPv6-capable Node-E, the IPv6 datagram is extracted from the data field of the IPv4 datagram.
  5) Finally, IPv6-capable Node-E forwards an IPv6 datagram to IPv6-capable Node-F.
- Disadvantage: During transition from IPv6 to IPv4, few IPv6-specific fields will be lost.

3.4.5.5.2 Tunneling

- Tunneling is illustrated in Figure 3.21.
- Suppose two IPv6-nodes B and E
  → want to interoperate using IPv6 datagrams and
  → are connected by intervening IPv4 routers.
- The intervening-set of IPv4 routers between two IPv6 routers are referred as a tunnel.
- Here is how it works:
  - On the sending side of the tunnel:
    → IPv6-node B takes & puts the IPv6 datagram in the data field of an IPv4 datagram.
    → The IPv4 datagram is addressed to the IPv6-node E.
  - On the receiving side of the tunnel: The IPv6-node E
    → receives the IPv4 datagram
    → extracts the IPv6 datagram from the data field of the IPv4 datagram and
    → routes the IPv6 datagram to IPv6-node F

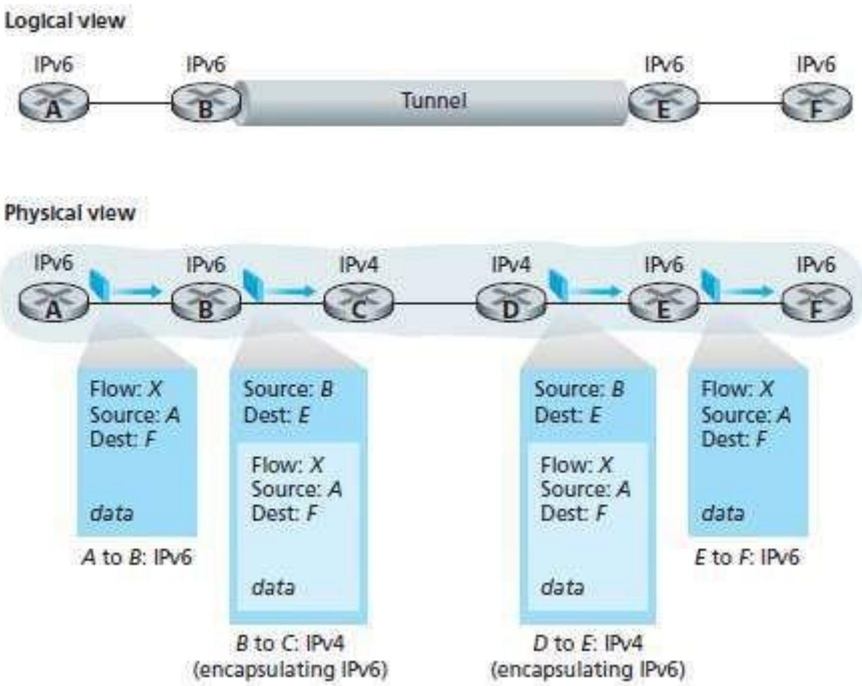


Figure 3.21: Tunneling

3.4.6 A Brief Foray into IP Security
- IPsec is a popular secure network-layer protocol.
- It is widely deployed in Virtual Private Networks (VPNs).
- It has been designed to be backward compatible with IPv4 and IPv6.
- It can be used to create a connection-oriented service between 2 entities.
- In transport mode, 2 hosts first establish an IPsec session between themselves.
- All TCP and UDP segments sent between the two hosts enjoy the security services provided by IPsec.
- On the source-side,
    1) The transport-layer passes a segment to IPsec.
    2) Then, IPsec
        → encrypts the segment
        → appends additional security fields to the segment and
        → encapsulates the resulting payload in a IP datagram.
    3) Finally, the sending-host sends the datagram into the Internet.
    ➢ The Internet then transports the datagram to the destination-host.
- On the destination-side,
    1) The destination receives the datagram from the Internet.
    2) Then, IPsec
        → decrypts the segment and
        → passes the unencrypted segment to the transport-layer.
- Three services provided by an IPsec:
    1) Cryptographic Agreement
    ➢ This mechanism allows 2 communicating hosts to agree on cryptographic algorithms & keys.
    2) Encryption of IP Datagram Payloads
    ➢ When the sender receives a segment from the transport-layer, IPsec encrypts the payload.
    ➢ The payload can only be decrypted by IPsec in the receiver.
    3) Data Integrity
    ➢ IPsec allows the receiver to verify that the datagram's header fields.
    ➢ The encrypted payload is not modified after transmission of the datagram into the n/w.
    4) Origin Authentication
    ➢ The receiver is assured that the source-address in datagram is the actual source of datagram.

3.5 Routing Algorithms
- A routing-algorithm is used to find a "good" path from source to destination.
- Typically, a good path is one that has the least cost.
- The least-cost problem: Find a path between the source and destination that has least cost.

3.5.1 Routing Algorithm Classification
- A routing-algorithm can be classified as follows:
    1) Global or decentralized
    2) Static or dynamic
    3) Load-sensitive or Load-insensitive

3.5.1.1 Global or Decentralized
Global Routing Algorithm
- The calculation of the least-cost path is carried out at one centralized site.
- This algorithm has complete, global knowledge about the network.
- Algorithms with global state information are referred to as link-state (LS) algorithms.

Decentralized Routing Algorithm
- The calculation of the least-cost path is carried out in an iterative, distributed manner.
- No node has complete information about the costs of all network links.
- Each node has only the knowledge of the costs of its own directly attached links.
- Each node performs calculation by exchanging information with its neighboring nodes.

3.5.1.2 Static or Dynamic
Static Routing Algorithms

• Routes change very slowly over time, as a result of human intervention.
• For example: a human manually editing a router's forwarding-table.

Dynamic Routing Algorithms

• The routing paths change, as the network-topology or traffic-loads change.
• The algorithm can be run either
  → periodically or
  → in response to topology or link cost changes.
• Advantage: More responsive to network changes.
• Disadvantage: More susceptible to routing loop problem.

3.5.1.3 Load Sensitive or Load Insensitive

Load Sensitive Algorithm

• Link costs vary dynamically to reflect the current level of congestion in the underlying link.
• If high cost is associated with congested-link, the algorithm chooses routes around congested-link.

Load Insensitive Algorithm

• Link costs do not explicitly reflect the current level of congestion in the underlying link.
• Today's Internet routing-algorithms are load-insensitive. For example: RIP, OSPF, and BGP

3.5.2 LS Routing Algorithm

3.5.2.1 Dijkstra's Algorithm

• Dijkstra's algorithm computes the least-cost path from one node to all other nodes in the network.
• Let us define the following notation:
  1) u: source-node
  2) D(v): cost of the least-cost path from the source u to destination v.
  3) p(v): previous node (neighbor of v) along the current least-cost path from the source to v.
  4) N': subset of nodes; v is in N' if the least-cost path from the source to v is known.

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

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

Loop
  find w not in N' such that D(w) is a minimum
  add w to N'
  update D(v) for each neighbor v of w and not in N':
        D(v) = min( D(v), D(w) + c(w,v) )
  /* new cost to v is either old cost to v or known
   least path cost to w plus cost from w to v */
until N'= N
```

• Example: Consider the network in Figure 3.22 and compute the least-cost paths from u to all possible destinations.

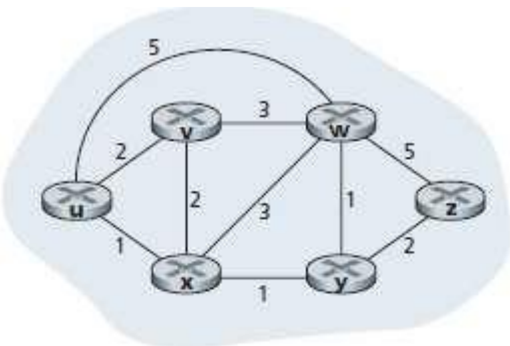


Figure 3.22: Abstract graph model of a computer network

Solution:
• Let's consider the few first steps in detail.

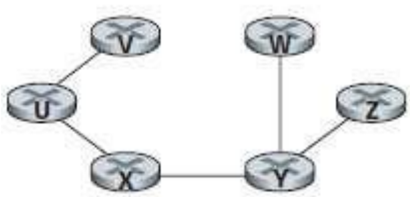
1) In the initialization step, the currently known least-cost paths from u to its directly attached neighbors, v, x, and w, are initialized to 2, 1, and 5, respectively.
2) In the first iteration, we
   → look among those nodes not yet added to the set N′ and
   → find that node with the least cost as of the end of the previous iteration.
3) In the second iteration,
   → nodes v and y are found to have the least-cost paths (2) and
   → we break the tie arbitrarily and
   → add y to the set N′ so that N′ now contains u, x, and y.
4) And so on. . . .
5) When the LS algorithm terminates,
   We have, for each node, its predecessor along the least-cost path from the source.

• A tabular summary of the algorithm's computation is shown in Table 3.5.

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

Table 3.5: Running the link-state algorithm on the network in Figure 3.20

• Figure 3.23 shows the resulting least-cost paths for u for the network in Figure 3.22.



Destination	Link
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)

Figure 3.23: Least cost path and forwarding-table for node u

3.5.3 DV Routing Algorithm
3.5.3.1 Bellman Ford Algorithm

• Distance vector (DV) algorithm is 1) iterative, 2) asynchronous, and 3) distributed.
1) It is distributed. This is because each node
   → receives some information from one or more of its directly attached neighbors
   → performs the calculation and
   → distributes then the results of the calculation back to the neighbors.
2) It is iterative. This is because
   → the process continues on until no more info is exchanged b/w neighbors.
3) It is asynchronous. This is because
   → the process does not require all of the nodes to operate in lockstep with each other.

• The basic idea is as follows:
1) Let us define the following notation:
   $D_x(y)$ = cost of the least-cost path from node x to node y, for all nodes in N.
   D_x = [$D_x(y)$: y in N] be node x's distance vector of cost estimates from x to all other nodes y in N.
2) Each node x maintains the following routing information:
   i) For each neighbor v, the cost c(x,v) from node x to directly attached neighbor v
   ii) Node x's distance vector, that is, D_x = [$D_x(y)$: y in N], containing x's estimate of its cost to all destinations y in N.

iii) The distance vectors of each of its neighbors, that is, $D_v = [D_v(y): y \text{ in } N]$ for each neighbor v of x.

3) From time to time, each node sends a copy of its distance vector to each of its neighbors.

4) The least costs are computed by the Bellman-Ford equation:

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

5) If node x's distance vector has changed as a result of this update step, node x will then send its updated distance vector to each of its neighbors.

Distance-Vector (DV) Algorithm

At each node, *x*:

```
Initialization:
   for all destinations y in N:
      Dx(y) = c(x,y)    /* if y is not a neighbor then c(x,y) = ∞ */
   for each neighbor w
      Dw(y) = ? for all destinations y in N
   for each neighbor w
      send distance vector Dx = [Dx(y): y in N] to w

loop
   wait (until I see a link cost change to some neighbor w or
         until I receive a distance vector from some neighbor w)

   for each y in N:
      Dx(y) = minv{c(x,v) + Dv(y)}

   if Dx(y) changed for any destination y
      send distance vector Dx = [Dx(y): y in N] to all neighbors

forever
```

• Figure 3.24 illustrates the operation of the DV algorithm for the simple three node network.

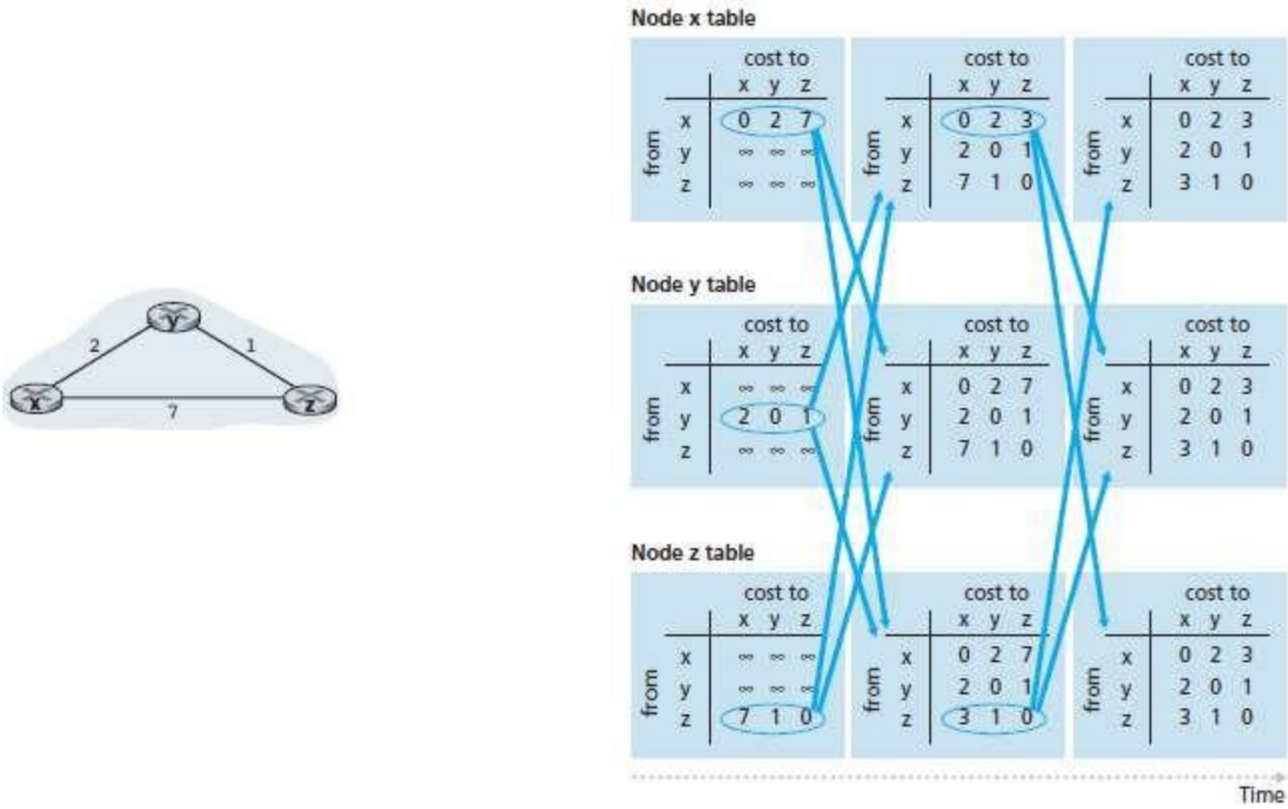


Figure 3.24: Distance-vector (DV) algorithm

• The operation of the algorithm is illustrated in a synchronous manner. Here, all nodes simultaneously

→ receive distance vectors from their neighbours

→ compute their new distance vectors, and
→ inform their neighbours if their distance vectors have changed.

- The table in the upper-left corner is node x's initial routing-table.
- In this routing-table, each row is a distance vector.
- The first row in node x's routing-table is $D_x = [D_x(x), D_x(y), D_x(z)] = [0, 2, 7]$.
- After initialization, each node sends its distance vector to each of its two neighbours.
- This is illustrated in Figure 3.24 by the arrows from the first column of tables to the second column of tables.
- For example, node x sends its distance vector Dx = [0, 2, 7] to both nodes y and z. After receiving the updates, each node recomputes its own distance vector.
- For example, node x computes

$D_x(x) = 0$
$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} = \min\{2 + 0, 7 + 1\} = 2$
$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} = \min\{2 + 1, 7 + 0\} = 3$

- The second column therefore displays, for each node, the node's new distance vector along with distance vectors just received from its neighbours.
- Note, that node x's estimate for the least cost to node z, $D_x(z)$, has changed from 7 to 3.
- The process of receiving updated distance vectors from neighbours, recomputing routing-table entries, and informing neighbours of changed costs of the least-cost path to a destination continues until no update messages are sent.
- The algorithm remains in the quiescent state until a link cost changes.

3.5.4 A Comparison of LS and DV Routing-algorithms

Distance Vector Protocol	Link State Protocol
Entire routing-table is sent as an update	Updates are incremental & entire routing-table is not sent as update
Distance vector protocol send periodic update at every 30 or 90 second	Updates are triggered not periodic
Updates are broadcasted	Updates are multicasted
Updates are sent to directly connected neighbour only	Update are sent to entire network & to just directly connected neighbour
Routers don't have end to end visibility of entire network.	Routers have visibility of entire network of that area only.
Prone to routing loops	No routing loops
Each node talks to only its directly connected neighbors	Each node talks with all other nodes (via broadcast)

3.5.5 Hierarchical Routing

- Two problems of a simple routing-algorithm:

1) Scalability

➢ As no. of routers increases, overhead involved in computing & storing routing info increases.

2) Administrative Autonomy

➢ An organization should be able to run and administer its network.
➢ At the same time, the organization should be able to connect its network to internet.

- Both of these 2 problems can be solved by organizing routers into autonomous-system (AS).
- An autonomous system (AS) is a group of routers under the authority of a single administration.
  For example: same ISP or same company network.
- Two types of routing-protocol:
  1) Intra-AS routing protocol: refers to routing inside an autonomous system.
  2) Inter-AS routing protocol: refers to routing between autonomous systems.

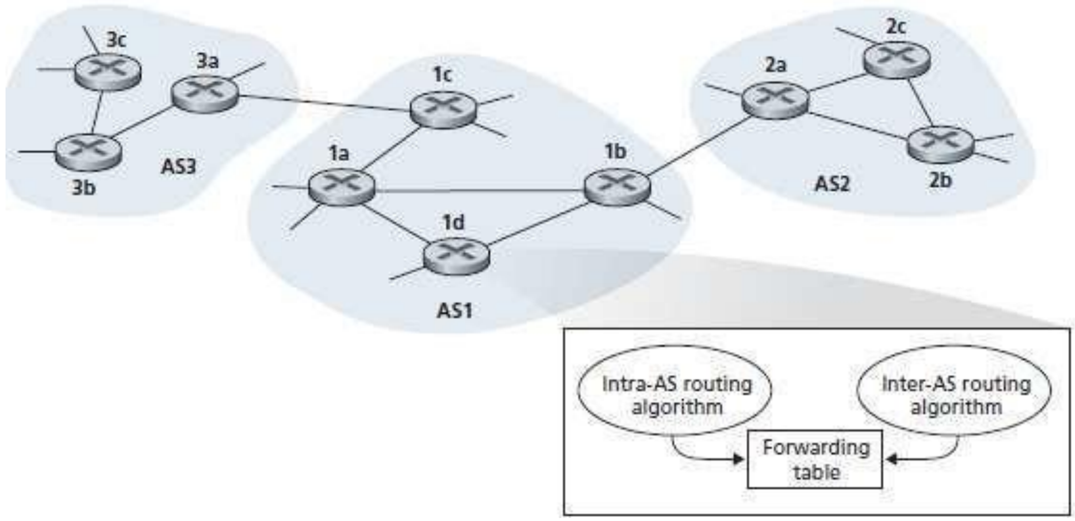


Figure 3.25: An example of interconnected autonomous-systems

3.5.5.1 Intra-AS Routing Protocol

- The routing-algorithm running within an autonomous-system is called intra-AS routing protocol.
- All routers within the same AS must run the same intra-AS routing protocol. For ex: RIP and OSPF
- Figure 3.25 provides a simple example with three ASs: AS1, AS2, and AS3.
- AS1 has four routers: 1a, 1b, 1c, and 1d. These four routers run the intra-AS routing protocol.
- Each router knows how to forward packets along the optimal path to any destination within AS1.

3.5.5.2 Inter-AS Routing Protocol

- The routing-algorithm running between 2 autonomous-systems is called inter-AS routing protocol.
- Gateway-routers are used to connect ASs to each other.
- Gateway-routers are responsible for forwarding packets to destinations outside the AS.
- Two main tasks of inter-AS routing protocol:
    1) Obtaining reachability information from neighboring Ass.
    2) Propagating the reachability information to all routers internal to the AS.
- The 2 communicating ASs must run the same inter-AS routing protocol. For ex: BGP.
- Figure 3.26 summarizes the steps in adding an outside-AS destination in a router's forwarding-table.

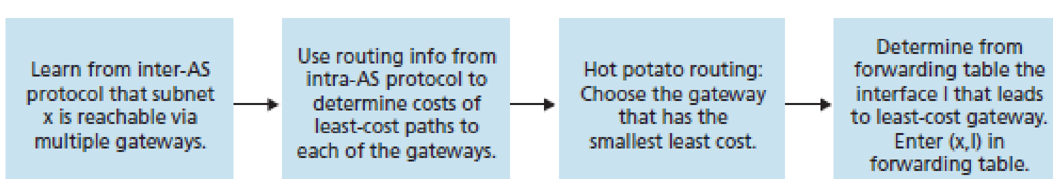


Figure 3.26: Steps in adding an outside-AS destination in a router's forwarding-table

3.6 Routing in the Internet

- Purpose of Routing protocols:

  To determine the path taken by a datagram between source and destination.
- An autonomous-system (AS) is a collection of routers under the same administrative control.
- In AS, all routers run the same routing protocol among themselves.

3.6.1 Intra-AS Routing in the Internet: RIP

- Intra-AS routing protocols are also known as interior gateway protocols.
- An intra-AS routing protocol is used to determine how routing is performed within an AS.
- Most common intra-AS routing protocols:

  1) Routing-information Protocol (RIP) and 2) Open Shortest Path First (OSPF)
- OSPF deployed in upper-tier ISPs whereas RIP is deployed in lower-tier ISPs & enterprise-networks.

3.6.1.1 RIP Protocol

- RIP is widely used for intra-AS routing in the Internet.
- RIP is a distance-vector protocol.
- RIP uses hop count as a cost metric. Each link has a cost of 1.
- Hop count refers to the no. of subnets traversed along the shortest path from source to destination.
- The maximum cost of a path is limited to 15.
- The distance vector is the current estimate of shortest path distances from router to subnets in AS.
- Consider an AS shown in Figure 3.27.

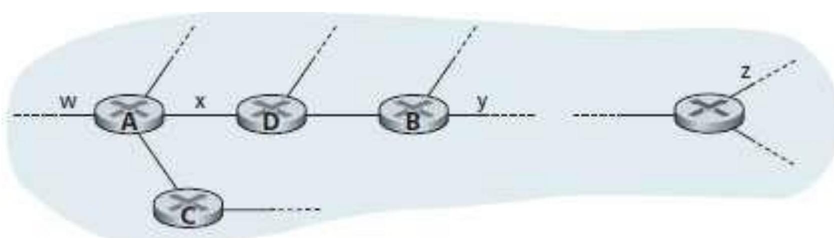


Figure 3.27: A portion of an autonomous-system

- Each router maintains a RIP table known as a routing-table.
- Figure 3.28 shows the routing-table for router D.

Destination Subnet	Next Router	Number of Hops to Destination
w	A	2
y	B	2
z	B	7
x	—	1

Figure 3.28: Routing-table in router D before receiving advertisement from router A

- Routers can send types of messages: 1) Response-message & 2) Request-message

  1) Response Message

  - Using this message, the routers exchange routing updates with their neighbors every 30 secs.
  - If a router doesn't hear from its neighbor every 180 secs, then that neighbor is not reachable.
  - When this happens, RIP

→ modifies the local routing-table and
→ propagates then this information by sending advertisements to its neighbors.

- The response-message contains
  → list of up to 25 destination subnets within the AS and
  → sender's distance to each of those subnets.
- Response-messages are also known as advertisements.

2) Request Message

- Using this message, router requests info about its neighbor's cost to a given destination.

- Both types of messages are sent over UDP using port# 520.
- The UDP segment is carried between routers in an IP datagram.

3.6.2 Intra-AS Routing in the Internet: OSPF

- OSPF is widely used for intra-AS routing in the Internet.
- OSPF is a link-state protocol that uses
  → flooding of link-state information and
  → Dijkstra least-cost path algorithm.
- Here is how it works:
  1) A router constructs a complete topological map (a graph) of the entire autonomous-system.
  2) Then, the router runs Dijkstra's algorithm to determine a shortest-path tree to all subnets.
  3) Finally, the router broadcasts link state info to all other routers in the autonomous-system. Specifically, the router broadcasts link state information
     → periodically at least once every 30 minutes and
     → whenever there is a change in a link's state. For ex: a change in up/down status.
- Individual link costs are configured by the network-administrator.
- OSPF advertisements are contained in OSPF messages that are carried directly by IP.
- HELLO message can be used to check whether the links are operational.
- The router can also obtain a neighboring router's database of network-wide link state.
- Some of the advanced features include:

1) Security

- Exchanges between OSPF routers can be authenticated.
- With authentication, only trusted routers can participate within an AS.
- By default, OSPF packets between routers are not authenticated.
- Two types of authentication can be configured: 1) Simple and 2) MD5.

i) Simple Authentication

¤ The same password is configured on each router.
¤ Clearly, simple authentication is not very secure.

ii) MD5 Authentication

¤ This is based on shared secret keys that are configured in all the routers.
¤ Here is how it works:

1) The sending router
   → computes a MD5 hash on the content of packet
   → includes the resulting hash value in the packet and
   → sends the packet
2) The receiving router
   → computes an MD5 hash of the packet
   → compares computed-hash value with the hash value carried in packet and

→ verifies the packet's authenticity

2) Multiple Same Cost Paths

- When multiple paths to a destination have same cost, OSPF allows multiple paths to be used.

3) Integrated Support for Unicast & Multicast Routing

- Multicast OSPF (MOSPF) provides simple extensions to OSPF to provide for multicast-routing.
- MOSPF
  → uses the existing OSPF link database and
  → adds a new type of link-state advertisement to the existing broadcast mechanism.

4) Support for Hierarchy within a Single Routing Domain

- An autonomous-system can be configured hierarchically into areas.
- In area, an area-border-router is responsible for routing packets outside the area.
- Exactly one OSPF area in the AS is configured to be the backbone-area.
- The primary role of the backbone-area is to route traffic between the other areas in the AS.

3.6.3 Inter-AS Routing: BGP

- BGP is widely used for inter-AS routing in the Internet.
- Using BGP, each AS can
  1) Obtain subnet reachability-information from neighboring ASs.
  2) Propagate the reachability-information to all routers internal to the AS.
  3) Determine good routes to subnets based on i) reachability-information and ii) AS policy.
- Using BGP, each subnet can advertise its existence to the rest of the Internet.

3.6.3.1 Basics

- Pairs of routers exchange routing-information over semi-permanent TCP connections using port-179.
- One TCP connection is used to connect 2 routers in 2 different autonomous-systems. Semipermanent TCP connection is used to connect among routers within an autonomous-system.
- Two routers at the end of each connection are called peers.
  The messages sent over the connection is called a session.
- Two types of session:
  1) External BGP (eBGP) session
  - This refers to a session that spans 2 autonomous-systems.
  2) Internal BGP (iBGP) session
  - This refers to a session between routers in the same AS.

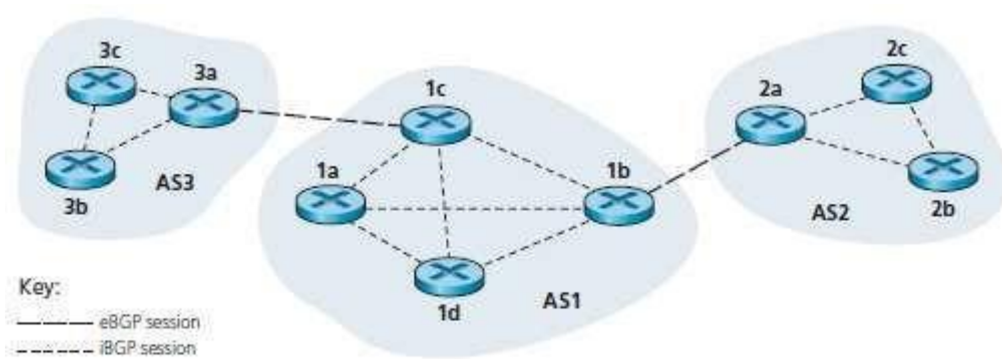


Figure 3.29: eBGP and iBGP sessions

- BGP operation is shown in Figure 3.29.
- The destinations are not hosts but instead are CIDRized prefixes.
- Each prefix represents a subnet or a collection of subnets.

3.6.3.2 Path Attributes & Routes

- An autonomous-system is identified by its globally unique ASN (Autonomous-System Number).
- A router advertises a prefix across a session.
- The router includes a number of attributes with the prefix.
- Two important attributes: 1) AS-PATH and 2) NEXT-HOP

  1) AS-PATH
  - ➢ This attribute contains the ASs through which the advertisement for the prefix has passed.
  - ➢ When a prefix is passed into an AS, the AS adds its ASN to the ASPATH attribute.
  - ➢ Routers use the AS-PATH attribute to detect and prevent looping advertisements.
  - ➢ Routers also use the AS-PATH attribute in choosing among multiple paths to the same prefix.

  2) NEXT-HOP
  - ➢ This attribute provides the critical link between the inter-AS and intra-AS routing protocols.
  - ➢ This attribute is the router-interface that begins the AS-PATH.
- BGP also includes
  - → attributes which allow routers to assign preference-metrics to the routes.
  - → attributes which indicate how the prefix was inserted into BGP at the origin AS.
- When a gateway-router receives a route-advertisement, the gateway-router decides
  - → whether to accept or filter the route and
  - → whether to set certain attributes such as the router preference metrics.

3.6.3.3 Route Selection

- For 2 or more routes to the same prefix, the following elimination-rules are invoked sequentially:
  1) Routes are assigned a local preference value as one of their attributes.
  2) The local preference of a route
     - → will be set by the router or
     - → will be learned by another router in the same AS.
  3) From the remaining routes, the route with the shortest AS-PATH is selected.
  4) From the remaining routes, the route with the closest NEXT-HOP router is selected.
  5) If more than one route still remains, the router uses BGP identifiers to select the route.

3.6.3.4 Routing Policy

- Routing policy is illustrated as shown in Figure 3.30.
- Let A, B, C, W, X & Y = six interconnected autonomous-systems. W, X & Y = three stub-networks.
  A, B & C = three backbone provider networks.
- All traffic entering a stub-network must be destined for that network.
  All traffic leaving a stub-network must have originated in that network.
- Clearly, W and Y are stub-networks.

• X is a multihomed stub-network, since X is connected to the rest of the n/w via 2 different providers
• X itself must be the source/destination of all traffic leaving/entering X.
• X will function as a stub-network if X has no paths to other destinations except itself.
• There are currently no official standards that govern how backbone ISPs route among themselves.

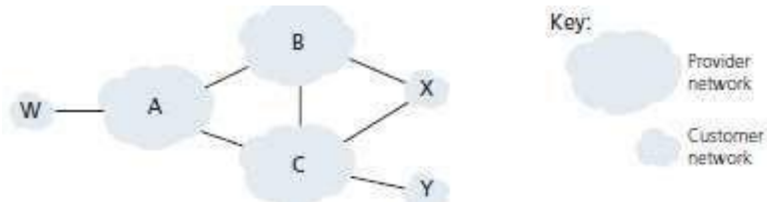


Figure 3.30: A simple BGP scenario

3.7 Broadcast & Multicast Routing
3.7.1 Broadcast Routing Algorithms
• Broadcast-routing means delivering a packet from a source-node to all other nodes in the network.

3.7.1.1 N-way Unicast
• Given N destination-nodes, the source-node
→ makes N copies of the packet and
→ transmits then the N copies to the N destinations using unicast routing (Figure 3.31).
• Disadvantages:
1) Inefficiency
➢ If source is connected to the n/w via single link, then N copies of packet will traverse this link.
2) More Overhead & Complexity
➢ An implicit assumption is that the sender knows broadcast recipients and their addresses.
➢ Obtaining this information adds more overhead and additional complexity to a protocol.
3) Not suitable for Unicast Routing
➢ It is not good idea to depend on the unicast routing infrastructure to achieve broadcast.

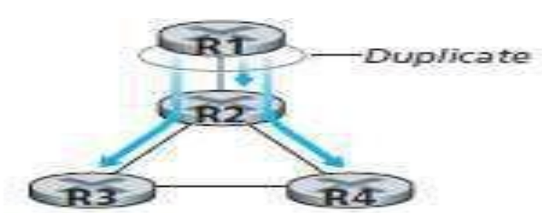


Figure 3.31: Duplicate creation/transmission

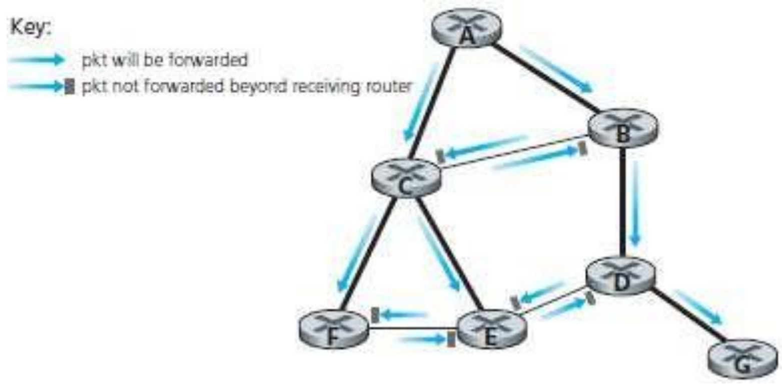


Figure 3.32: Reverse path forwarding

3.7.1.2 Uncontrolled Flooding

- The source-node sends a copy of the packet to all the neighbors.
- When a node receives a broadcast-packet, the node duplicates & forwards packet to all neighbors.
- In connected-graph, a copy of the broadcast-packet is delivered to all nodes in the graph.
- Disadvantages:
  1) If the graph has cycles, then copies of each broadcast-packet will cycle indefinitely.
  2) When a node is connected to 2 other nodes, the node creates & forwards multiple copies of packet
- Broadcast-storm refers to

  "The endless multiplication of broadcast-packets which will eventually make the network useless."

3.7.1.3 Controlled Flooding

- A node can avoid a broadcast-storm by judiciously choosing

  → when to flood a packet and when not to flood a packet.
- Two methods for controlled flooding:

  1) Sequence Number Controlled Flooding

  ➢ A source-node

  → puts its address as well as a broadcast sequence-number into a broadcast-packet

  → sends then the packet to all neighbors.

  ➢ Each node maintains a list of the source-address & sequence# of each broadcast-packet.

  ➢ When a node receives a broadcast-packet, the node checks whether the packet is in this list.

  ➢ If so, the packet is dropped; if not, the packet is duplicated and forwarded to all neighbors.

  2) Reverse Path Forwarding (RPF)

  ➢ If a packet arrived on the link that has a path back to the source; Then the router transmits the packet on all outgoing-links.

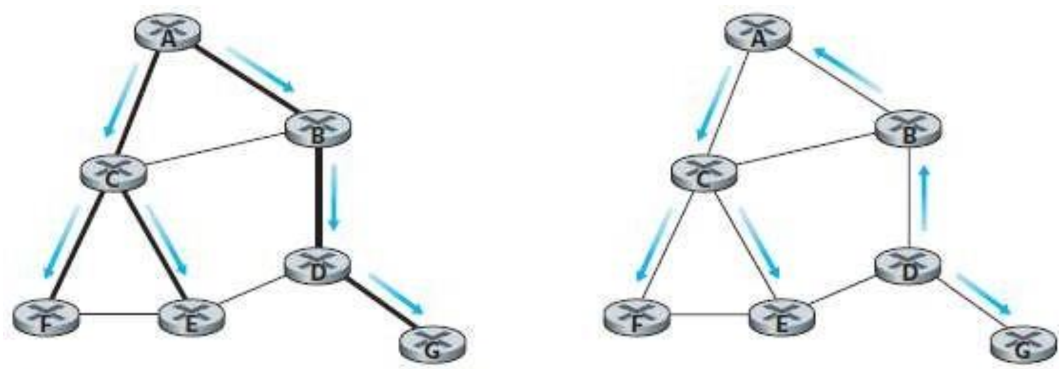
  Otherwise, the router discards the incoming-packet.

  ➢ Such a packet will be dropped. This is because

  → the router has already received a copy of this packet (Figure 3.32).

3.7.1.4 Spanning - Tree Broadcast

- This is another approach to providing broadcast. (MST → Minimum Spanning Tree).
- Spanning-tree is a tree that contains each and every node in a graph.
- A spanning-tree whose cost is the minimum of all of the graph's spanning-trees is called a MST.
- Here is how it works (Figure 3.33):
  1) Firstly, the nodes construct a spanning-tree.
  2) The node sends broadcast-packet out on all incident links that belong to the spanning-tree.
  3) The receiving-node forwards the broadcast-packet to all neighbors in the spanning-tree.



a. Broadcast initiated at A   b. Broadcast initiated at D

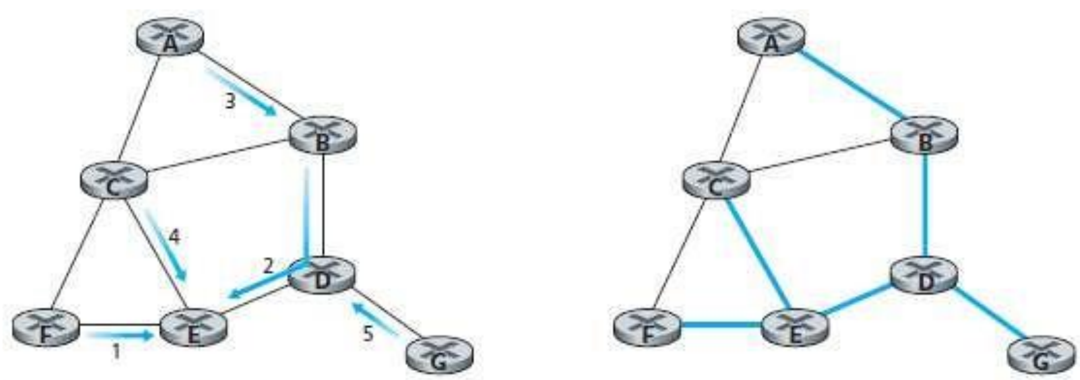
Figure 3.33: Broadcast along a spanning-tree

- Disadvantage:

  Complex: The main complexity is the creation and maintenance of the spanning-tree.

3.7.1.4.1 Center Based Approach

- This is a method used for building a spanning-tree.
- Here is how it works:
  1) A center-node (rendezvous point or a core) is defined.
  2) Then, the nodes send unicast tree-join messages to the center-node.
  3) Finally, a tree-join message is forwarded toward the center until the message either
     - → arrives at a node that already belongs to the spanning-tree or
     - → arrives at the center.
- Figure 3.34 illustrates the construction of a center-based spanning-tree.



a. Stepwise construction of spanning tree   b. Constructed spanning tree

Figure 3.34: Center-based construction of a spanning-tree

3.7.2 Multicast

- Multicasting means a multicast-packet is delivered to only a subset of network-nodes.
- A number of emerging network applications requires multicasting. These applications include
    1) Bulk data transfer (for ex: the transfer of a software upgrade)
    2) Streaming continuous media (for ex: the transfer of the audio/video)
    3) Shared data applications (for ex: a teleconferencing application)
    4) Data feeds (for ex: stock quotes)
    5) Web cache updating and
    6) Interactive gaming (for ex: multiplayer games).
- Two problems in multicast communication:
    1) How to identify the receivers of a multicast-packet.
    2) How to address a packet sent to these receivers.
- A multicast-packet is addressed using address indirection.
- A single identifier is used for the group of receivers.
- Using this single identifier, a copy of the packet is delivered to all multicast receivers.
- In the Internet, class-D IP address is the single identifier used to represent a group of receivers.
- The multicast-group abstraction is illstrated in Figure 3.35.

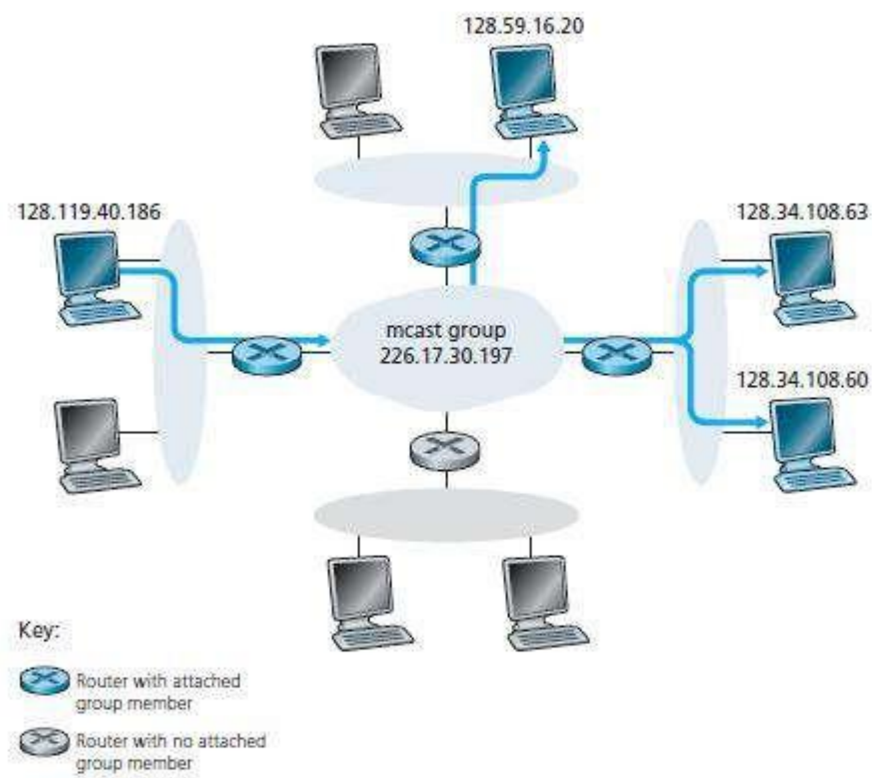


Figure 3.35: The multicast group: A datagram addressed to the group is delivered to all members of the multicast group

3.7.2.1 IGMP

- In the Internet, the multicast consists of 2 components:
    1) IGMP (Internet Group Management Protocol)

➢ IGMP is a protocol that manages group membership.
➢ It provides multicast-routers info about the membership-status of hosts connected to the n/w
➢ The operations are i) Joining/Leaving a group and ii) monitoring membership

2) Multicast Routing Protocols

➢ These protocols are used to coordinate the multicast-routers throughout the Internet.
➢ A host places a multicast address in the destination address field to send packets to a set of hosts belonging to a group.

• The IGMP protocol operates between a host and its attached-router.
• Figure 3.36 shows three first-hop multicast-routers.

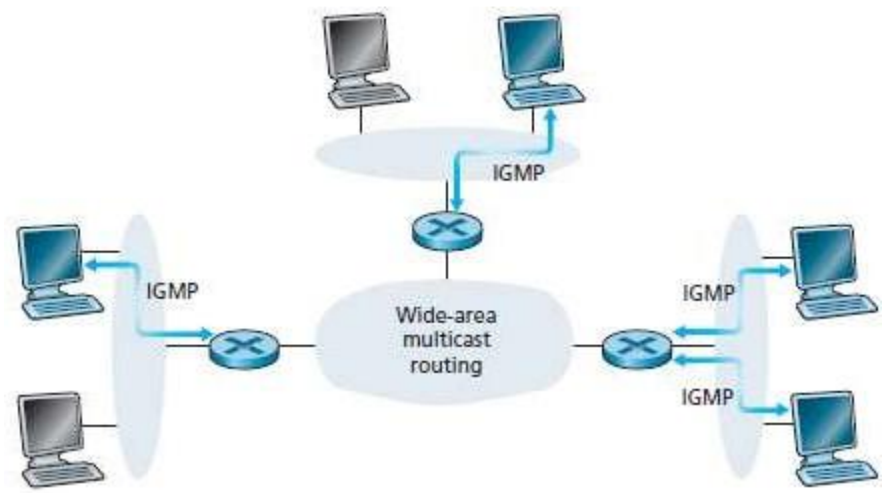


Figure 3.36: The two components of network-layer multicast in the Internet: IGMP and multicast- routing protocols

• IGMP messages are encapsulated within an IP datagram.
• Three types of message: 1) membership_query 2) membership_report 3) leave_group

1) membership_query

➢ A host sends a membership-query message to find active group-members in the network.

2) membership_report

➢ A host sends membership_report message when an application first joins a multicast-group.
➢ The host sends this message w/o waiting for a membership_query message from the router.

3) leave_group

➢ This message is optional.
➢ The host sends this message to leave the multicast-group.

• How does a router detect when a host leaves the multicast-group?
Answer: The router infers that a host is no longer in the multicast-group if it no longer responds to a membership_query message. This is called soft state.

3.7.2.2 Multicast Routing Algorithms

• The multicast-routing problem is illustrated in Figure 3.37.
• Two methods used for building a multicast-routing tree:
    1) Single group-shared tree.
    2) Source-specific routing tree.

1) Multicast Routing using a Group Shared Tree

- A single group-shared tree is used to distribute the traffic for all senders in the group.
- This is based on
  Building a tree that includes all edge-routers & attached-hosts belonging to the multicast-group.
- In practice, a center-based approach is used to construct the multicast-routing tree.
- Edge-routers send join messages addressed to the center-node.
- Here is how it works:
  1) A center-node (rendezvous point or a core) is defined.
  2) Then, the edge-routers send unicast tree-join messages to the center-node.
  3) Finally, a tree-join message is forwarded toward the center until it either
     → arrives at a node that already belongs to the multicast tree or
     → arrives at the center.

2) Multicast Routing using a Source Based Tree

- A source-specific routing tree is constructed for each individual sender in the group.
- In practice, an RPF algorithm is used to construct a multicast forwarding tree.
- The solution to the problem of receiving unwanted multicast-packets under RPF is known as pruning.
- A multicast-router that has no attached-hosts will send a prune message to its upstream router.

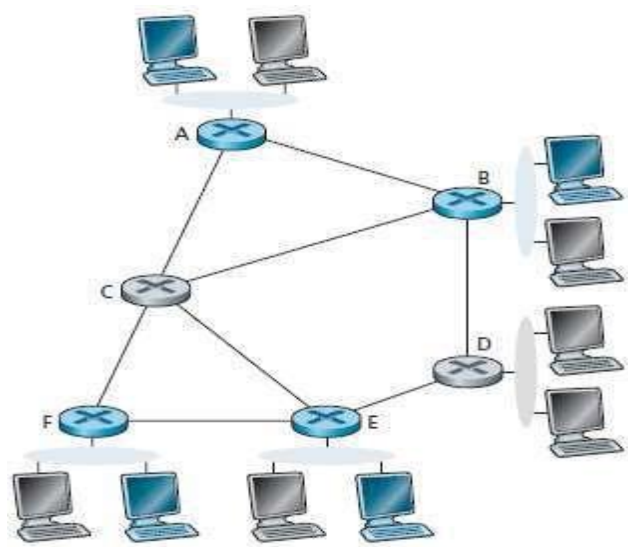


Figure 3.37: Multicast hosts, their attached routers, and other routers

3.7.2.3 Multicast Routing in the Internet

- Three multicast routing protocols are:
  1) Distance Vector Multicast Routing Protocol (DVMRP)
  2) Protocol Independent Multicast (PIM) and
  3) Source Specific Multicast (SSM)

1) DVMRP

- DVMRP was the first multicast-routing protocol used in the Internet.
- DVMRP uses an RPF algorithm with pruning. (Reverse Path Forwarding).

2) PIM

- PIM is the most widely used multicast-routing protocol in the Internet.
- PIM divides multicast routing into sparse and dense mode.

  i) Dense Mode

  - Group-members are densely located.

- Most of the routers in the area need to be involved in routing the data.
- PIM dense mode is a flood-and-prune reverse path forwarding technique.

i) Sparse Mode

- The no. of routers with attached group-members is small with respect to total no. of routers.
- Group-members are widely dispersed.
- This uses rendezvous points to set up the multicast distribution tree.

3) SSM

• Only a single sender is allowed to send traffic into the multicast tree. This simplifies tree construction & maintenance.