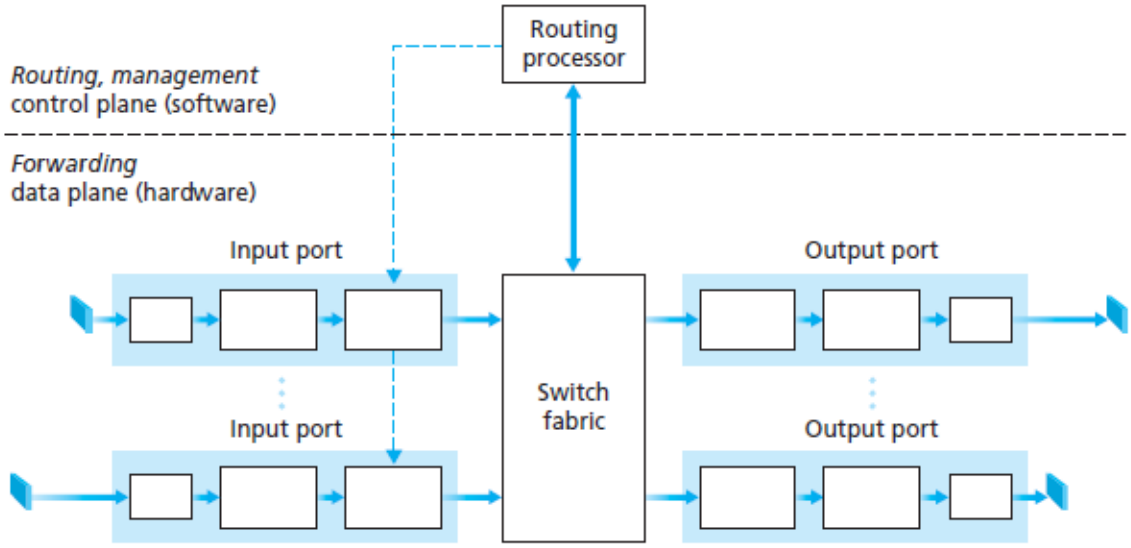
Module – 3

NETWORK LAYER

Structure of a Router

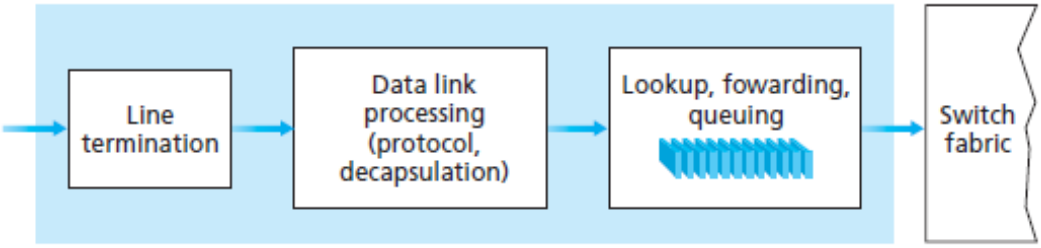
A high-level view of generic router architecture is shown below. Four router components can be identified:



- **Input ports:** An input port performs several key functions.
  It performs the physical layer function of terminating an incoming physical link at a router.
  An input port also performs link-layer functions needed to interoperate with the link layer at the other side of the incoming link.
  The lookup function is also performed at the input port.
  Forwarding table is consulted to determine the router output port to which an arriving packet will be forwarded via the switching fabric.
- **Switching fabric:** The switching fabric connects the router's input ports to its output ports.
- **Output port:** An Output port stores packet received from the switching fabric and transmits these packets on the outgoing link by performing the necessary link-layer and physical-layer functions.
- **Routing processor:** The routing processor executes the routing protocols maintains routing tables and attached link state information, and computes the forwarding table for the router. It also performs the network management functions.

- A router's input ports, output ports, and switching fabric together implement the forwarding function and are almost always implemented in hardware. These forwarding functions are sometimes collectively referred to as the **router forwarding plane**.
- **Router control plane** functions are usually implemented in software and execute on the routing processor

Input Processing

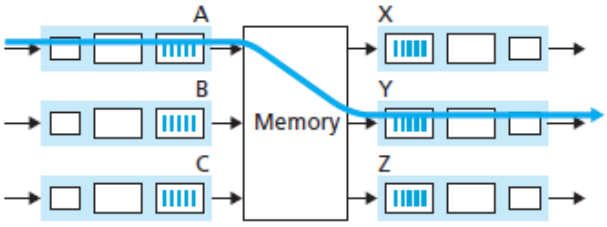


- The input port's line termination function and link-layer processing implement the physical and link layers for that individual input link.
- The lookup performed in the input port is central to the router's operation—it is here that the router uses the forwarding table to look up the output port to which an arriving packet will be forwarded via the switching fabric.
- The forwarding table is computed and updated by the routing processor, with a shadow copy typically stored at each input port. The forwarding table is copied from the routing processor to the line cards over a separate bus.
- Once a packet's output port has been determined via the lookup, the packet can be sent into the switching fabric. In some designs, a packet may be temporarily blocked from entering the switching fabric if packets from other input ports are currently using the fabric. A blocked packet will be queued at the input port and then scheduled to cross the fabric at a later point in time.

Switching

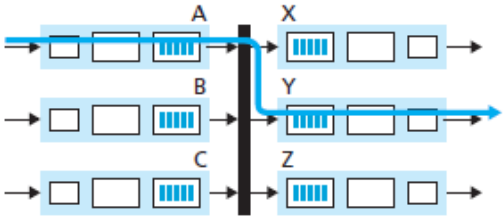
The switching fabric switches the packet from an input port to an output port. Switching can be accomplished in a number of ways:

1. **Switching via memory:**
   - The simplest, earliest routers were traditional computers, with switching between input and output ports being done under direct control of the CPU (routing processor).
   - Input and output ports functioned as traditional I/O devices in a traditional operating system.
   - An input port with an arriving packet first signaled the routing processor via an interrupt.
   - The packet was then copied from the input port into processor memory.
   - The routing processor then extracted the destination address from the header, looked up the appropriate output port in the forwarding table, and copied the packet to the output port's buffers.
   - Here two packets cannot be forwarded at the same time, even if they have different destination ports, since only one memory read/write over the shared system bus can be done at a time.
   - Many modern routers switch via memory. A major difference from early routers is that the lookup of the destination address and the storing of the packet into the appropriate memory location are performed by processing on the input line cards.



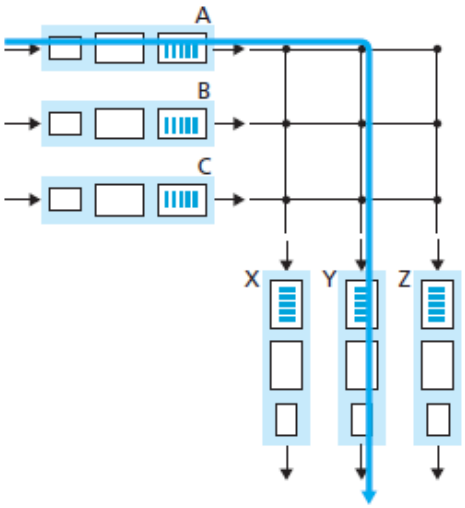
2. **Switching via a bus:**
   - In this approach, an input port transfers a packet directly to the output port over a shared bus, without intervention by the routing processor.
   - This is typically done by having the input port pre-pend a switch-internal label (header) to the packet indicating the local output port to which this packet is being transferred and transmitting the packet onto the bus.
   - The packet is received by all output ports, but only the port that matches the label will keep the packet.
   - The label is then removed at the output port, as this label is only used within the switch to cross the bus.

- If multiple packets arrive to the router at the same time, each at a different input port, all but one must wait since only one packet can cross the bus at a time.



3. Switching via an interconnection network:

- One way to overcome the bandwidth limitation of a single, shared bus is to use a more sophisticated interconnection network, such as those that have been used in the past to interconnect processors in multiprocessor computer architecture.
- A crossbar switch is an interconnection network consisting of 2N buses that connect N input ports to N output ports.
- Each vertical bus intersects each horizontal bus at a crosspoint, which can be opened or closed at any time by the switch fabric.

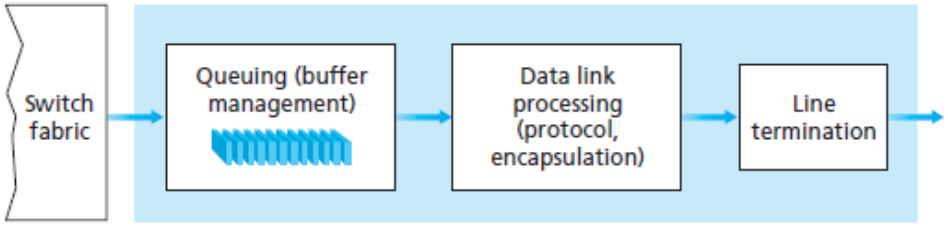


When a packet arrives from port A and needs to be forwarded to port Y, the switch controller closes the crosspoint at the intersection of busses A and Y, and port A then sends the packet onto its bus, which is picked up (only) by bus Y. Note that a packet from port B can be forwarded to port X at the same time, since the A-to-Y and B-to-X packets use different input and output busses. Thus, unlike the previous two switching approaches, crossbar networks are capable of forwarding multiple packets in parallel. However, if two packets

from two different input ports are destined to the same output port, then one will have to wait at the input, since only one packet can be sent over any given bus at a time.

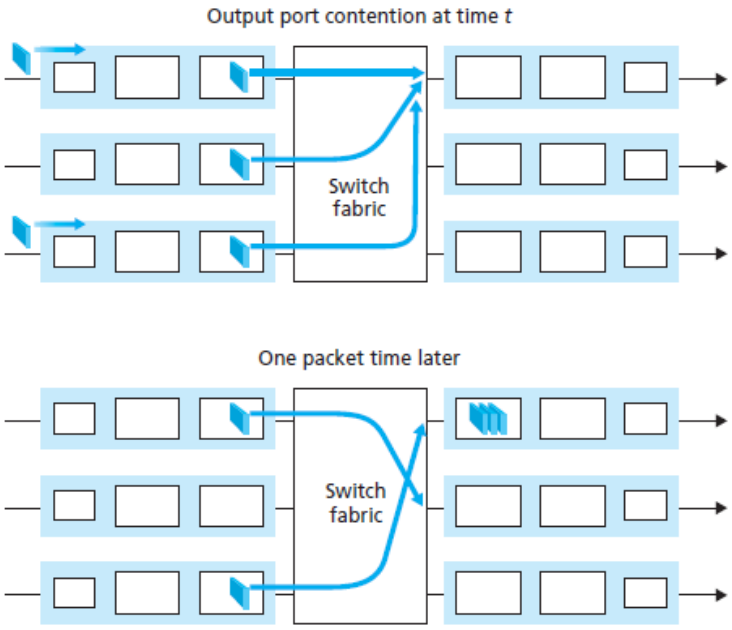
Output Processing

Output port processing takes packets that have been stored in the output port's memory and transmits them over the output link. This includes selecting and de-queuing packets for transmission, and performing the needed link layer and physical-layer transmission functions.



Queuing

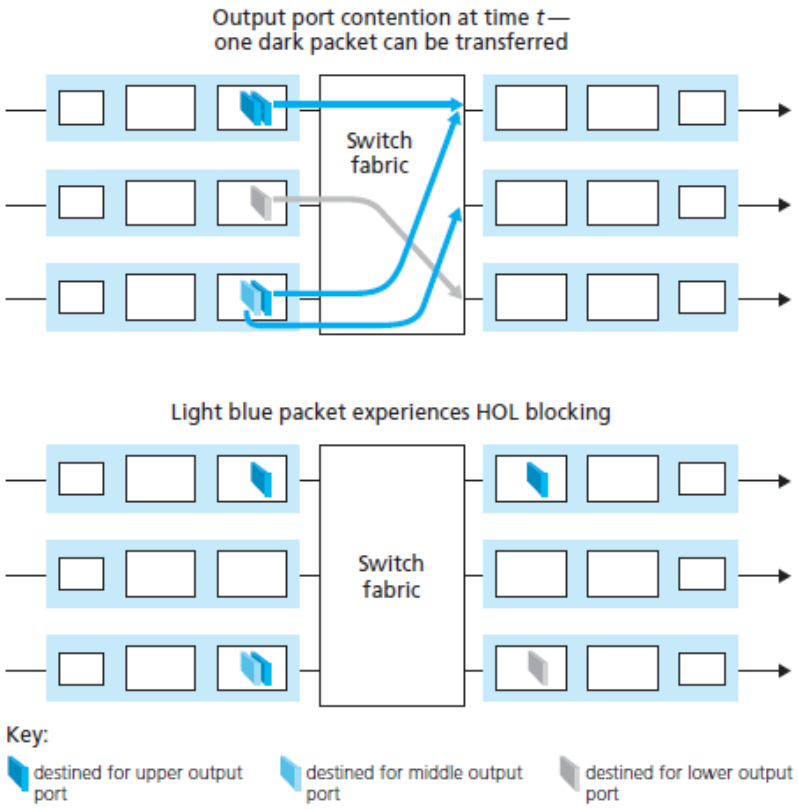
- Packet queues may form at both the input ports and the output ports.
- The location and extent of queuing will depend on the traffic load, the relative speed of the switching fabric, and the line speed.
- When many packets arrive from same source at a faster rate than switching rate queuing at input port occurs.
- When many packets are destined towards same output port queuing at output port occurs.



- A consequence of output port queuing is that a **packet scheduler** at the output port must choose one packet among those queued for transmission.
- Many packet scheduling algorithms like first-come-first-served (FCFS) scheduling, priority queuing, fair queuing or a more sophisticated scheduling discipline such as weighted fair queuing (WFQ) is available.
- Packet scheduling plays a crucial role in providing quality-of-service guarantees.
- Similarly, if there is not enough memory to buffer an incoming packet, a decision must be made to either drop the arriving packet (a policy known as **drop-tail**) or remove one or more already-queued packets to make room for the newly arrived packet.
- In some cases, it may be advantageous to drop (or mark the header of) a packet before the buffer is full in order to provide a congestion signal to the sender.
- A number of packet-dropping and -marking policies (which collectively have become known as **active queue management (AQM)** algorithms) have been proposed and analyzed.
- One of the most widely studied and implemented AQM algorithms are the **Random Early Detection (RED)** algorithm. Under RED, a weighted average is maintained for the length of the output queue.
- If the average queue length is less than a minimum threshold, min_{th}, when a packet arrives, the packet is admitted to the queue.
- Conversely, if the queue is full or the average queue length is greater than a maximum threshold, max_{th}, when a packet arrives, the packet is marked or dropped.
- Finally, if the packet arrives to find an average queue length in the interval [min_{th}, max_{th}], the packet is marked or dropped with a probability that is typically some function of the average queue length, min_{th}, and max_{th}.

Consider the following scenario:

Suppose that in the below figure the switch fabric chooses to transfer the packet from the front of the upper-left queue. In this case, the darkly shaded packet in the lower-left queue must wait. But not only must this darkly shaded packet wait, so too must the lightly shaded packet that is queued behind that packet in the lower-left queue, even though there is no contention for the middle-right output port (the destination for the lightly shaded packet). This phenomenon is known as **head-of-the-line (HOL)** blocking.

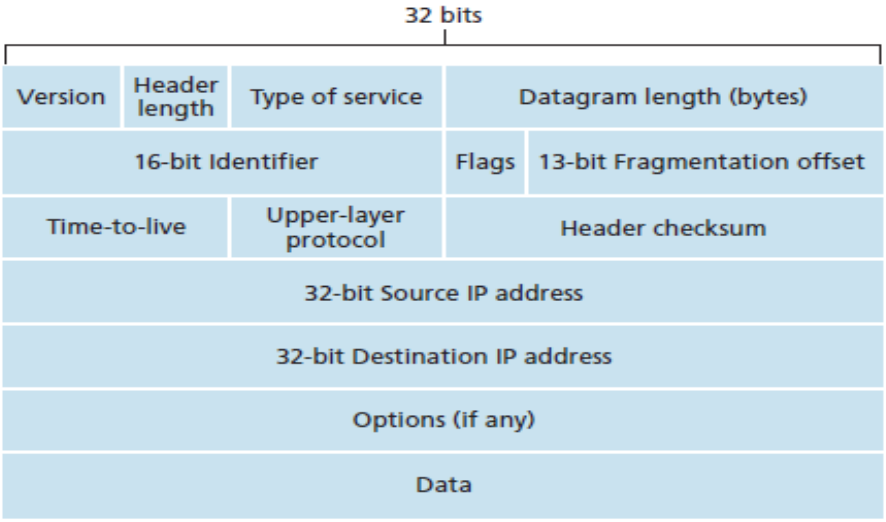


The Internet Protocol (IP)

Internet addressing and forwarding are important components of the Internet Protocol (IP).

There are two versions of IP in use today: IPv4, IPv6.

Datagram Format



- **Version number:** These 4 bits specify the IP protocol version of the datagram.
- **Header length:** Because an IPv4 datagram can contain a variable number of options these 4 bits specify the total header bytes.
- **Type of service:** The type of service (TOS) bits were included in the IPv4 header to allow different types of IP datagrams (for example, datagrams particularly requiring low delay, high throughput, or reliability) to be distinguished from each other.
- **Datagram length:** This is the total length of the IP datagram (header plus data), measured in bytes.
- **Identification:** this field represents identification number assigned to related fragments.
- **Flag:** there are three flags, first bit is unused, second bit is do not fragment bit, If this bit is set intermediate nodes should not perform fragmentation. Last bit is more fragment bit. More fragment bit represents more fragments to follow after this.
- **Fragmentation offset:** Starting byte of a fragment (In multiples of 8 bytes).
- **Time-to-live:** The time-to-live (TTL) field is included to ensure that datagrams do not circulate forever in the network. It represents hop limit.
- **Protocol:** Represents the upper layer protocol, 6 for TCP, 17 for UDP.
- **Header checksum:** field is used for error detection.
- **Source and destination IP addresses:** represents 32 bit source and destination IP address.
- **Options:** The options fields allow an IP header to be extended. It allows to include additional functionalities.
- **Data (payload):** represents the data that has to be transmitted.

IP Datagram Fragmentation

- The maximum amount of data that a link-layer frame can carry is called the maximum transmission unit (MTU). Because each IP datagram is encapsulated within the link-layer frame for transport from one router to the next router, the MTU of the link-layer protocol places a hard limit on the length of an IP datagram.
- IP datagram is divided into smaller packets according to MTU. These smaller packets are called fragments and process is called fragmentation.
- Fragments need to be reassembled before they reach the transport layer at the destination.

- Receiver needs to reassemble all the fragments belong to same original IP datagram. In order to identify all the related fragments **Identification** field is used.
- There are three flags, first bit is unused, second bit is do not fragment bit, If this bit is set intermediate nodes should not perform fragmentation. Last bit is more fragment bit. More fragment bit represents more fragments to follow after this.
- In order for the destination host to determine whether a fragment is missing the offset field is used to specify where the fragment fits within the original IP datagram.

Example:

A datagram of 4,000 bytes (20 bytes of IP header plus 3,980 bytes of IP payload) arrives at a router and must be forwarded to a link with an MTU of 1,500 bytes. This implies that the 3,980 data bytes in the original datagram must be allocated to three separate fragments. Suppose that the original datagram is stamped with an identification number of 777. Following table shows the fragmentation.

Fragment	Bytes	ID	Offset	Flag
1st fragment	1,480 bytes in the data field of the IP datagram	identification = 777	offset = 0 (meaning the data should be inserted beginning at byte 0)	flag = 1 (meaning there is more)
2nd fragment	1,480 bytes of data	identification = 777	offset = 185 (meaning the data should be inserted beginning at byte 1,480. Note that 185 · 8 = 1,480)	flag = 1 (meaning there is more)
3rd fragment	1,020 bytes (= 3,980–1,480–1,480) of data	identification = 777	offset = 370 (meaning the data should be inserted beginning at byte 2,960. Note that 370 · 8 = 2,960)	flag = 0 (meaning this is the last fragment)

At the destination, the payload of the datagram is passed to the transport layer only after the IP layer has fully reconstructed the original IP datagram. If one or more of the fragments does not arrive at the destination, the incomplete datagram is discarded and not passed to the transport layer.

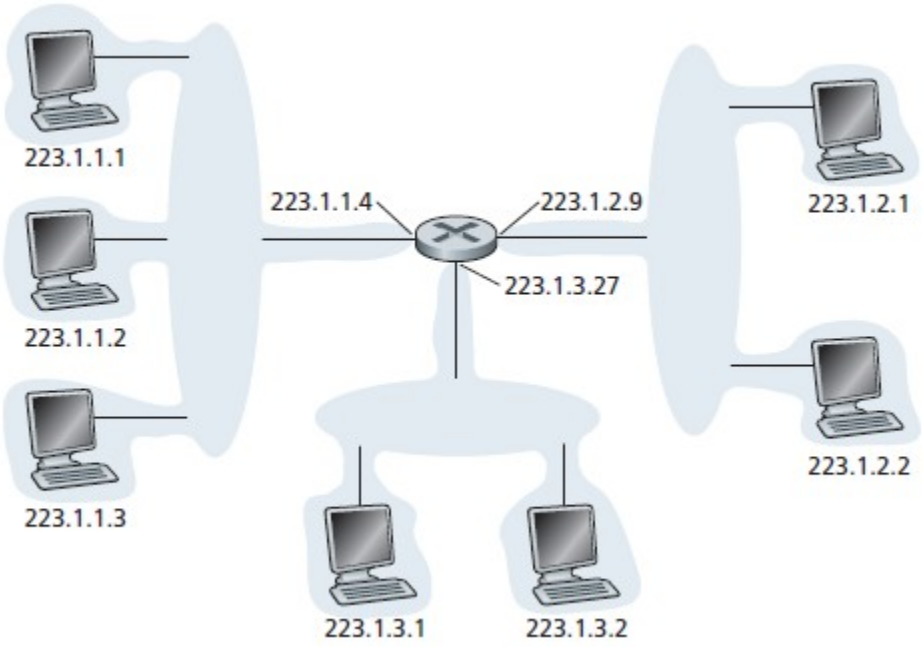
IPv4 Addressing

- Each IP address is 32 bits long (equivalently, 4 bytes), and there are thus a total of 232 possible IP addresses. Approximately there are about 4 billion possible IP addresses.

- IP addresses are typically written in so-called dotted-decimal notation, in which each byte of the address is written in its decimal form and is separated by a period (dot) from other bytes in the address.
- Ex: 193.32.216.9
- The address 193.32.216.9 in binary notation is 11000001 00100000 11011000 00001001
- Each interface on every host and router in the global Internet must have an IP address that is globally unique.
- IP address has 2 parts: Network ID and Host ID. Network ID is used to identify the network and Host ID is used to identify host in the network.
- A network can be divided into smaller sub networks called subnets.
- Initially IP address was divided into 5 classes. We call this as classful addressing. It leads to shortage of IP Address. Now Classless Inter Domain Routing (CIDR) addressing is used. In CIDR IP address is represented as a.b.c.d/x where x represents number of bits used for Network ID.
- Example.

  If 256 hosts are there last 8 bit is allocated to host ID remaining 24 bit is allocated to network ID. Here /x value is /24.

Below figure shows sub netting.



Dynamic Host Configuration Protocol

- Once an organization has obtained a block of addresses, it can assign individual IP addresses to the host and router interfaces in its organization.
- A system administrator will typically manually configure the IP addresses into the.
- Host addresses can also be configured manually, but more often this task is now done using the Dynamic Host Configuration Protocol (DHCP).
- DHCP allows a host to obtain (be allocated) an IP address automatically.
- DHCP works over UDP with port number 67.

DHCP involves four steps:

1) DHCP server discovery

Host broadcast DHCP discovery message with source address 0.0.0.0 and destination address 255.255.255.255

2) DHCP server offer(s)

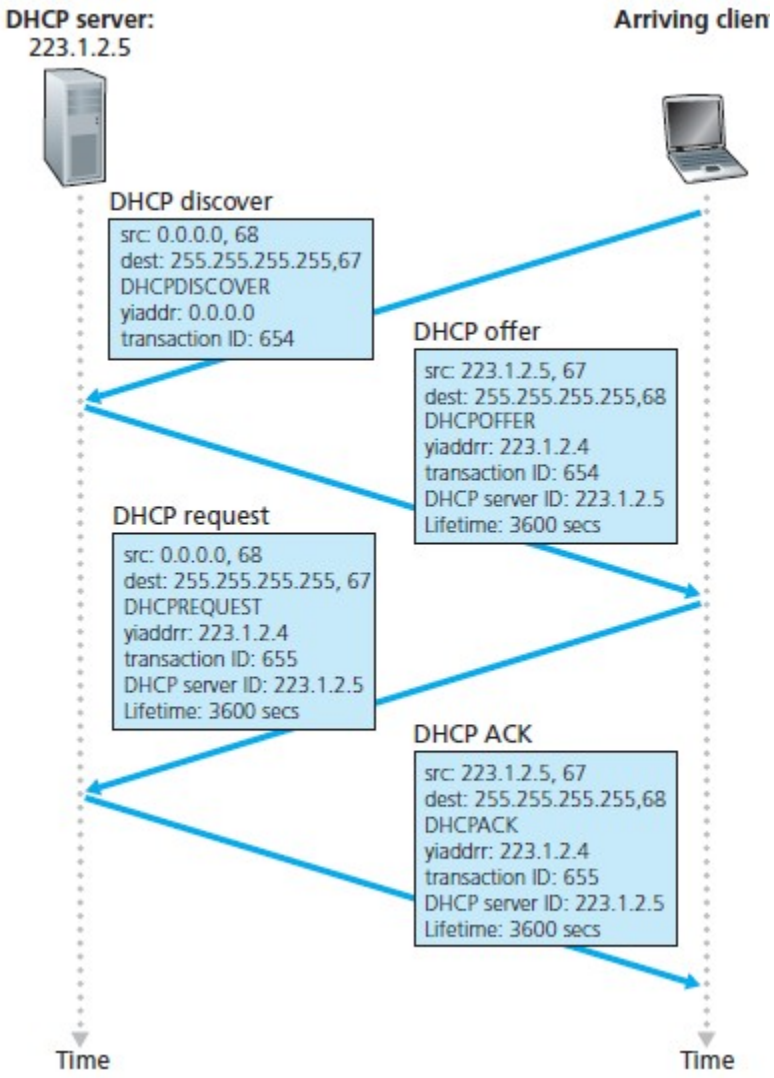
A DHCP server receiving a DHCP discover message responds to the client with a DHCP offer message that is broadcast to all nodes on the subnet, again using the IP broadcast address of 255.255.255.255.

3) DHCP request

The newly arriving client will choose from among one or more server offers and respond to its selected offer with a DHCP request message.

4) DHCP ACK

The server responds to the DHCP request message with a DHCP ACK message, confirming the requested parameters.



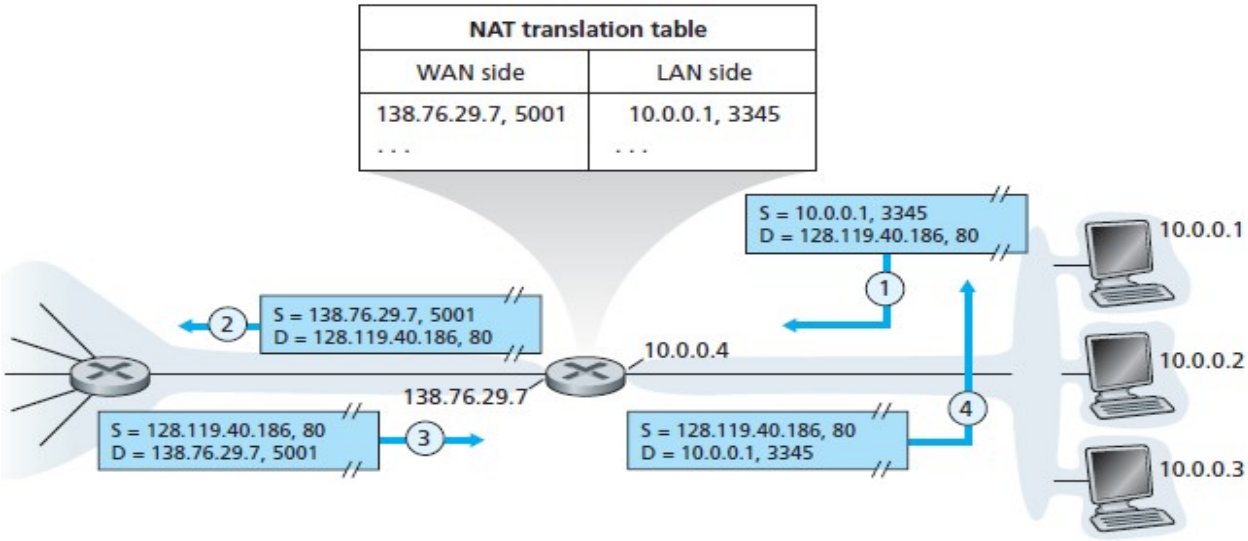
Network Address Translation (NAT)

- Private IP addresses shown below are used inside company/campus/organization or home network.

  10.0.0.0 to 10.255.255.255

  171.16.0.0 to 171.31.255.255

  192.168.0.0 to 192.168.255.255
- But company/campus/organization or home network connect to internet through global IP address.
- To convert from private IP address to global IP address and vice-versa NAT is used.
- NAT can be illustrated with the following diagram.



- Here the host with IP 10.0.0.1 sends the IP datagram with source address, port number 10.0.0.1, 3345 and destination address, port number 128.119.40.186, 80.
- NAT router maintains a NAT table as shown above
- NAT router make an entry in its table and replaces the source IP address with global IP address 138.76.29.7 and port number 5001 and send it to destination.
- When the response comes from destination, the global IP address will be replaced by private IP address according to entry available in NAT table. Then the message is delivered to appropriate host.

UPnP

- NAT traversal is increasingly provided by Universal Plug and Play (UPnP), which is a protocol that allows a host to discover and configure a nearby NAT.
- UPnP requires that both the host and the NAT be UPnP compatible.
- With UPnP, an application running in a host can request a NAT mapping between its (private IP address, private port number) and the (public IP address, public port number) for some requested public port number.
- If the NAT accepts the request and creates the mapping, then nodes from the outside can initiate TCP connections to (public IP address, public port number).
- Furthermore, UPnP lets the application know the value of (public IP address, public port number), so that the application can advertise it to the outside world.

Internet Control Message Protocol (ICMP)

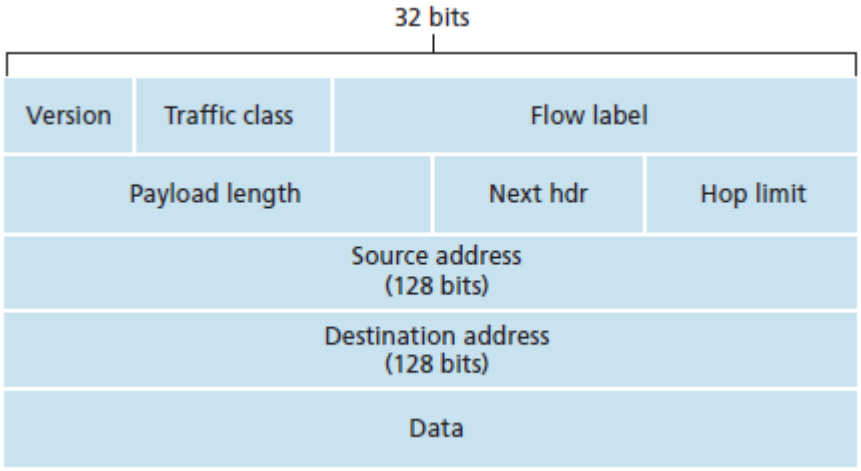
- ICMP is used by hosts and routers to communicate network- layer information to each other. The most typical use of ICMP is for error reporting.
- ICMP is part of IP but architecturally it lies just above IP hence ICMP messages are carried inside IP datagrams. That is, ICMP messages are carried as IP payload, just as TCP or UDP segments are carried as IP payload.
- ICMP messages have a type and a code field, and contain the header and the first 8 bytes of the IP datagram that caused the ICMP message to be generated in the first place.
- Popular ICMP messages are listed below

ICMP Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

- The well-known ping program sends an ICMP type 8 code 0 message to the specified host. The destination host, seeing the echo request, sends back a type 0 code 0 ICMP echo reply.
- Congested router send an ICMP source quench message to a host toforce that host to reduce its transmission rate.

IPv6

IPv6 Datagram Format



The most important changes introduced in IPv6 are:

- **Expanded addressing capabilities:** IPv6 increases the size of the IP address from 32 to 128 bits.
- **A streamlined 40-byte header:** 40 bytes of mandatory header is used in IPv6 whereas IPv4 uses 20 bytes of mandatory header.
- **Flow labeling and priority:** Flow label refers to labeling of packets belonging to particular flows for which the sender requests special handling, such as a non default quality of service or real-time service. The IPv6 header also has an 8-bit traffic class field. This field, like the TOS field in IPv4, can be used to give priority to certain datagrams within a flow.

The following fields are defined in IPv6:

- **Version:** This 4-bit field identifies the IP version number.
- Traffic class: This 8-bit field specify priority.
- **Flow label:** this 20-bit field is used to identify a flow of datagrams.
- **Payload length:** This 16-bit value is treated as an unsigned integer giving the number of bytes in the IPv6 datagram following the fixed-length, 40-byte datagram header.
- **Next header:** This field identifies the next following header.
- **Hop limit:** The contents of this field are decremented by one by each router that forwards the datagram. If the hop limit count reaches zero, the datagram is discarded.
- **Source and destination addresses:** 128 bit IPv6 address.

- **Data:** This is the payload portion of the IPv6 datagram.

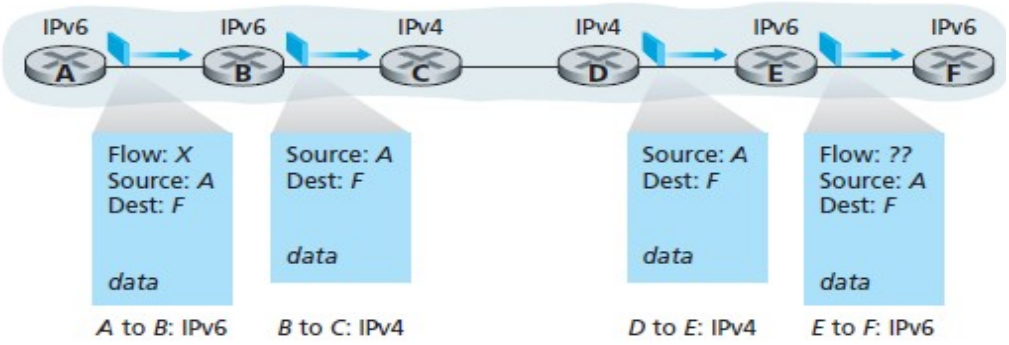
Following fields appearing in the IPv4 datagram are no longer present in the IPv6 datagram:

- **Fragmentation/Reassembly:** IPv6 does not allow for fragmentation and reassembly at intermediate routers; these operations can be performed only by the source and destination. If an IPv6 datagram received by a router is too large to be forwarded over the outgoing link, the router simply drops the datagram and sends a "Packet Too Big" ICMP error message (see below) back to the sender. The sender can then resend the data, using a smaller IP datagram size.
- **Header checksum:** Because the transport-layer and link-layer protocols in the Internet layers perform check-summing, the designers of IP probably felt that this functionality was sufficiently redundant in the network layer that it could be removed.
- **Options:** An options field is no longer a part of the standard IP header. Instead of option field extension headers are used.

Transitioning from IPv4 to IPv6

1) Dual-stack:

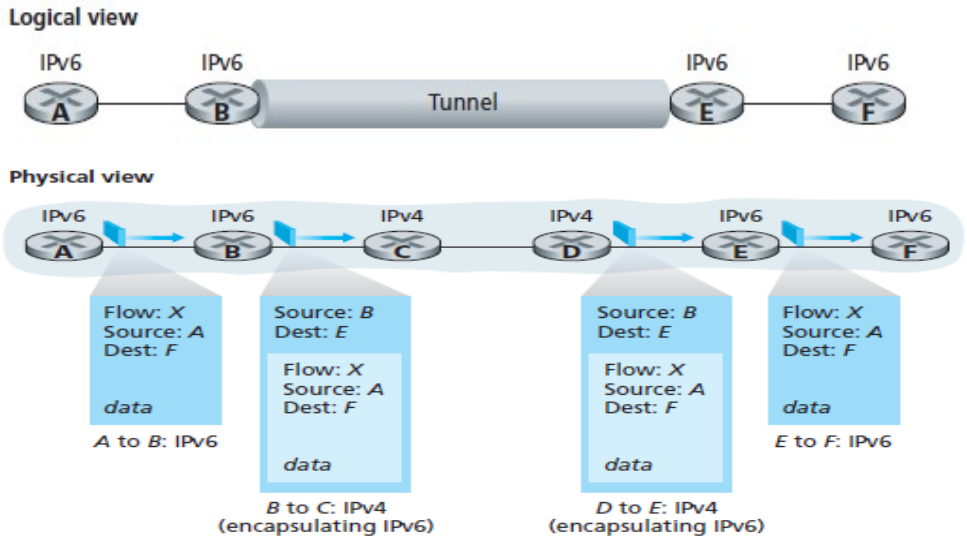
- Here IPv6 nodes also have a complete IPv4 implementation. Such a node, has the ability to send and receive both IPv4 and IPv6 datagrams. When interoperating with an IPv4 node, an IPv6/IPv4 node can use IPv4 datagrams; when interoperating with an IPv6 node, it can speak IPv6. IPv6/IPv4 nodes must have both IPv6 and IPv4 addresses. They must furthermore be able to determine whether another node is IPv6-capable or IPv4-only.
- In the dual-stack approach, if either the sender or the receiver is only IPv4- capable, an IPv4 datagram must be used. As a result, it is possible that two IPv6- capable nodes can end up, in essence, sending IPv4 datagrams to each other.



Example: Suppose Node A is IPv6-capable and wants to send an IP datagram to Node F, which is also IPv6-capable. Nodes A and B can exchange an IPv6 datagram. However, Node B must create an IPv4 datagram to send to C. Certainly, the data field of the IPv6 datagram can be copied into the data field of the IPv4 datagram and appropriate address mapping can be done. However, in performing the conversion from IPv6 to IPv4, there will be IPv6-specific fields in the IPv6 datagram that have no counterpart in IPv4. The information in these fields will be lost. Thus, even though E and F can exchange IPv6 datagrams, the arriving IPv4 datagrams at E from D do not contain all of the fields that were in the original IPv6 datagram sent from A.

2) Tunneling

Tunneling can solve the problem noted above. The basic idea behind tunneling is the following. Suppose two IPv6 nodes want to interoperate using IPv6 datagrams but are connected to each other by intervening IPv4 routers. We refer to the intervening set of IPv4 routers between two IPv6 routers as a tunnel. With tunneling, the IPv6 node on the sending side of the tunnel takes the entire IPv6 datagram and puts it in the data (payload) field of an IPv4 datagram.



IP Security

IPsec is the security protocol used for IP security. The services provided by an IPsec session include:

- **Cryptographic agreement:** Mechanisms that allow the two communicating hosts to agree on cryptographic algorithms and keys.

- **Encryption of IP datagram payloads:** When the sending host receives a segment from the transport layer, IPsec encrypts the payload. The payload can only bedecrypted by IPsec in the receiving host.
- **Data integrity:** IPsec allows the receiving host to verify that the datagram's header fields and encrypted payload were not modified while the datagram was en route from source to destination.
- **Origin authentication:** When a host receives an IPsec datagram from a trusted source, the host is assured that the source IP address in the datagram is the actual source of the datagram.

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.

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

Global or Decentralized 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.

Static or Dynamic 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.

Load Sensitive or Load Insensitive 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*

```
1  Initialization:
2    N' = {u}
3    for all nodes v
4      if v is a neighbor of u
5        then D(v) = c(u,v)
6      else D(v) = ∞
7
8  Loop
9    find w not in N' such that D(w) is a minimum
10   add w to N'
11   update D(v) for each neighbor v of w and not in N':
12         D(v) = min( D(v), D(w) + c(w,v) )
13   /* new cost to v is either old cost to v or known
14    least path cost to w plus cost from w to v */
15 until N'= N
```

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

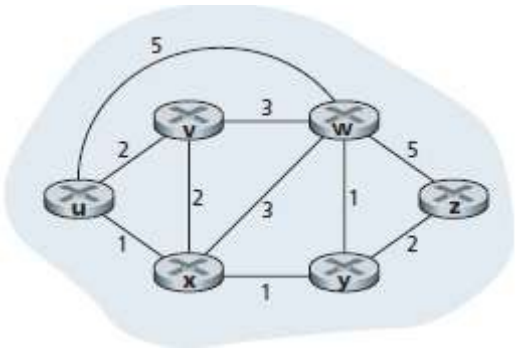


Figure 3.22: Abstract graph model of a computer network

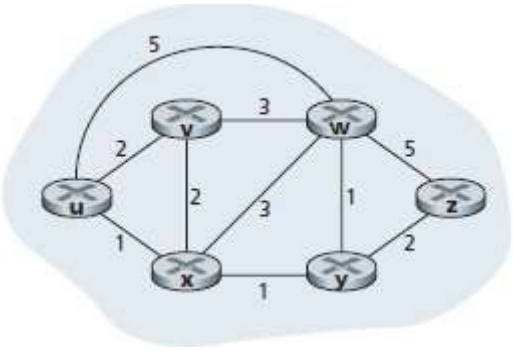
Solution:

- 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.

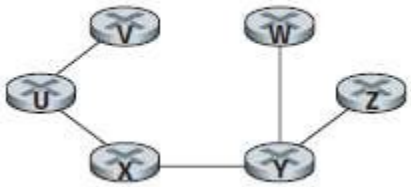
- A tabular summary of the algorithm's computation is shown in Table 3.5.

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

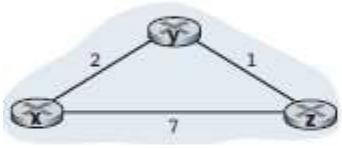
- Distance vector (DV) algorithm is 1) iterative, 2) asynchronous, and 3) distributed.
  1) It is distributed. This is because each node
     → receives some information from one or more of its directly attached neighbors
     → performs the calculation and
     → distributes then the results of the 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:
     Dx(y) = cost of the least-cost path from node x to node y, for all nodes in N.
     Dx = [Dx(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, Dx = [Dx(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, Dv = [Dv(y): y 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:
     Dx(y) = minv{c(x,v) + Dv(y)} for each node y 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*:

```
1  Initialization:
2     for all destinations y in N:
3        Dx(y) = c(x,y)    /* if y is not a neighbor then c(x,y) = ∞ */
4     for each neighbor w
5        Dw(y) = ? for all destinations y in N
6     for each neighbor w
7        send distance vector Dx = [Dx(y): y in N] to w
8
9  loop
10    wait (until I see a link cost change to some neighbor w or
11          until I receive a distance vector from some neighbor w)
12
13    for each y in N:
14       Dx(y) = minv{c(x,v) + Dv(y)}
15
16    if Dx(y) changed for any destination y
17       send distance vector Dx = [Dx(y): y in N] to all neighbors
18
19 forever
```

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



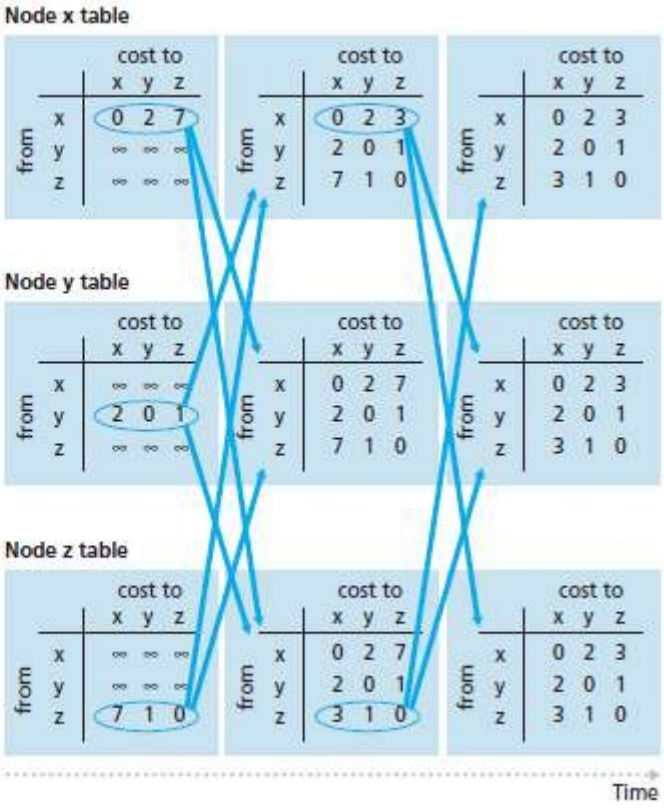


Figure 3.24: Distance-vector (DV) algorithm

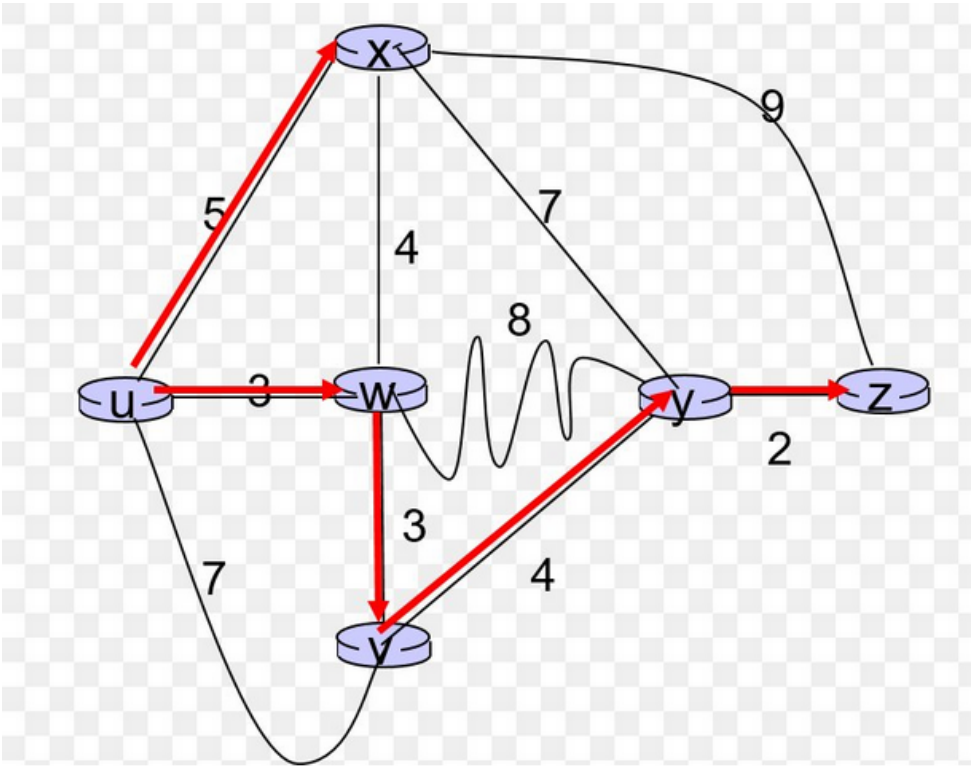
- The operation of the algorithm is illustrated in a synchronous manner.Here, all nodes simultaneously
  → receive distance vectors from their neighbours
  → compute their new distance vectors, and
  → inform their neighbours if their distance vectors have changed.
- The table in the upper-left corner 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 Dx = [Dx(x), Dx(y), Dx(z)] = [0, 2, 7].
- After initialization, each node sends its distance vector to each of its twoneighbours.
- 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
  Dx(x) = 0
  Dx(y) = min{c(x,y) + Dy(y), c(x,z) + Dz(y)} = min{2 + 0, 7 + 1} = 2
  Dx(z) = min{c(x,y) + Dy(z), c(x,z) + Dz(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, Dx(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)

LS



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.

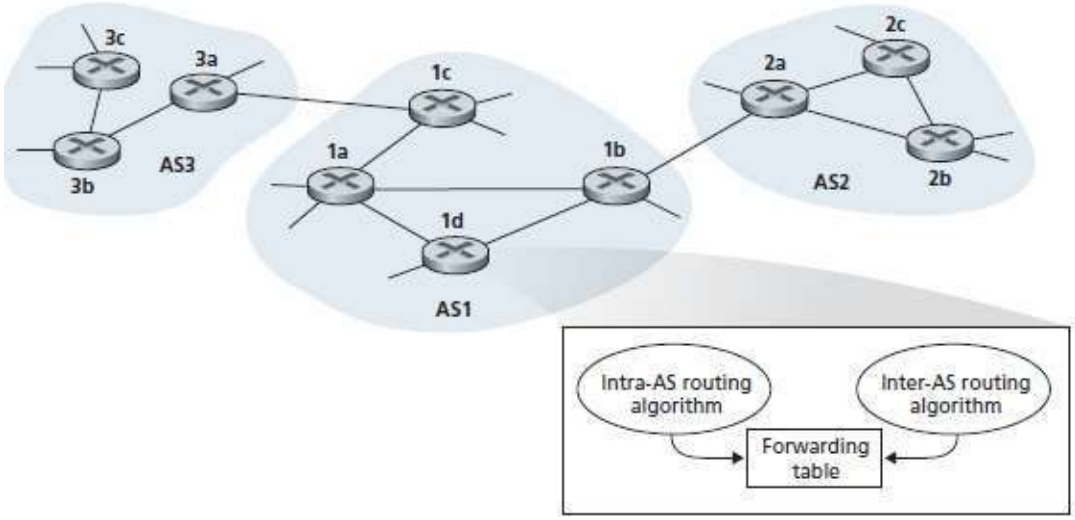


Figure 3.25: An example of interconnected autonomous-systems

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.

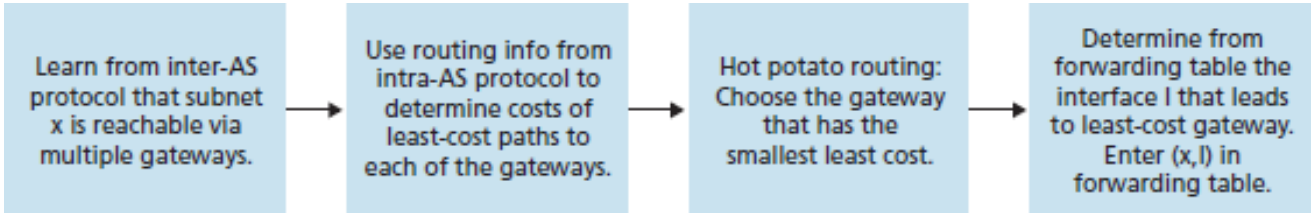


Figure 3.26: Steps in adding an outside-AS destination in a router's forwarding-table

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.

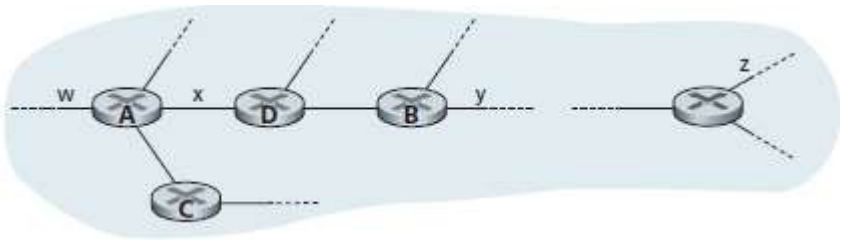


Figure 3.27: A portion of an autonomous-system

- 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.

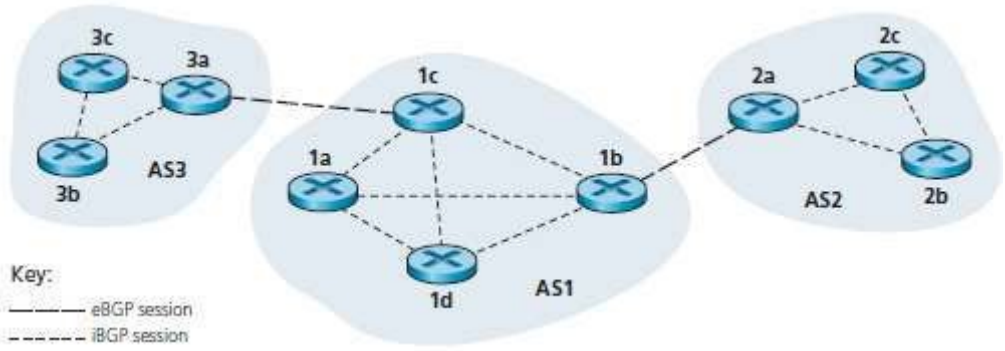


Figure 3.29: eBGP and iBGP sessions

- 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.

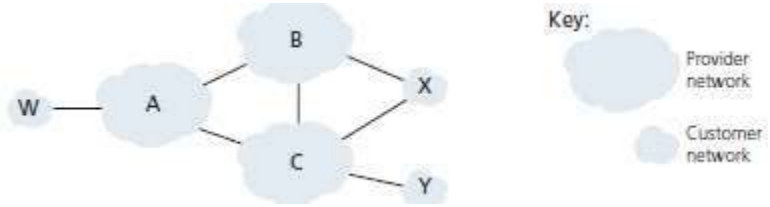


Figure 3.30: A simple BGP scenario

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.

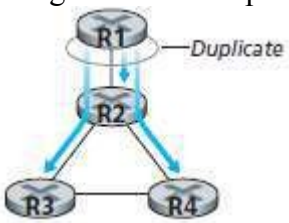


Figure 3.31: Duplicate creation/transmission

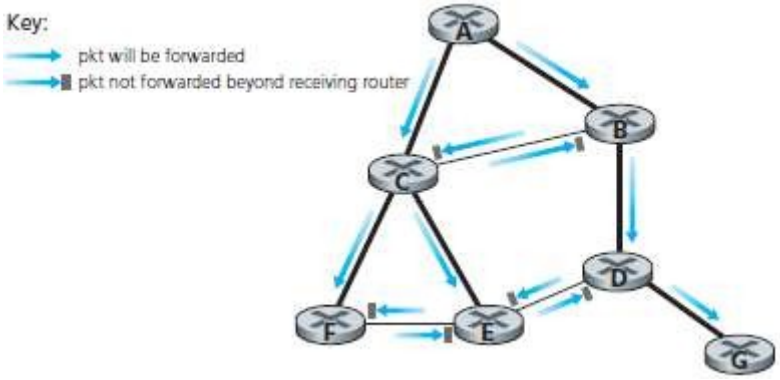


Figure 3.32: Reverse path forwarding

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.

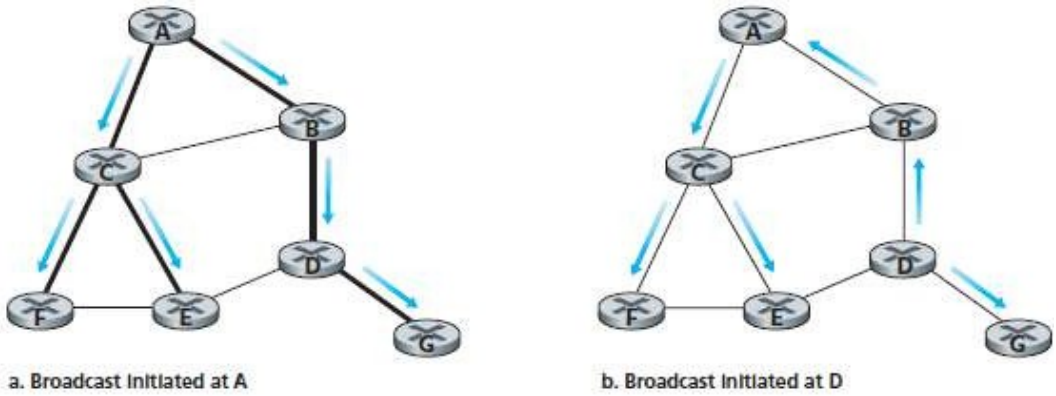


Figure 3.33: Broadcast along a spanning-tree

- 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.

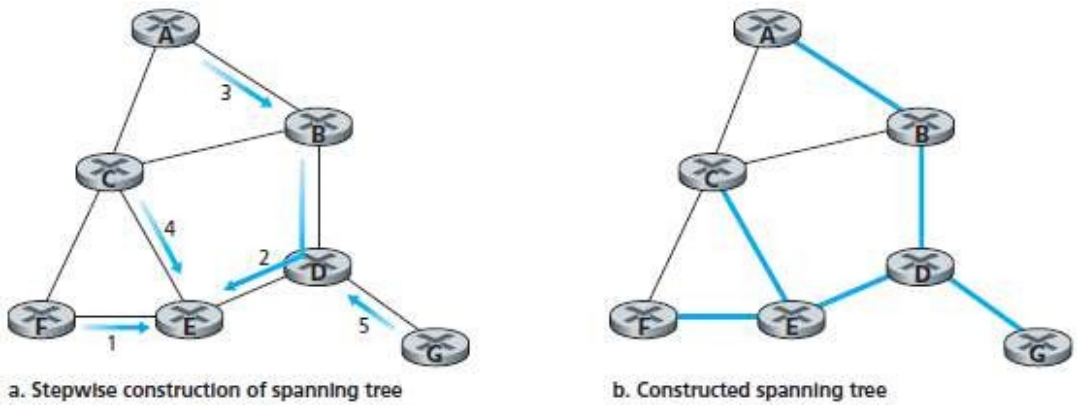


Figure 3.34: Center-based construction of a spanning-tree

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 illustrated in Figure 3.35.

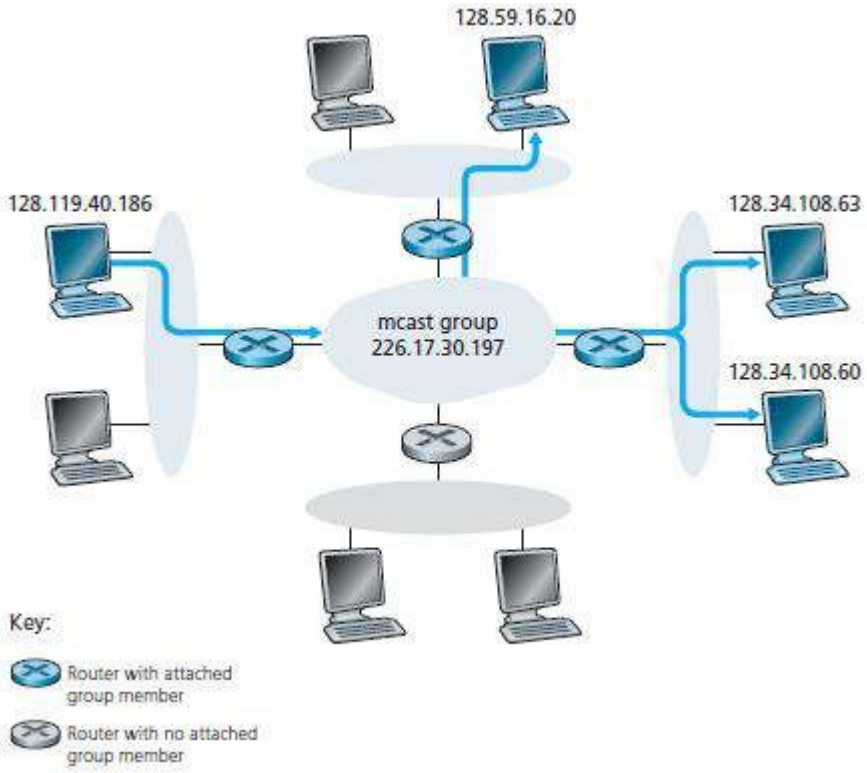


Figure 3.35: The multicast group: A datagram addressed to the group is delivered to all members of the multicast group

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.

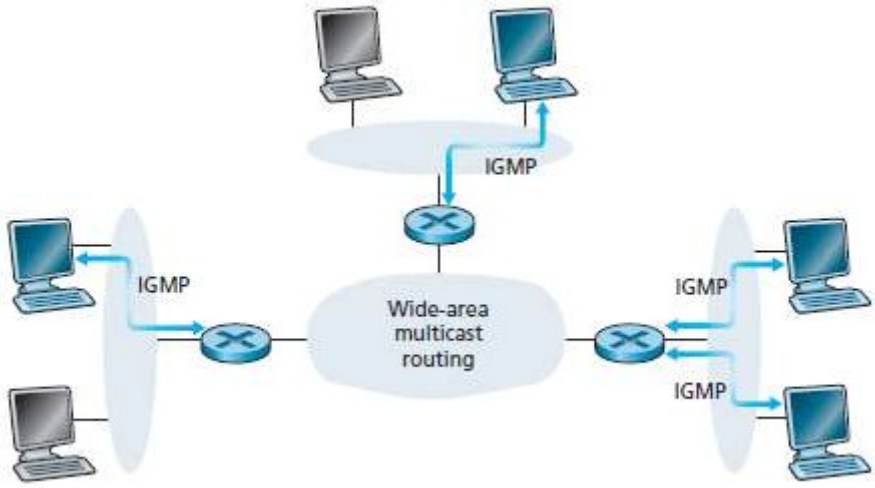


Figure 3.36: The two components of network-layer multicast in the Internet: IGMP and multicast-routing protocols

- 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?

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.

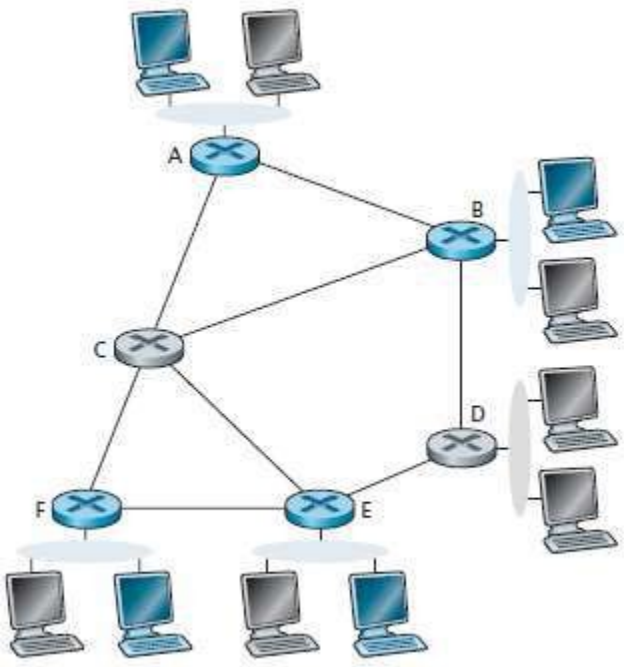


Figure 3.37: Multicast hosts, their attached routers, and other routers

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.