Module 3
Network Layer

Introduction

- Figure 4.1 shows a simple network with two hosts, H1 and H2, and several routers on the path between H1 and H2.
- Suppose that H1 is sending information to H2, and consider the role of the network layer in these hosts and in the intervening routers.
- The network layer in H1 takes segments from the transport layer in H1, encapsulates each segment into a datagram (that is, a network-layer packet), and then sends the datagrams to its nearby router, R1.
- At the receiving host, H2, the network layer receives the datagrams from its nearby router R2, extracts the transport-layer segments, and delivers the segments up to the transport layer at H2.
- The primary role of the routers is to forward datagrams from input links to output links.

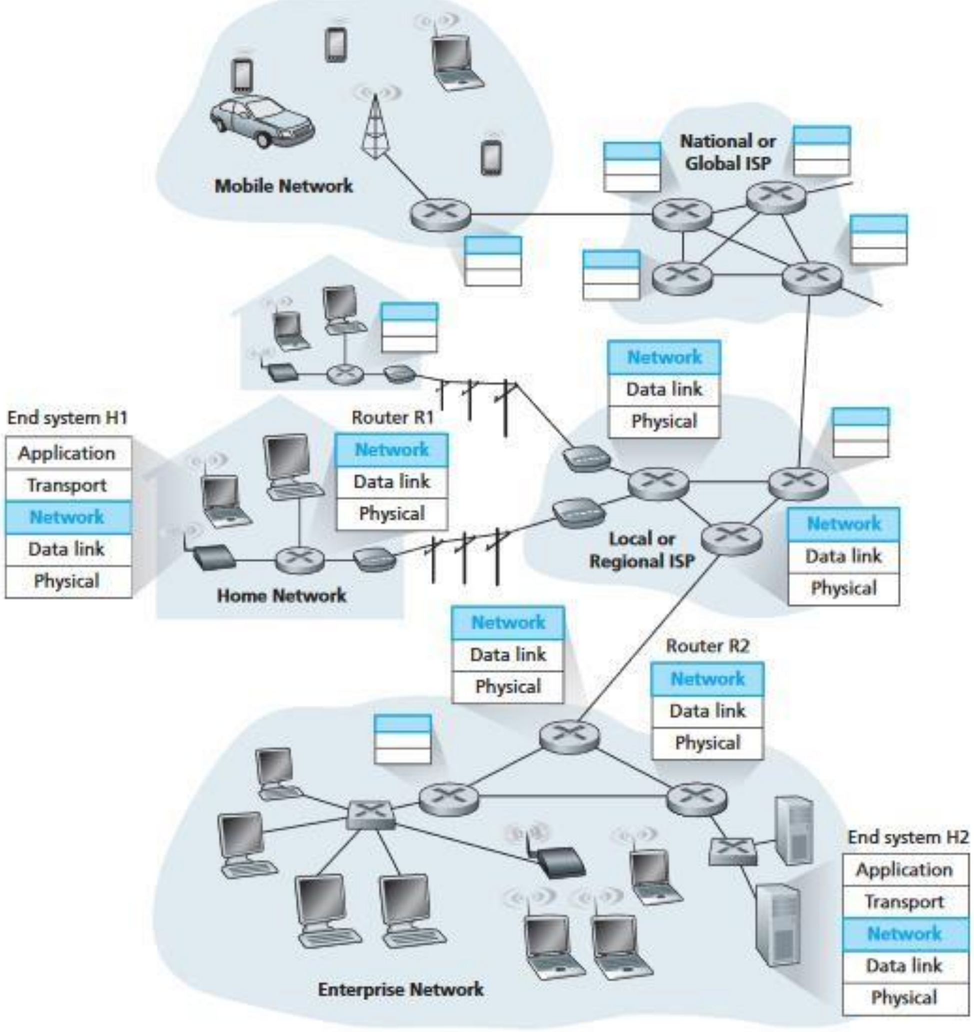
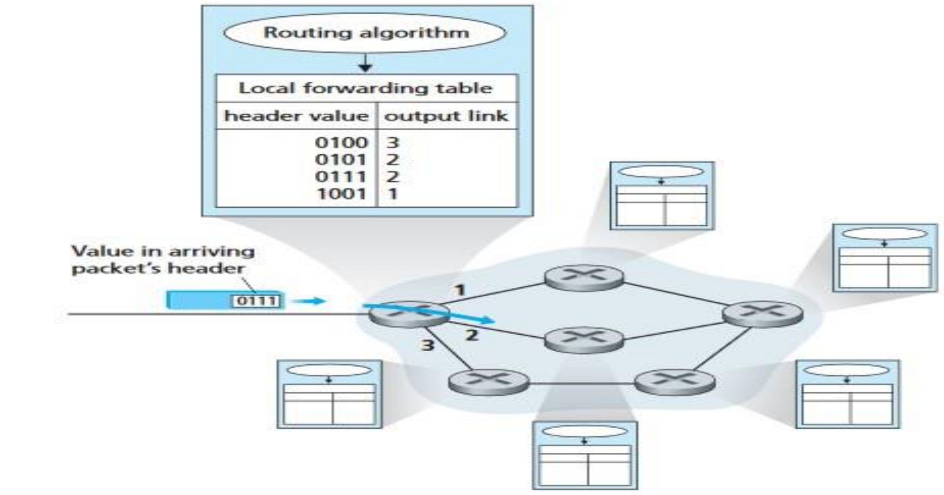


Figure 4.1 • The network layer

4.1.1 Forwarding and Routing

- The role of the network layer is thus deceptively simple—to move packets from a sending host to a receiving host.
- To do so, two important network-layer functions can be identified:

- Forwarding.
  - When a packet arrives at a router’s input link, the router must move the packet to the appropriate output link.
  - For example, a packet arriving from Host H1 to Router R1 must be forwarded to the next router on a path to H2.
- Routing.
  - The network layer must determine the route or path taken by packets as they flow from a sender to a receiver.
  - The algorithms that calculate these paths are referred to as routing algorithms.
  - A routing algorithm would determine, for example, the path along which packets flow from H1 to H2.
  - Forwarding refers to the router-local action of transferring a packet from an input link interface to the appropriate output link interface.
  - Routing refers to the network-wide process that determines the end-to-end paths that packets take from source to destination.
  - Every router has a forwarding table.
  - A router forwards a packet by examining the value of a field in the arriving packet’s header, and then using this header value to index into the router’s forwarding table.
  - The value stored in the forwarding table entry for that header indicates the router’s outgoing link interface to which that packet is to be forwarded. Depending on the network-layer protocol, the header value could be the destination address of the packet or an indication of the connection to which the packet belongs.
  - Figure 4.2 provides an example.
  - In Figure 4.2, a packet with a header field value of 0111 arrives to a router.
  - The router indexes into its forwarding table and determines that the output link interface for this packet is interface 2.
  - The router then internally forwards the packet to interface 2.



Local forwarding table

header value	output link
0100	3
0101	2
0111	2
1001	1

Figure 4.2 ◆ Routing algorithms determine values in forwarding tables

  - As shown in Figure 4.2, the routing algorithm determines the values that are inserted into the routers’ forwarding tables.

- The routing algorithm may be centralized (e.g., with an algorithm executing on a central site and downloading routing information to each of the routers) or decentralized (i.e., with a piece of the distributed routing algorithm running in each router).
- In either case, a router receives routing protocol messages, which are used to configure its forwarding table.

Network Service Models

- The network service model defines the characteristics of end-to-end transport of packets between sending and receiving end systems.
- In the sending host, when the transport layer passes a packet to the network layer, specific services that could be provided by the network layer include:
- Guaranteed delivery.
- This service guarantees that the packet will eventually arrive at its destination.
- Guaranteed delivery with bounded delay.
- This service not only guarantees delivery of the packet, but delivery within a specified host-to-host delay bound (for example, within 100 msec).

The following services could be provided to a flow of packets between a given source and destination:

- In-order packet delivery
  - This service guarantees that packets arrive at the destination in the order that they were sent.
- Guaranteed minimal bandwidth
  - This network-layer service emulates the behaviour of a transmission link of a specified bit rate (for example, 1 Mbps) between sending and receiving hosts.
  - As long as the sending host transmits bits (as part of packets) at a rate below the specified bit rate, then no packet is lost and each packet arrives within a pre-specified host-to-host delay (for example, within 40 msec).
- Guaranteed maximum jitter
  - This service guarantees that the amount of time between the transmission of two successive packets at the sender is equal to the amount of time between their receipt at the destination (or that this spacing changes by no more than some specified value).
- Security services
  - Using a secret session key known only by a source and destination host, the network layer in the source host could encrypt the payloads of all datagrams being sent to the destination host.
  - The network layer in the destination host would then be responsible for decrypting the payloads.

Virtual Circuit Networks

A VC consists of

(1) A path (that is, a series of links and routers) between the source and destination hosts

(2) VC numbers, one number for each link along the path

(3) Entries in the forwarding table in each router along the path. A packet belonging to a virtual circuit will carry a VC number in its header.

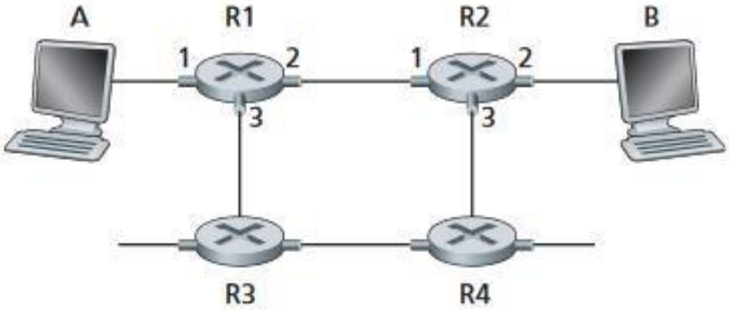
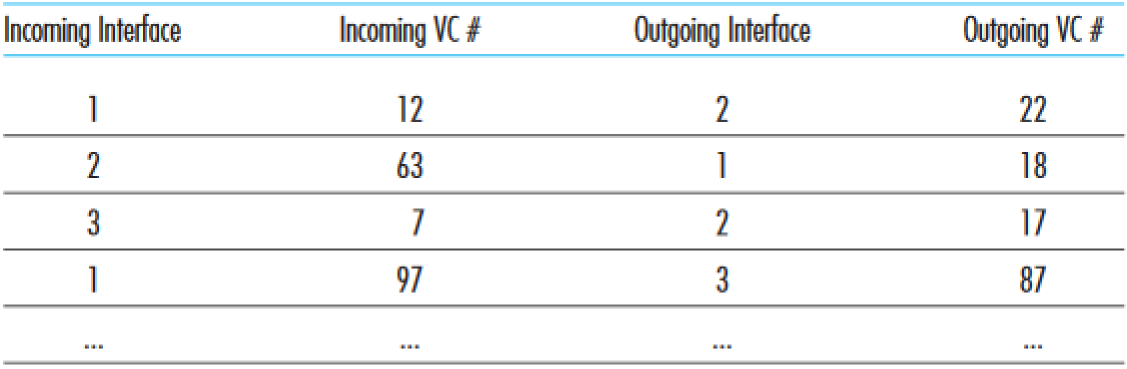


Figure 4.3 ♦ A simple virtual circuit network

- To illustrate the concept, consider the network shown in Figure 4.3.
- The numbers next to the links of R1 in Figure 4.3 are the link interface numbers.
- Suppose now that Host A requests that the network establish a VC between itself and Host B.
- Suppose also that the network chooses the path A-R1-R2-B and assigns VC numbers 12, 22, and 32 to the three links in this path for this virtual circuit. In this case, when a packet in this VC leaves Host A, the value in the VC number field in the packet header is 12; when it leaves R1, the value is 22; and when it leaves R2, the value is 32.
- For a VC network, each router's forwarding table includes VC number translation for example, the forwarding table in R1 might look something like this:



Incoming Interface	Incoming VC #	Outgoing Interface	Outgoing VC #
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

There are three identifiable phases in a virtual circuit:

- VC setup
  - During the setup phase, the sending transport layer contacts the network layer, specifies the receiver's address, and waits for the network to set up the VC.
  - The network layer determines the path between sender and receiver, that is, the series of links and routers through which all packets of the VC will travel.
  - The network layer also determines the VC number for each link along the path.
  - Finally, the network layer adds an entry in the forwarding table in each router along the path.
  - During VC setup, the network layer may also reserve resources (for example, bandwidth) along the path of the VC
- Data transfer
  - As shown in Figure 4.4, once the VC has been established, packets can begin to flow along the VC.
- VC teardown
  - This is initiated when the sender (or receiver) informs the network layer of its desire to terminate the VC.
  - The network layer will then typically inform the end system on the other side of the network of the call termination and update the forwarding tables in each of the packet routers on the path to indicate that the VC no longer exists.
  - The messages that the end systems send into the network to initiate or terminate a VC, and the messages passed between the routers to set up the VC (that is, to modify connection state in router tables) are known as signalling messages, and the protocols used to exchange these messages are often referred to as signalling protocols.
  - VC setup is shown pictorially in Figure 4.4.

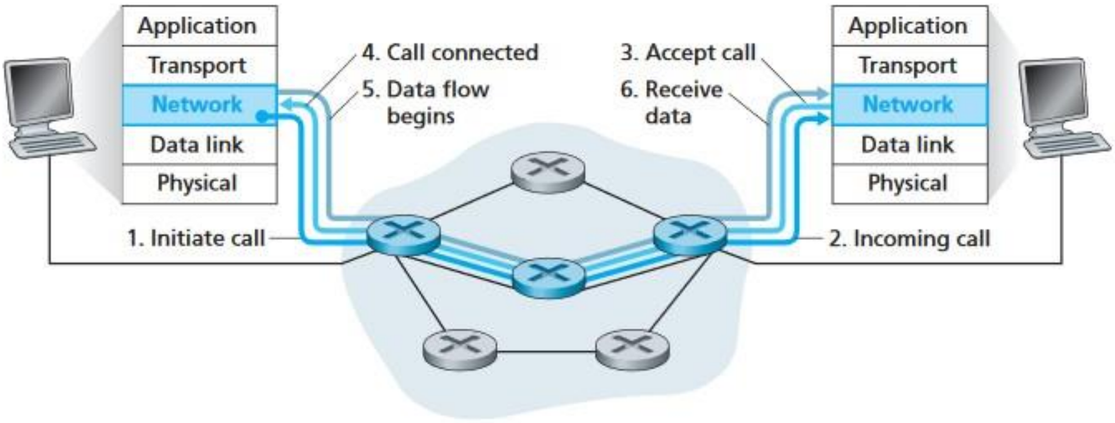


Figure 4.4 ♦ Virtual-circuit setup

Datagram Networks

- In a datagram network, each time an end system wants to send a packet, it stamps the packet with the address of the destination end system and then pops the packet into the network.
- As shown in Figure 4.5, there is no VC setup and routers do not maintain any VC state information

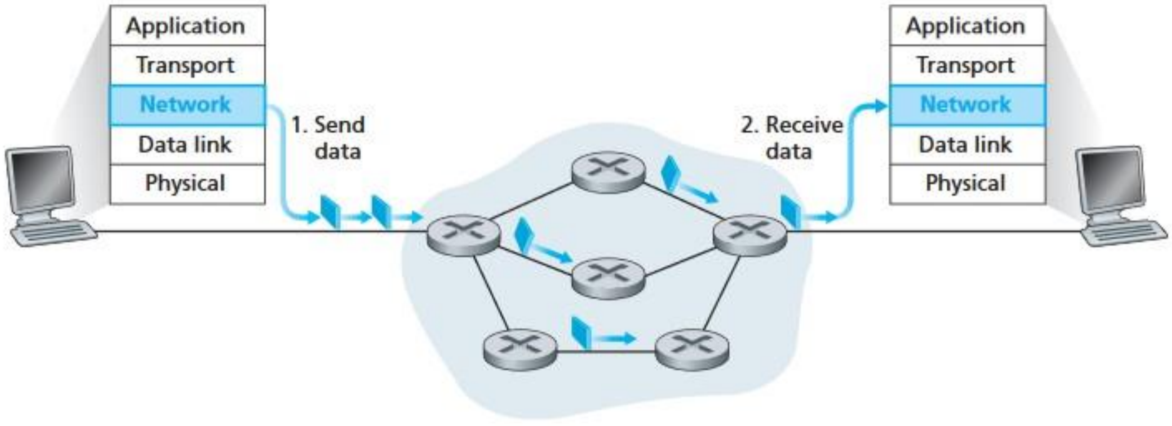


Figure 4.5 ♦ Datagram network

What's inside Router

- A high-level view of generic router architecture is shown in Figure 4.6.
- Four router components can be identified:

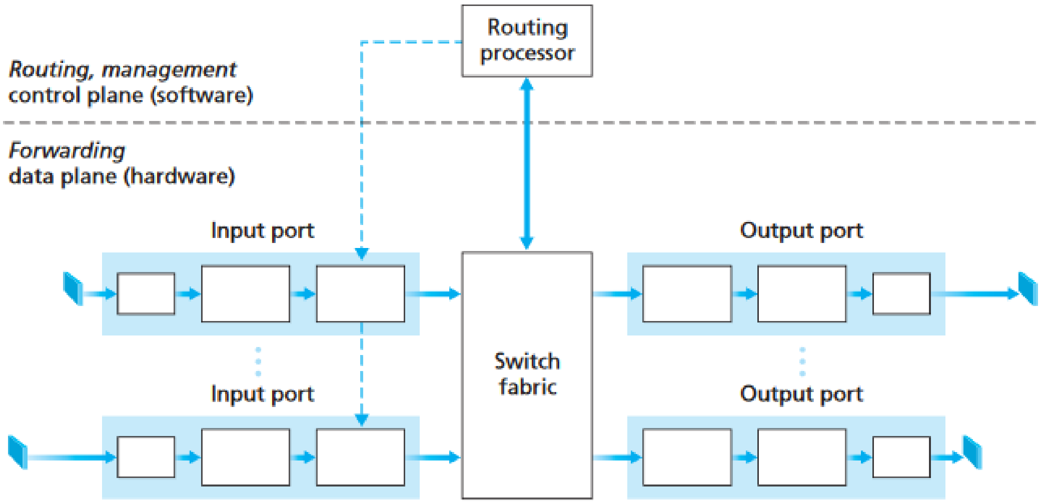


Figure 4.6 ♦ Router architecture

Input ports

- An input port performs several key functions.
- It performs the physical layer function of terminating an incoming physical link at a router; this is shown in the leftmost box of the input port and the rightmost box of the output port in Figure 4.6.
- An input port also performs link-layer functions needed to interoperate with the link layer at the other side of the incoming link; this is represented by the middle boxes in the input and output ports.

Switching fabric

- The switching fabric connects the router's input ports to its output ports.
- This switching fabric is completely contained within the router - a network inside of a network router!

Output ports

- An output port stores packets received from the switching fabric and transmits these packets on the outgoing link by performing the necessary link-layer and physical-layer functions.
- When a link is bidirectional (that is, carries traffic in both directions), an output port will typically be paired with the input port for that link on the same line card (a printed circuit board containing one or more input ports, which is connected to the switching fabric).

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, as shown in Figure 4.6.
- These forwarding functions are sometimes collectively referred to as the **router forwarding plane.**

Input Processing

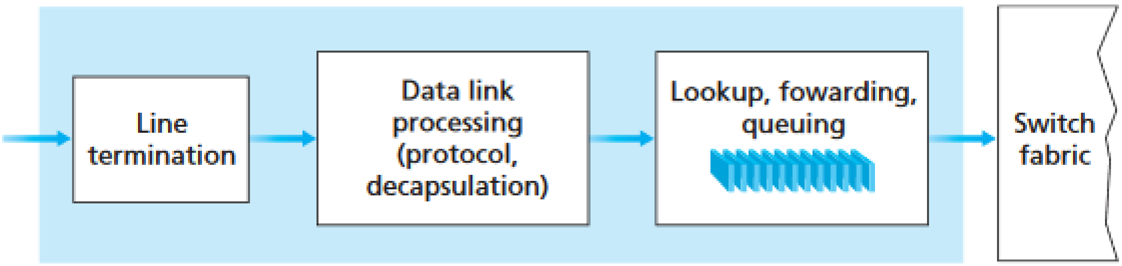

Line termination
Data link processing (protocol, decapsulation)
Lookup, fowarding, queuing
Switch fabric

Figure 4.7 ♦ Input port 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 (e.g., a PCI bus) indicated by the dashed line from the routing processor to the input line cards in Figure 4.6.
- With a shadow copy, forwarding decisions can be made locally, at each input port, without invoking the centralized routing processor on a per-packet basis and thus avoiding a centralized processing bottleneck.

Switching

- The switching fabric is at the very heart of a router, as it is through this fabric that the packets are actually switched (that is, forwarded) from an input port to an output port. Switching can be accomplished in a number of ways, as shown in Figure 4.8:

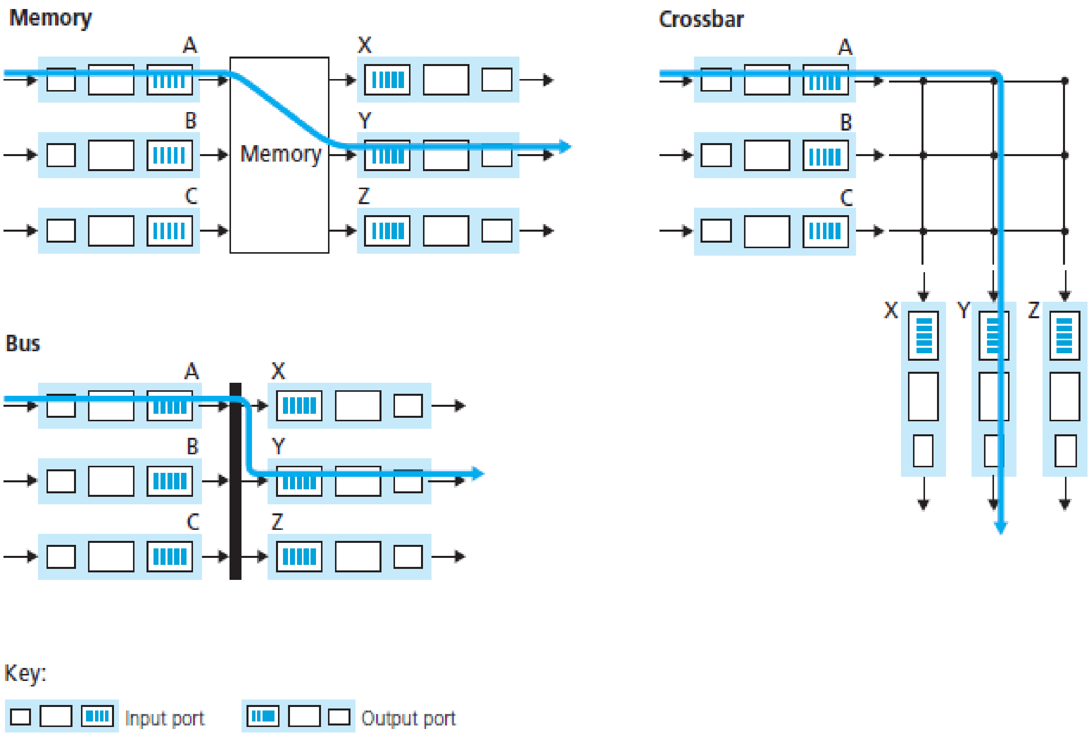


Figure 4.8 ♦ Three switching techniques

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 signalled 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.
- In this scenario, if the memory bandwidth is such that B packets per second can be written into, or read from, memory, then the overall forwarding throughput (the total rate at which packets are transferred from input ports to output ports) must be less than B/2. Note also that 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.

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.
- Because every packet must cross the single bus, the switching speed of the router is limited to the bus speed; in our roundabout analogy, this is as if the roundabout could only contain one car at a time.

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 a 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 cross point, which can be opened or closed at any time by the switch fabric controller (whose logic is part of the switching fabric itself).
- When a packet arrives from port A and needs to be forwarded to port Y, the switch controller closes the cross point 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 Porting

- 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.

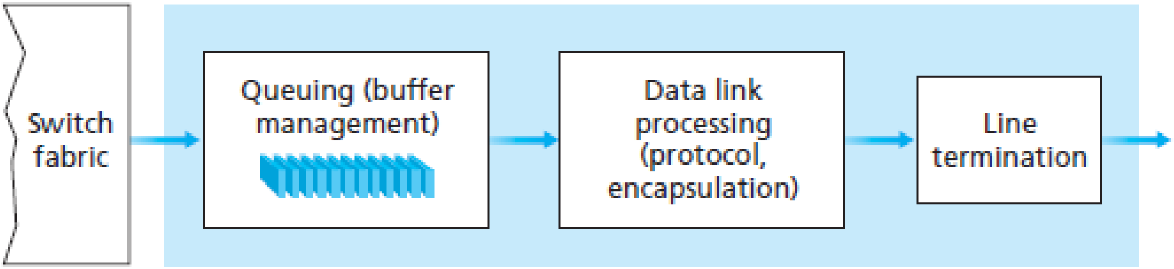

Switch fabric
Queuing (buffer management)
Data link processing (protocol, encapsulation)
Line termination

Figure 4.9 ♦ Output port processing

Where does Queuing Occur?

- Packet queues may form at both the input ports and the output ports.
- The location and extent of queuing (either at the input port queues or the output port queues) will depend on the traffic load, the relative speed of the switching fabric, and the line speed.
- Suppose that the input and output line speeds (transmission rates) all have an identical transmission rate of Rline packets per second, and that there are N input ports and N output ports.
- Let's assume that all packets have the same fixed length, and the packets arrive to input ports in a synchronous manner.
- That is, the time to send a packet on any link is equal to the time to receive a packet on any link, and during such an interval of time, either zero or one packet can arrive on an input link.
- Define the switching fabric transfer rate Rswitch as the rate at which packets can be moved from input port to output port.
- If Rswitch is N times faster than Rline, then only negligible queuing will occur at the input ports.
- This is because even in the worst case, where all N input lines are receiving packets, and all packets are to be forwarded to the same output port, each batch of N packets (one packet per input port) can be cleared through the switch fabric before the next batch arrives.

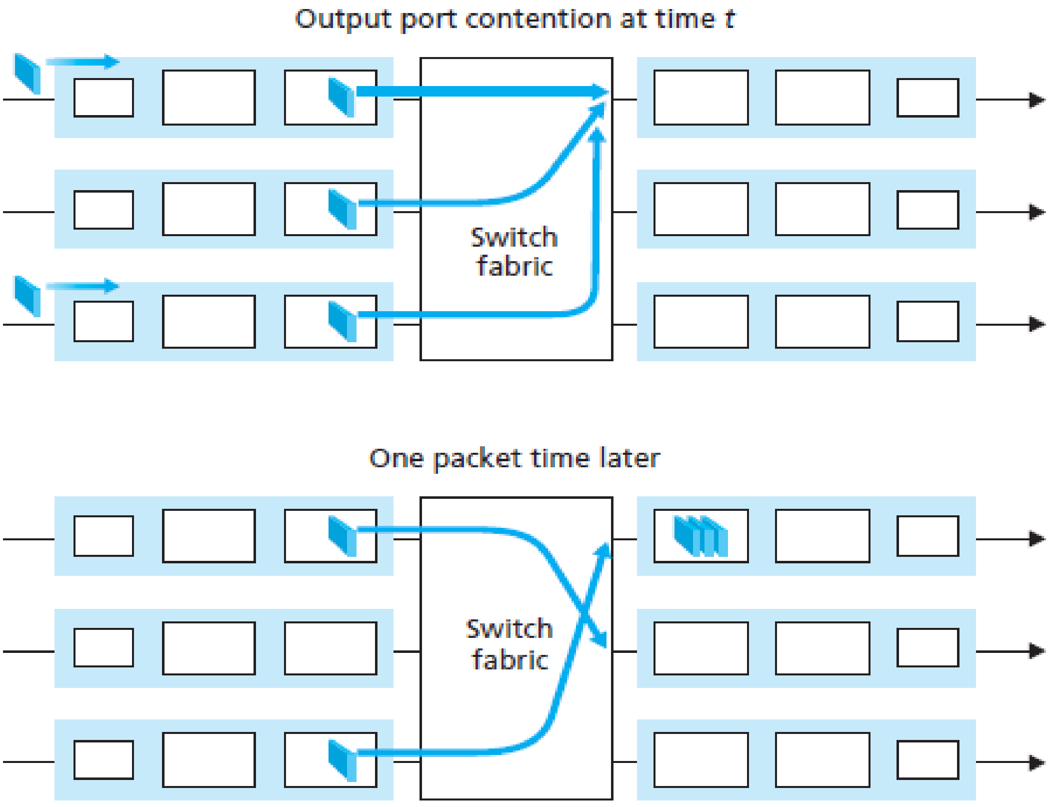


Figure 4.10 ♦ Output port queuing

- Output port queuing is illustrated in Figure 4.10. At time t, a packet has arrived at each of the incoming input ports, each destined for the uppermost outgoing port.
- Assuming identical line speeds and a switch operating at three times the line speed, one time unit later (that is, in the time needed to receive or send a packet), all three original packets have been transferred to the outgoing port and are queued awaiting transmission. In the next time unit, one of these three packets will have been transmitted over the outgoing link.
- In our example, two new packets have arrived at the incoming side of the switch; one of these packets is destined for this uppermost output port.
- For many years, the rule of thumb [RFC 3439] for buffer sizing was that the amount of buffering (B) should be equal to an average round-trip time (RTT, say 250 msec) times the link capacity (C).
- This result is based on an analysis of the queuing dynamics of a relatively small number of TCP flows.
- Thus, a 10 Gbps link with an RTT of 250 msec would need an amount of buffering equal to B = RTT·C= 2.5 Gbits of buffers.
- Recent theoretical and experimental efforts, however, suggest that when there are a large number of TCP flows (N) passing through a link, the amount of buffering needed is B = RTTC/√N.
- A consequence of output port queuing is that a **packet scheduler** at the output port must choose one packet among those queued for transmission.
- This selection might be done on a simple basis, such as first-come-first-served (FCFS)

scheduling, or a more sophisticated scheduling discipline such as weighted fair queuing (WFQ), which shares the outgoing link fairly among the different end-to-end connections that have packets queued for transmission.

- 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 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 analysed.
- One of the most widely studied and implemented AQM algorithms is the Random Early Detection (RED) algorithm.

The Internet Protocol (IP)

Forwarding and 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.
- Connection-less means there is no connection setup b/w the sender and the receiver.
- 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

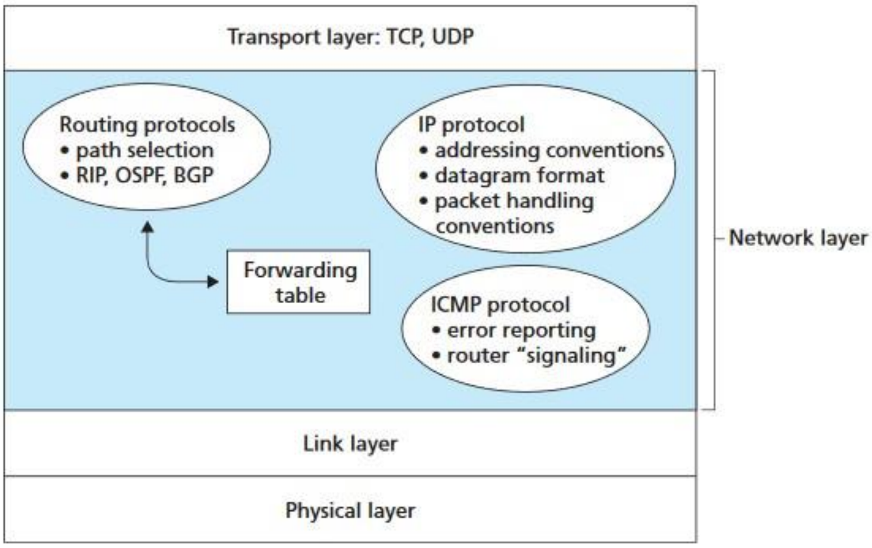


Figure 4.12 ♦ A look inside the Internet's network layer

Two important components of IP:

- Internet addressing and
- 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 datagram

IPV4 Datagram Format

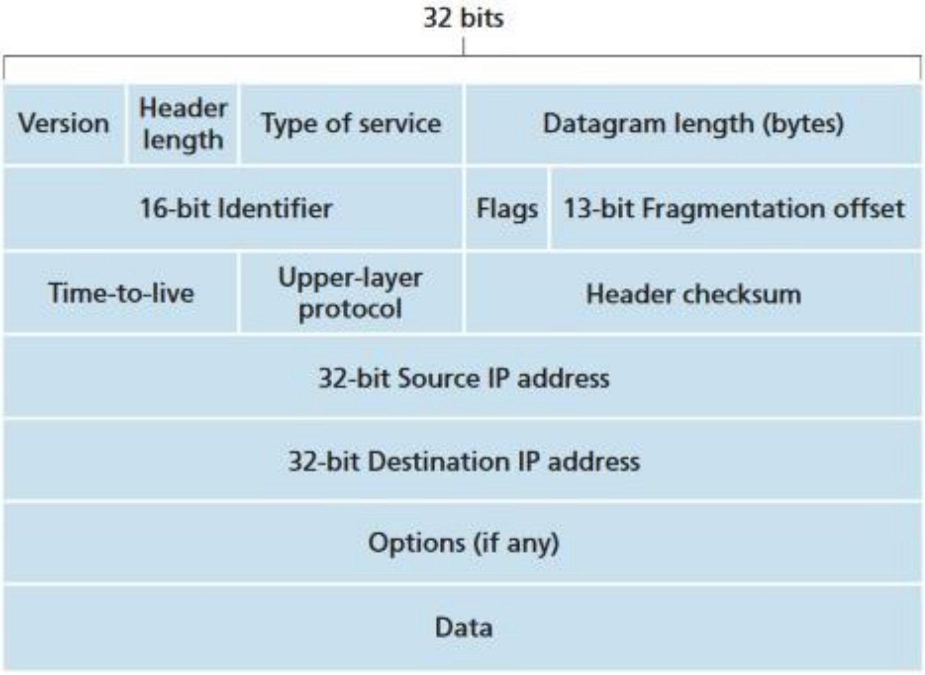


Figure 4.13 ♦ IPv4 datagram format

- Payload (or Data)
- This field contains the data to be delivered to the destination.
- Header
- Header contains information essential to routing and delivery.
- IP header contains following fields
- Version
- This field specifies version of the IPv4 datagram, i.e. 4.
- Header Length
- This field specifies length of header.
- Without options field, header length = 5 bytes.
- Type of Service (TOS)
- This field specifies priority of packet based on parameters such as delay, throughput, reliability & cost.
- Datagram Length
- This field specifies the total length of the datagram (header + data)
- Maximum length=65535 bytes.
- 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.
- **Identifier**: This field uniquely identifies a datagram packet.
    - **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.

- **Fragmentation Offset**: This field identifies location of a fragment in a datagram.
- 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.
- Protocol
- This field specifies upper layer protocol used to receive the datagram at the destination-host.
- For example, TCP=6 and UDP=17.
- Header Checksum
- This field is used to verify integrity of header only.
- If the verification process fails, the packet is discarded.
- Source IP Address & Destination IP Address
- These fields contain the addresses of source and destination respectively
- Options

  This field allows the packet to request special features such as
  →security level
  →route to be taken by packet at each router.

IP Datagram Fragmentation

- Each network imposes a restriction on maximum size of packet that can be carried.
- This is called the MTU (maximum transmission unit).
- For example:

  MTU Ethernet = 1500 bytes
  MTU FDDI = 4464 bytes
- Fragmentation means
- "The datagram is divided into smaller fragments when size of a datagram is larger than MTU"
- Each fragment is routed independently.
- A fragmented datagram may be further fragmented, if it encounters a network with a smaller MTU.
- Source/router is responsible for fragmentation of original datagram into the fragments.
- Only destination is responsible for reassembling the fragments into the original datagram.

Fields Related to Fragmentation & Reassembly

- Three fields in the IP header are used to manage fragmentation and reassembly:
- Identification
- Flags
- Fragmentation offset

Identification

- This field is used to identify to which datagram a particular fragment belongs to (so that fragments for different packets do not get mixed up).

- When a datagram is created, the source attaches the datagram with an identification number.
- When a datagram is fragmented, the value in the identification field is copied into all fragments.
- The identification number helps the destination in reassembling the datagram.

Flags

- This field has 3 bits.
- The first bit is not used.
- DF bit (Don't Fragment)
    - If DF=1, the router should not fragment the datagram. Then, the router discards the datagram.
        - If DF=0, the router can fragment the datagram.
        - MF bit (More Fragment)
        - If MF=1, there are some more fragments to come.
        - If MF=0, this is last fragment.

Fragmentation Offset

- This field identifies location of a fragment in a datagram.
- This field is the offset of the data in the original datagram.

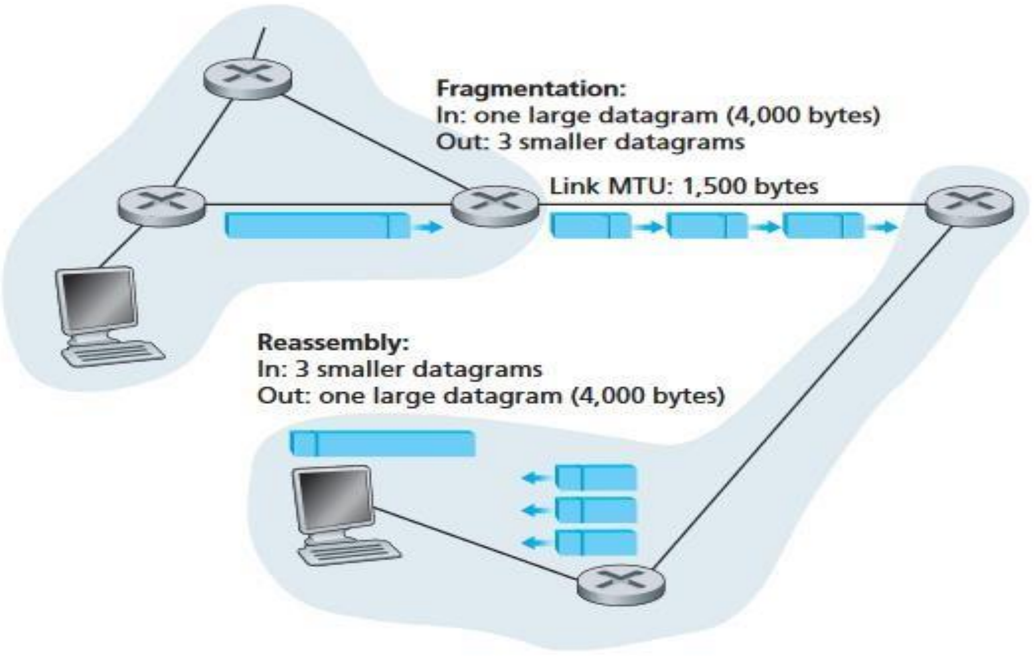
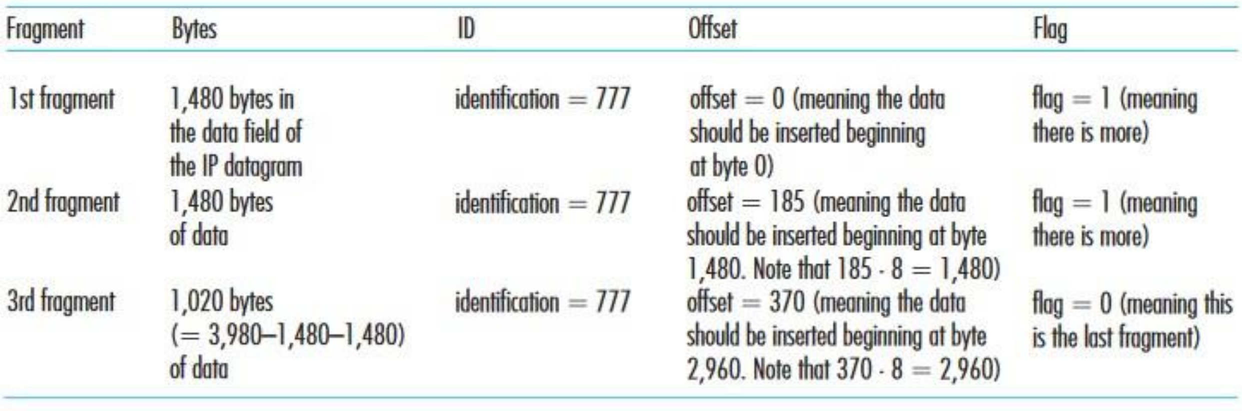


Figure 4.14 ♦ IP fragmentation and reassembly

- Figure 4.14 illustrates an 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 (each of which is also an IP datagram).
- Suppose that the original datagram is stamped with an identification number of 777.
- The characteristics of the three fragments are shown in Table 4.2.
- The values in Table 4.2 reflect the requirement that the amount of original payload data in all but the last fragment be a multiple of 8 bytes, and that the offset value be specified in units of 8-byte chunks.



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)

Table 4.2 ♦ IP fragments

IPv4 Addressing

- IP address is a numeric identifier assigned to each machine on the internet.
- IP address consists of two parts: network ID(NID) and host ID(HID).
- NID identifies the network to which the host is connected. All the hosts connected to the same network have the same NID.
- HID is used to uniquely identify a host on that network.
- HID is assigned by the network-administrator at the local site.
- NID for an organization may be assigned by the ISP (Internet Service Provider).
- IPv4 uses 32-bit addresses, i.e., approximately 4 billion addresses (2^{32}).
- IP addresses are usually written in dotted-decimal notation. The address is broken into four bytes. For example, an IP address of

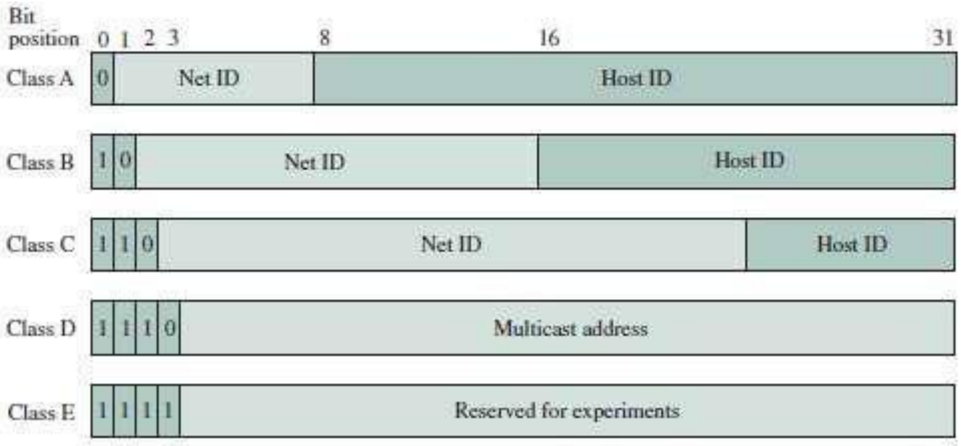
  10000000 10000111 01000100 00000101

  is written as 128.135.68.5

- IP address can be classified as
- Classful IP addressing &
- Classless IP addressing (CIDR- Classless Inter Domain Routing)

IPv4 Classful Addressing

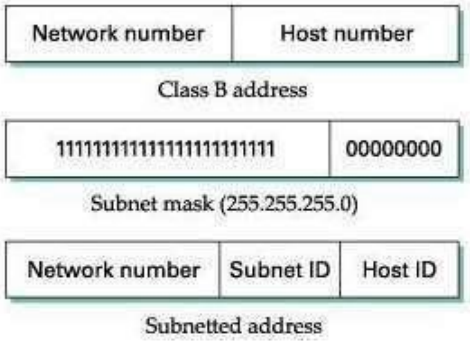
- In classful addressing, the address space is divided into five classes: A, B, C, D and E.
- IP address class is identified by MSBs in binary.
- Classes A, B and C are used for unicast addressing.
- Class D was designed for multicasting and class E is reserved.



Subnet Addressing

- Problem with classful addressing:
  - Consider an organization has a Class B address which can support about 64,000 hosts.
  - It will be a huge task for the network-administrator to manage all 64,000 hosts.
- Solution: Use subnet addressing.
- Subnetting reduces the total number of network-numbers by assigning a single network-number to many adjacent physical networks.
- Each adjacent physical network is referred to as subnet.
- All nodes on a subnet are configured with a subnet mask. For example: 255.255.255.0.
- The 1's in the subnet-mask represent the positions that refer to the network or subnet-numbers.
- The bitwise AND of IP address and its subnet mask gives the subnet number.
- Advantage: The subnet-addressing scheme is oblivious to the network outside the organization.

Inside the organization the network-administrator is free to choose any combination of lengths for the subnet & host ID fields.

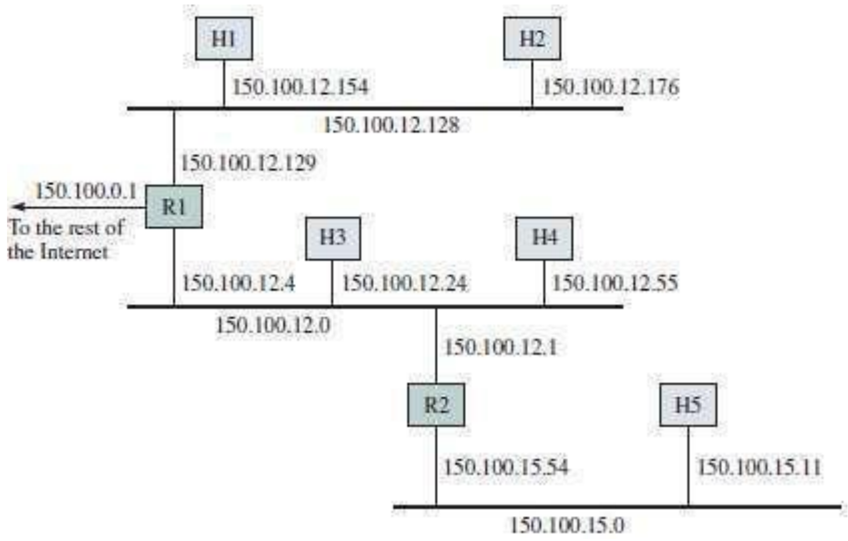


Class B address

Subnet mask (255.255.255.0)

Subnetted address

Question: If a packet with a destination IP address of 150.100.12.176 arrives at site from the outside network, which subnet should a router forward this packet to? Assume subnet mask is 255.255.255.128



Solution: The router can determine the subnet number by performing a binary AND between the subnet mask and the IP address.

IP address:	10010110 01100100 00001100 10110000(150.100.12.176)
Subnet mask:	11111111 11111111 11111111 10000000(255.255.255.128)
Subnet number:	10010110 01100100 00001100 10000000(150.100.12.128)

This number (150.100.12.128) is used to forward the packet to the correct subnet work inside the organization.

CIDR (Classless Interdomain Routing)

- Problem with classful IP addressing:
    - Consider an organization needs about 500 hosts.
    - Obviously, the organization will get a Class B license, even though it has far fewer than 64,000 hosts.
    - At most, over 64,000 addresses can go unused.
    - This results in inefficient usage of the available address-space.

Solution: Use CIDR (Classless Inter Domain Routing).

- A single IP address can be used to designate many unique IP addresses. This is called supernetting.
- A CIDR IP address looks like a normal IP address except that the address ends with a slash followed by a number, called the IP network prefix.
    - CIDR addresses
    - reduce the size of routing-tables and
    - make more IP addresses available within organizations.

Obtaining a Block of Addresses

- To obtain a block of IP addresses for use within an organization's subnet, a network-administrator contacts the ISP.
- IP addresses are managed under the authority of the ICANN.
- The responsibility of the ICANN (Internet Corporation for Assigned Names and

Numbers):

- to allocate IP addresses,
- to manage the DNS root servers.
- to assign domain names and resolve domain name disputes.
- to allocate addresses to regional Internet registries.

Obtaining a Host Address: DHCP

- Two ways to assign an IP address to a host:
  1) Manual Configuration
     Operating systems allow system-administrator to manually configure IP address.
  2) Dynamic Host Configuration Protocol (DHCP)
     DHCP enables auto-configuration of IP address to host.

DHCP Protocol

- DHCP enables auto-configuration of IP address to host.
- DHCP assigns dynamic IP addresses to devices on a network.
- Dynamic address allocation is required when a host moves from one network to another or when a host is connected to a network for the first time.
- Because of DHCP's ability to automate the network-related aspects of connecting a host into a network, it is often referred to as a plug-and-play protocol.
- DHCP is a client-server protocol.
- A client is typically a newly arriving host wanting to obtain network configuration information, including an IP address for itself.
- In the simplest case, each subnet will have a DHCP server.
- If no server is present on the subnet, a DHCP relay agent (typically a router) that knows the address of a DHCP server for that network is needed.
- Figure 4.20 shows a DHCP server attached to subnet 223.1.2/24, with the router serving as the relay agent for arriving clients attached to subnets 223.1.1/24 and 223.1.3/24.

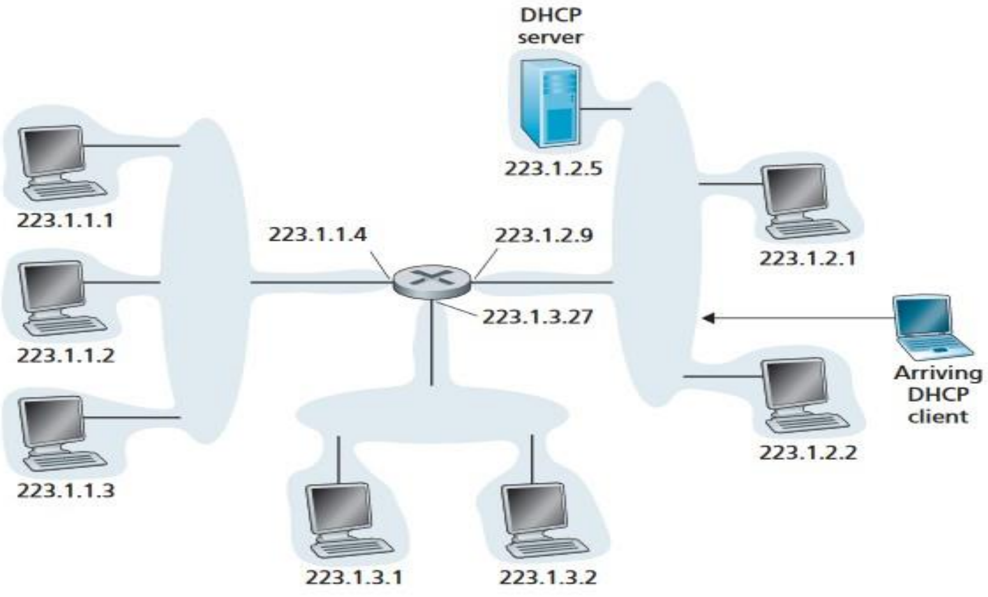


Figure 4.20 ♦ DHCP client-server scenario

- DHCP protocol is a four-step process, as shown in Figure 4.21 for the network setting shown in Figure 4.20.
- In this figure, yiaddr(as in “your Internet address”) indicates the address being allocated to the newly arriving client. The four steps are:

DHCP server discovery.

- The first task of a newly arriving host is to find a DHCP server with which to interact.
- This is done using a DHCP discover message, which a client sends within a UDP packet to port 67.
- The UDP packet is encapsulated in an IP datagram.
- DHCP client creates an IP datagram containing its DHCP discover message along with the broadcast destination IP address of 255.255.255.255 and a “this host” source IP address of 0.0.0.0. The DHCP client passes the IP datagram to the link layer, which then broadcasts this frame to all nodes attached to the subnet.

DHCP server offer(s).

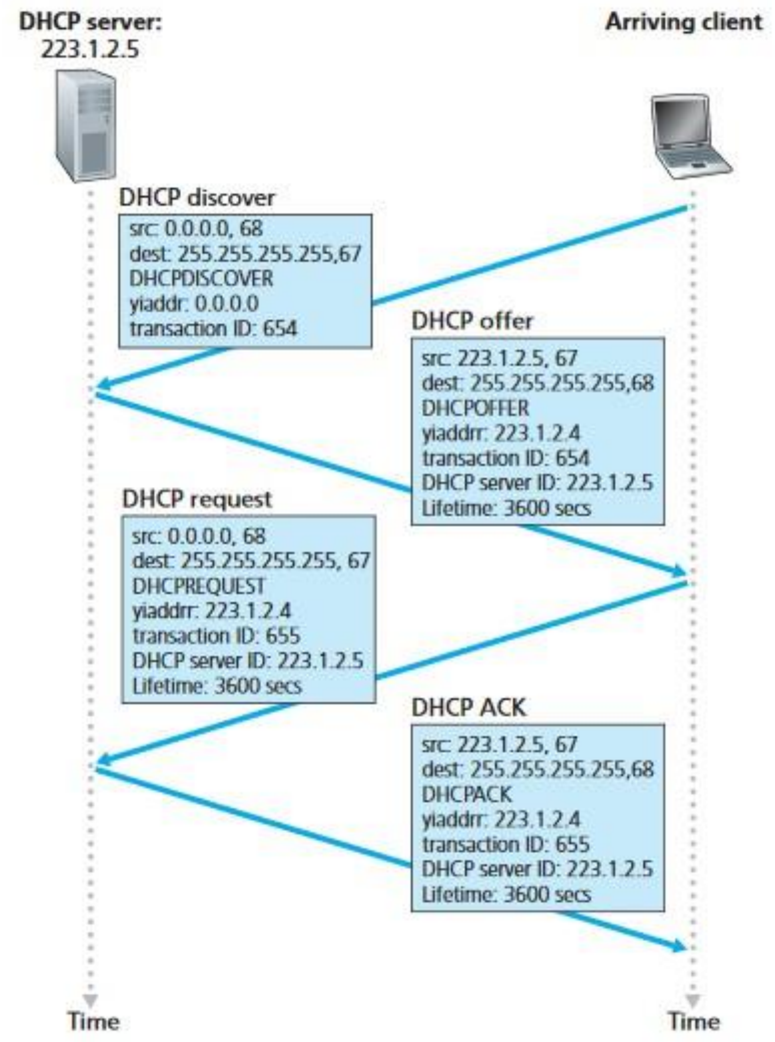
- A DHCP server receiving a DHCP discover message responds to the client with a DHCP offer message that is broadcast to all nodes on the subnet, again using the IP broadcast address of 255.255.255.255.
- DHCP server broadcasts DHCPOFFER message containing
    1. Client's IP address
    2. Network mask and
    3. IP address lease time (i.e. the amount of time for which the IP address will be valid).

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, echoing back the configuration parameters.

DHCP ACK.

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



4.21 ♦ DHCP client-server interaction

NAT

- Network Address Translation (NAT) enables hosts to use Internet without the need to have globally unique addresses.
- NAT enables organization to have a large set of addresses internally and one address externally.
- The organization must have single connection to the Internet through a NAT-enabled router.
- NAT allows a single device (such as a router) to act as an agent b/w internet (or "public network") and local (or "private") network.
- This means only a single, unique IP address is required to represent an entire group of computers.
- Figure shows the operation of a NAT-enabled router.

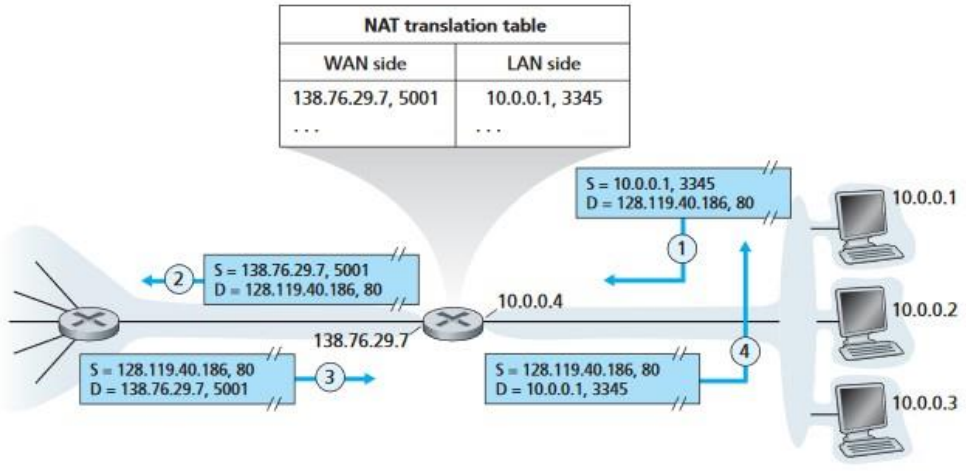


Figure 4.22 ♦ Network address translation

- The private addresses only have meaning to devices within a given network.
- The NAT-enabled router does not look like a router to the outside world.
- Instead, the NAT-enabled router behaves to the outside world as a single device with a single IP address.

In Figure

1) All traffic leaving the home-router for the Internet has a source-address of 138.76.29.7.

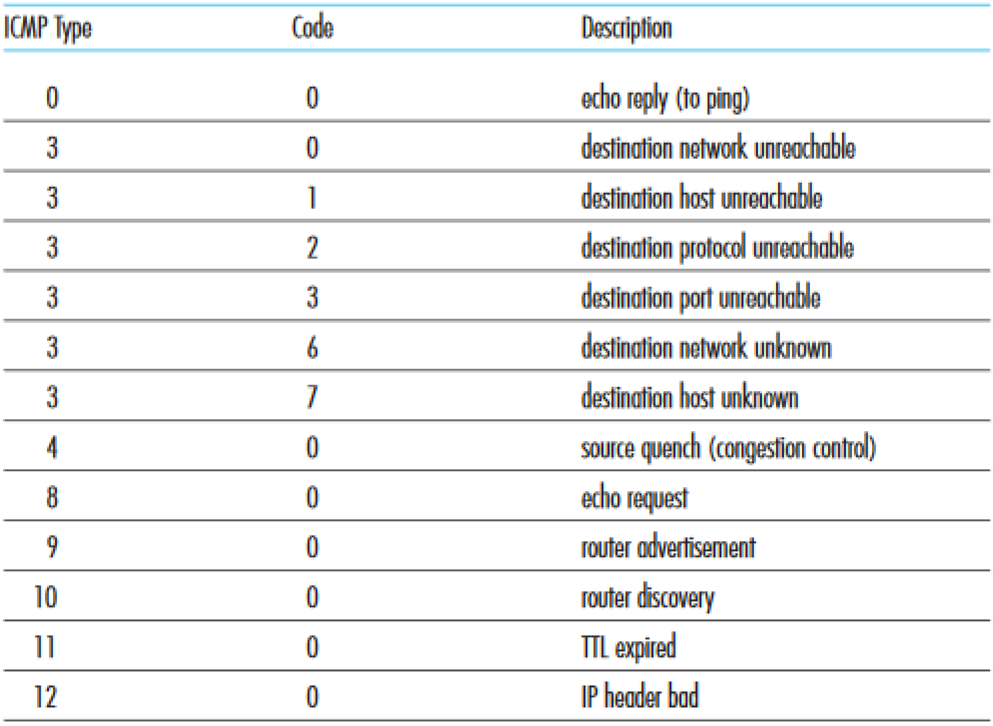
2) All traffic entering the home-router must have a destination-address of 138.76.29.7.
    - The NAT-enabled router is hiding the details of the home-network from the outside world.
    - At the NAT router, NAT translation-table includes
        1) Port numbers and
        2) IP addresses.

- IETF community is against the use of NAT. This is because of following reasons:
    1) They argue, port numbers are to be used for addressing processes, not for addressing hosts.
    2) They argue routers are supposed to process packets only up to layer 3.
    3) They argue the NAT protocol violates the end-to-end argument.
    4) They argue, we should use IPv6 to solve the shortage of IP addresses.
    5) NAT interferes with P2P applications. If Peer B is behind NAT, Peer B cannot act as a server.

ICMP (Internet Control Message Protocol)

- ICMP is a network-layer protocol. (ICMP Internet Control Message Protocol).
- This is used to handle error and other control messages.
- Main responsibility of ICMP: To report errors that occurs during the processing of the datagram.
- ICMP does not correct errors; ICMP simply reports the errors to the source.

- 12 types of ICMP messages are defined as shown in Fig 4.23



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

Figure 4.23 ♦ ICMP message types

- Each ICMP message type is encapsulated in an IP packet.

1) Destination Unreachable (Type=3)

- This message is related to problem reaching the destinations.
- This message uses different codes (0 to 15) to define type of error-message.
- Possible values for code field:

  Code 0 = network unreachable Code 1 = host
  unreachable Code 2 = protocol unreachable
  Code 3 = port
  unreachable

2) Source Quench (Type=4)

- The main purpose is to perform congestion control.
- This message
  - informs the sender that network has encountered congestion & datagram has been dropped.
  - informs the sender to reduce its transmission-rate.

3) Echo Request & Echo Reply (Type=8 & Type=0)

- These messages are used to determine whether a remote-host is alive.
- A source sends an echo request-message to destination;

- If the destination is alive, the destination responds with an echo reply message.
- Type=8 is used for echo request; type=0 is used for echo reply.
- These messages can be used in two debugging tools: ping and traceroute.

i) Ping

□ The ping program can be used to find if a host is alive and
□ responding. The source-host sends ICMP echo-request-
□ messages.
The destination, if alive, responds with ICMP echo-reply messages.

ii) Traceroute

□ The traceroute program can be used to trace the path of a packet from
□ source to destination. It can find the IP addresses of all the routers that are
□ visited along the path.
The program is usually set to check for the maximum of 30 hops (routers) to be visited.

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)

- **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.
  - 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.
- **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

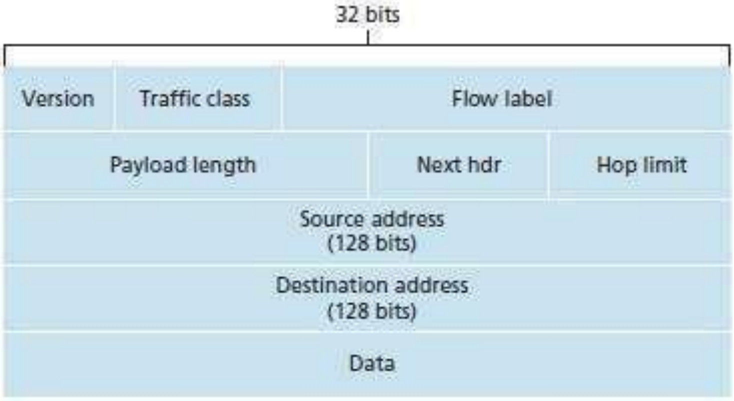


Figure 3.18: IPv6 datagram format

The following fields are defined in IPv6:

1) Version

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

2) Traffic Class

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

3) Flow Label

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

4) Payload Length

□ This field shows the length of the IPv6 payload.

5) 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.

6) 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.

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

8) 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).

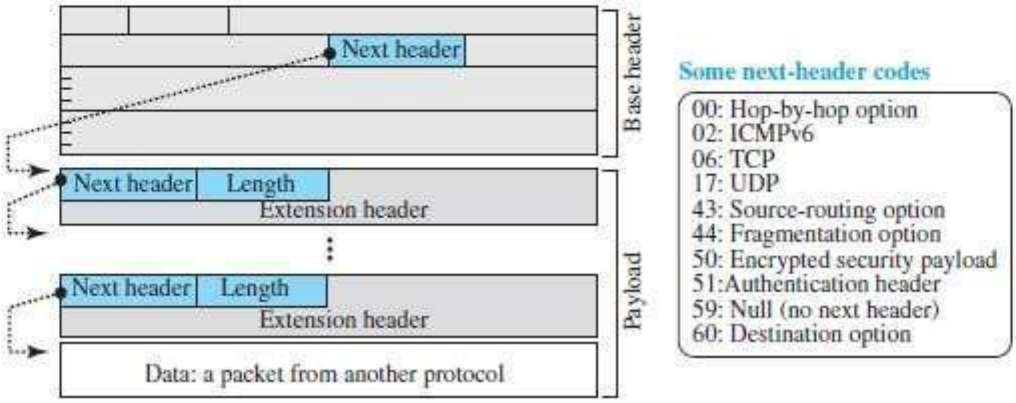


Figure 3.19: Payload in IPv6 datagram

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.
- 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.

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

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.

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.

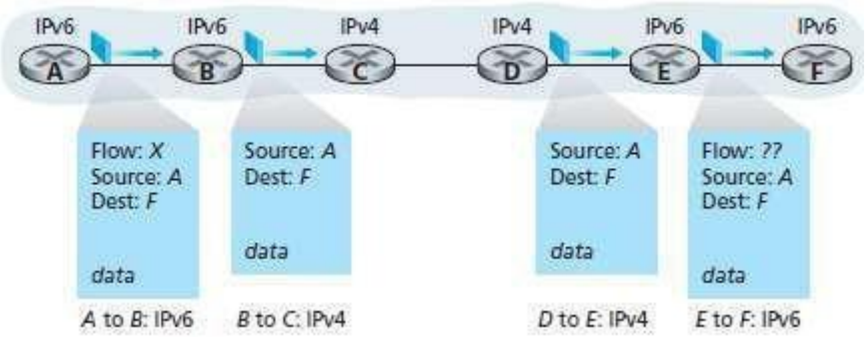


Figure 3.20: A dual-stack approach

- 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 ode-E, the IPv6 datagram is extracted from the data field of the IPv4 datagram.

5) Finally, IPv6-capable ode-E forwards an IPv6 datagram to IPv6-capable Node-F.

- **Disadvantage:** During transition from IPv6 to IPv4, few IPv6-specific fields will be lost.

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

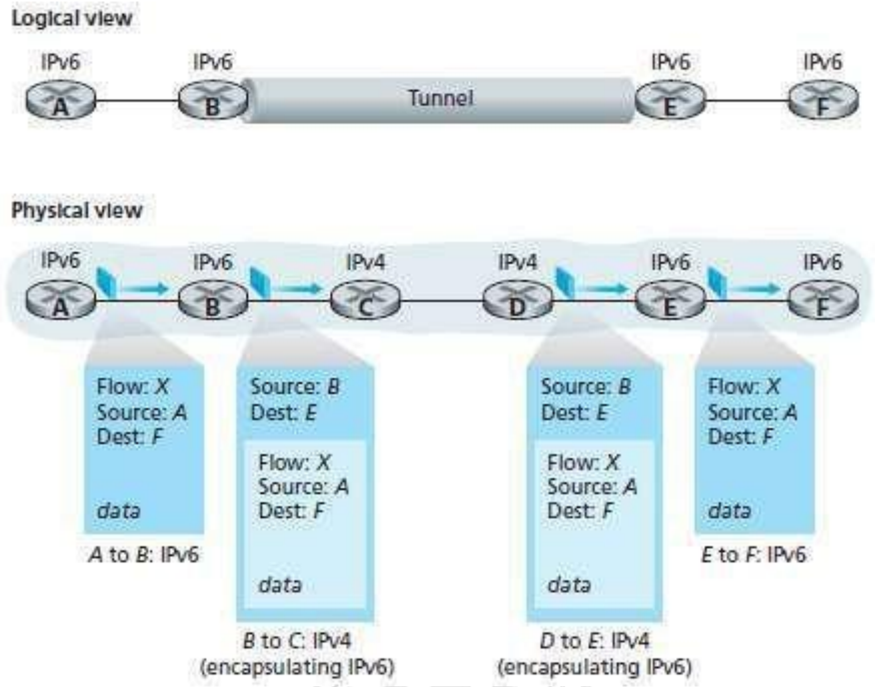


Figure 3.21: Tunneling

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.

□ 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.

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 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.

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.

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.

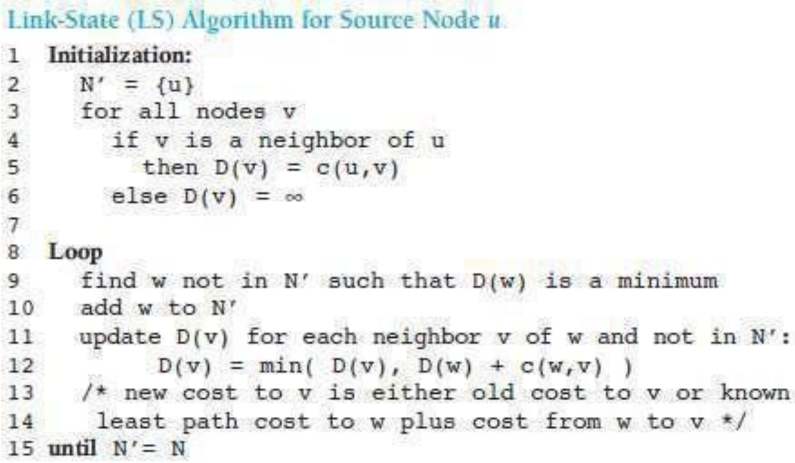
Load Insensitive Algorithm

- 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

LS Routing Algorithm 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.

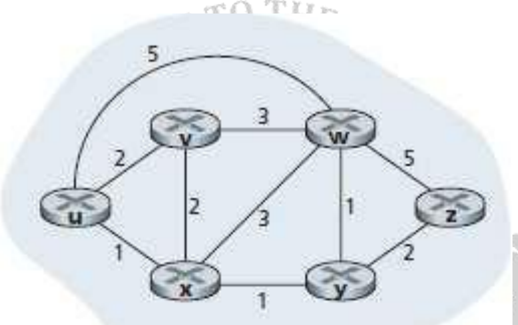


Figure 3.22: Abstract graph model of a computer network

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

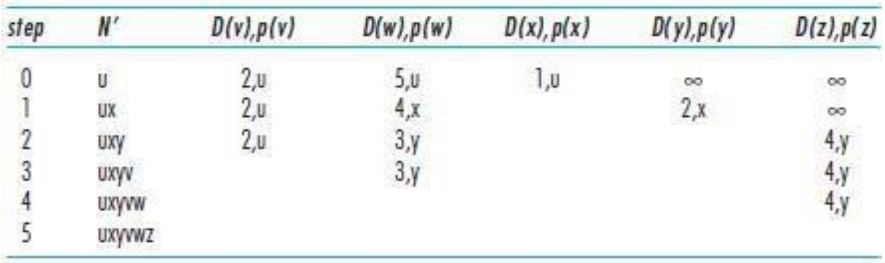
→ add y to the set N' so that N' now contains u, x, and y.

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.

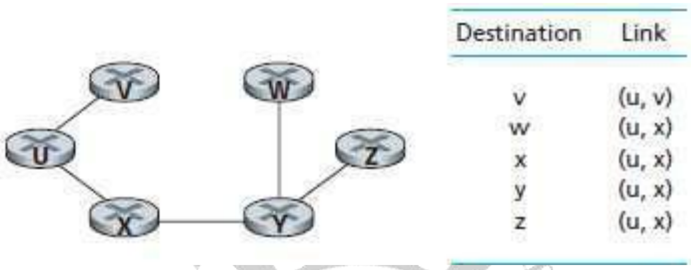
- A tabular summary of the algorithm’s computation is shown in Table 3.5.



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

DV Routing Algorithm 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

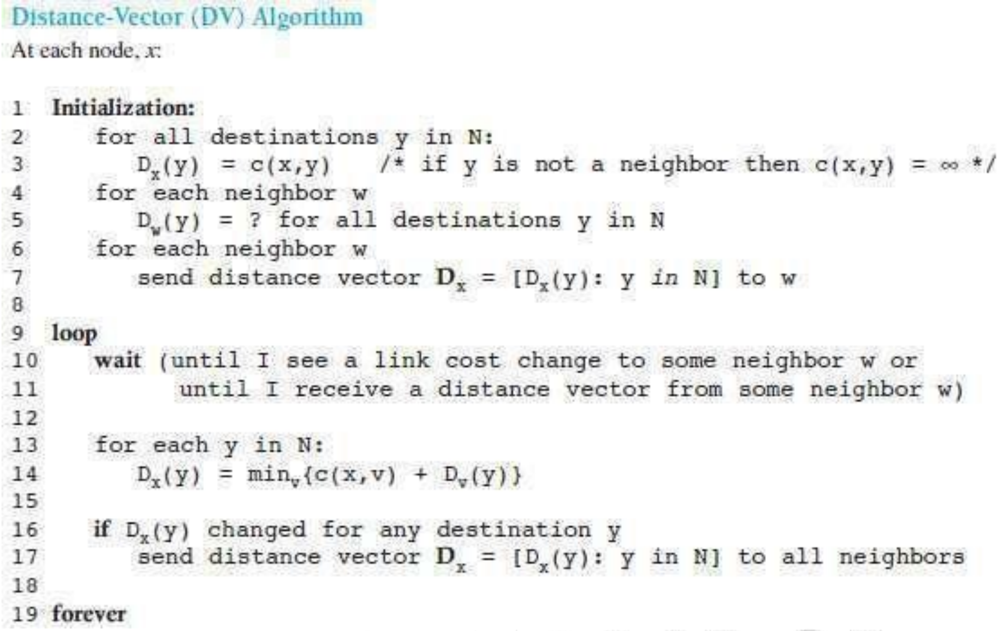
ii) Node x’s distance vector, that is, Dx = [Dx(y): y in N], containing x’s estimate of its cost to all destinations y in N.

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:

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.

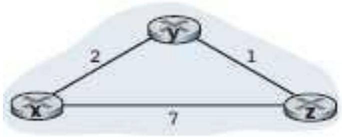


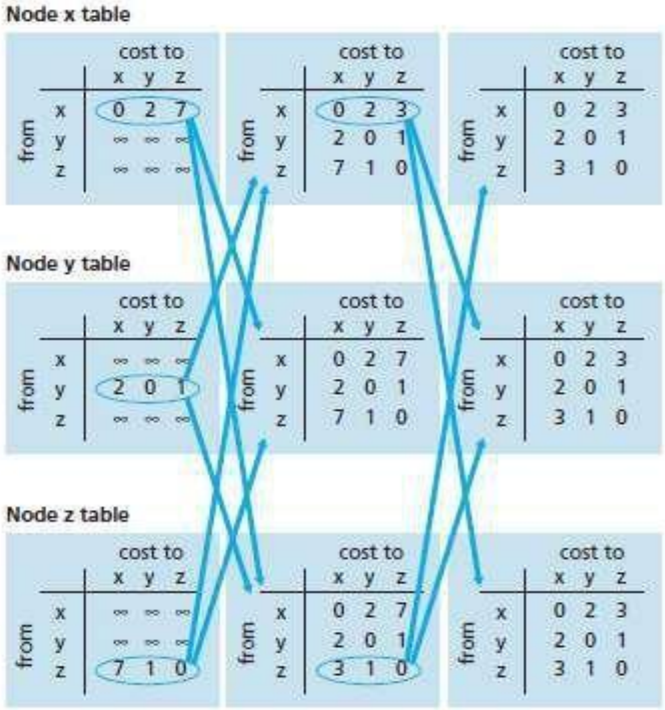
Distance-Vector (DV) Algorithm

At each node, x:

```
Initialization:
   for all destinations y in N:
      $D_x(y) = c(x,y)$    /* if y is not a neighbor then $c(x,y) = \infty$ */
   for each neighbor w
      $D_w(y)$ = ? for all destinations y in N
   for each neighbor w
      send distance vector $\mathbf{D}_x = [D_x(y)$: 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:
      $D_x(y) = \min_v\{c(x,v) + D_v(y)\}$

   if $D_x(y)$ changed for any destination y
      send distance vector $\mathbf{D}_x = [D_x(y)$: y in N] to all neighbors

forever
```

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





Node x table

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

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

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

Node y table

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

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

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

Node z table

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

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

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

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 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
- 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.

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 mode talks to only its directly connected neighbours.	Each node talks with all other nodes(via broadcast)

Hierarchical Routing

- Two problems of a simple routing-algorithm:
  - 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.

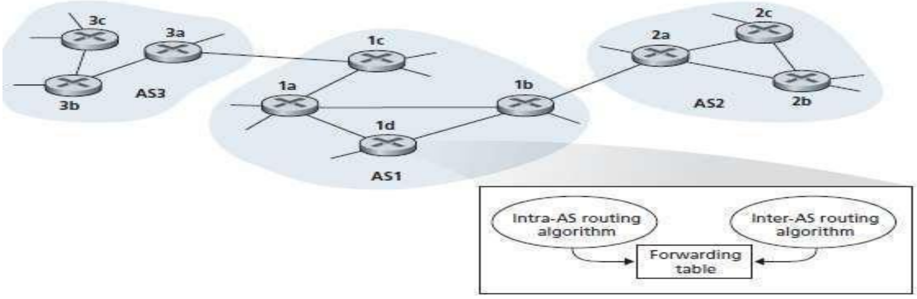


Figure 3.25: An example of interconnected autonomous-systems

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.

Intra-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.

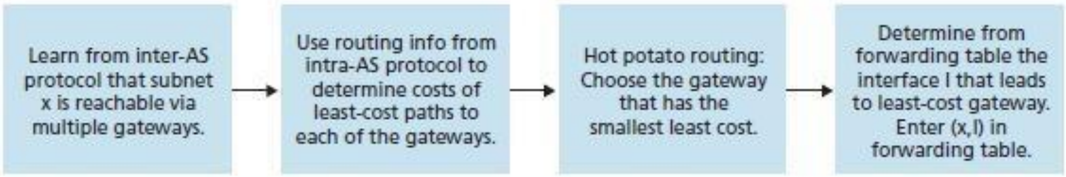

Learn from inter-AS protocol that subnet x is reachable via multiple gateways.
Use routing info from intra-AS protocol to determine costs of least-cost paths to each of the gateways.
Hot potato routing: Choose the gateway that has the smallest least cost.
Determine from forwarding table the interface I that leads to least-cost gateway. Enter (x,I) in forwarding table.

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

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.

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.

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.

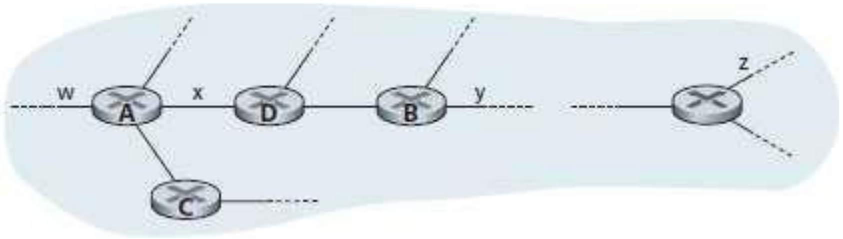
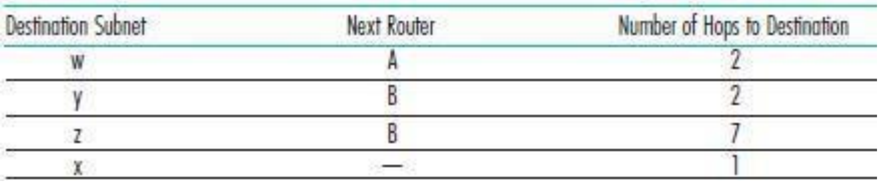


Figure 3.27: A portion of an autonomous-system

- 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.

  □ 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.

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.
he primary role of the backbone-area is to route traffic between the other areas in the AS.

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.

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.
- 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.

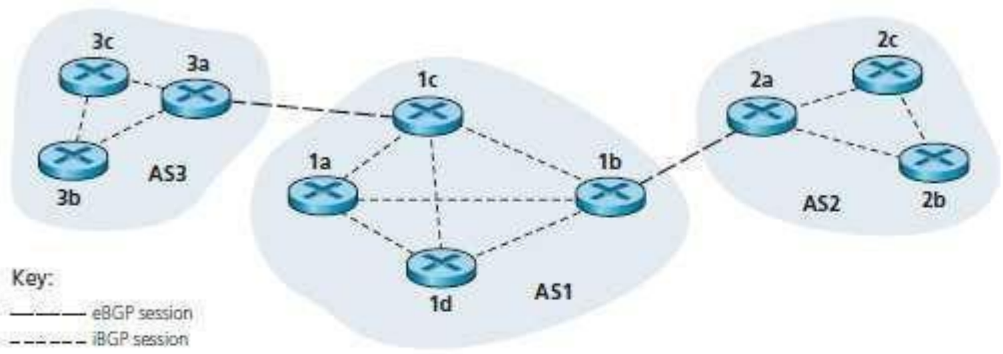


Figure 3.29: eBGP and iBGP sessions

- 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.

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

□ his 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.

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.

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.
- 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.

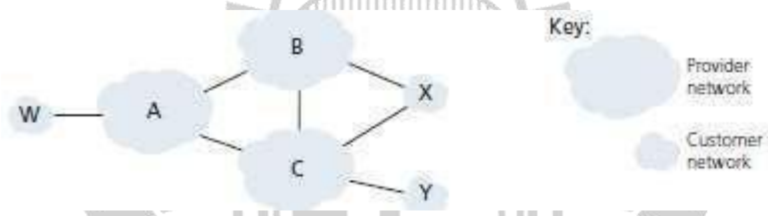


Figure 3.30: A simple BGP scenario

Broadcast & Multicast Routing Broadcast Routing Algorithms

- Broadcast-routing means delivering a packet from a source-node to all other nodes in the network.

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:

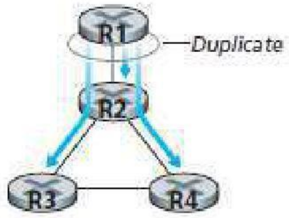
  □ If source is connected to the n/w via single link, then N copies of packet will traverse this link.

  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.

2) Not suitable for Unicast Routing

□ t is not good idea to depend on the unicast routing infrastructure to achieve broadcast.



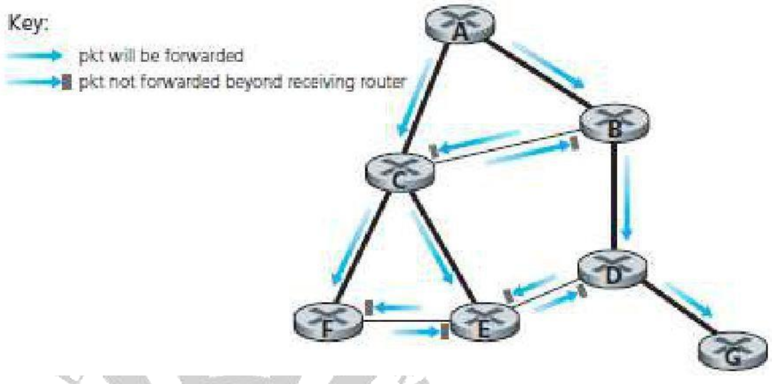


Figure 3.31: Duplicate creation/transmission

Figure 3.32: Reverse path forwarding

Uncontrolled Flooding

- The source-node sends a copy of the packet to all the neighbours.
- When a node receives a broadcast-packet, the node duplicates & forwards packet to all neighbours.
- 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."

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.

- 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.

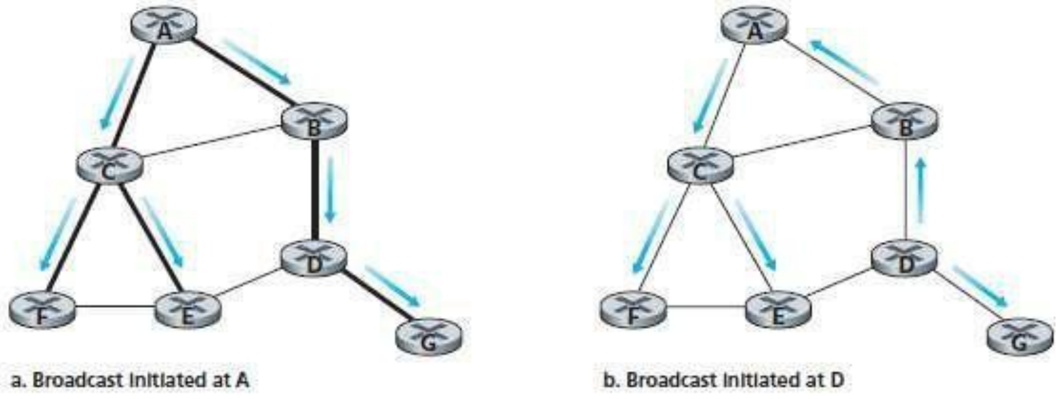
- Such a packet will be dropped. This is because

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

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.

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.

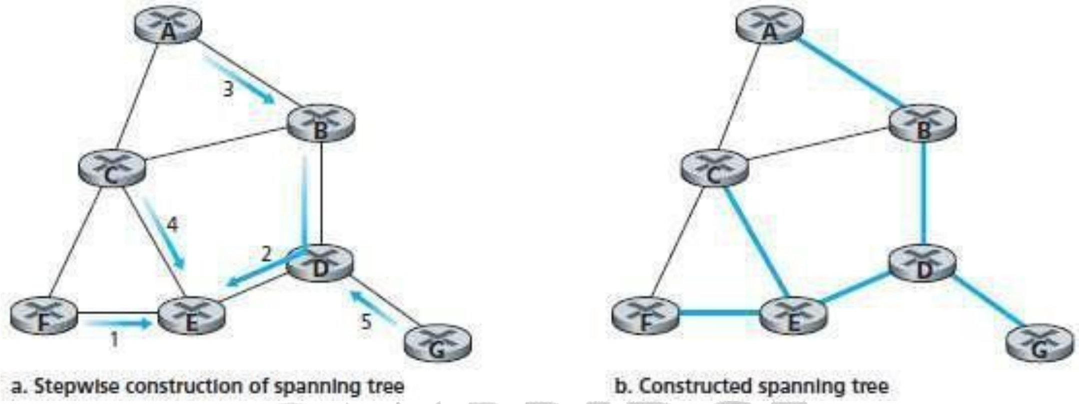

A
B
C
D
E
F
G
1
2
3
4
5
a. Stepwise construction of spanning tree
A
B
C
D
E
F
G
b. Constructed spanning tree

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

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.

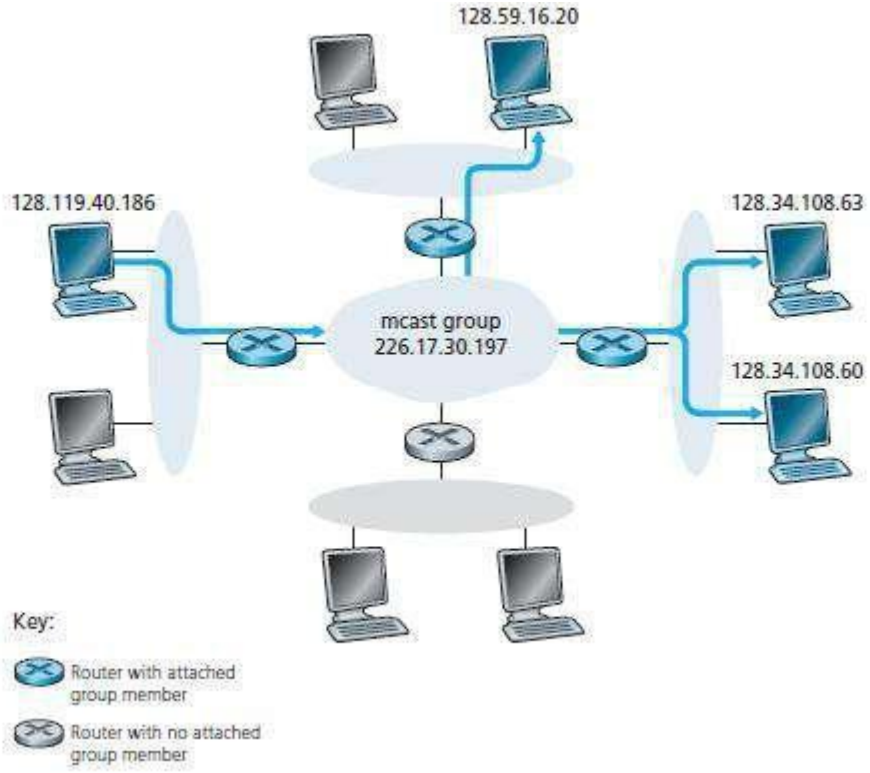


Figure 3.35: The multicast group: A datagram addressed to the group is delivered to all members of the multicast group

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.

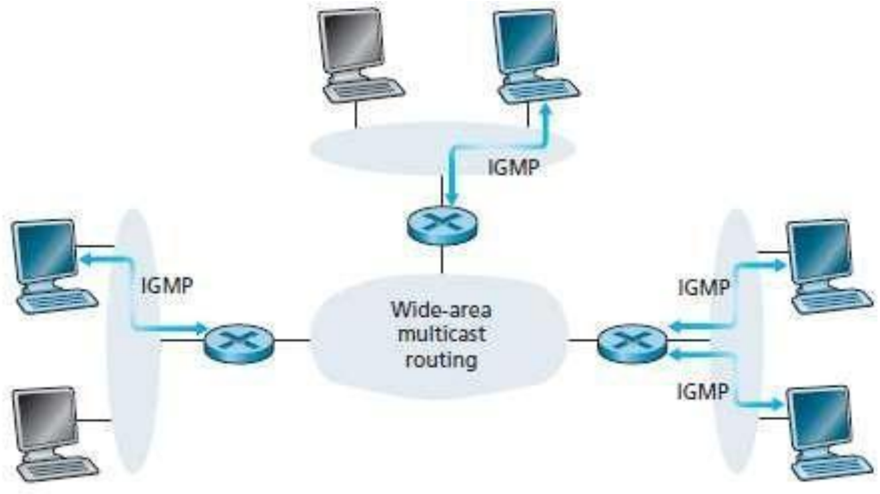


Figure 3.36: The two components of network-layer multicast in the Internet: IGMP and multicast-routing protocols

- 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

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.

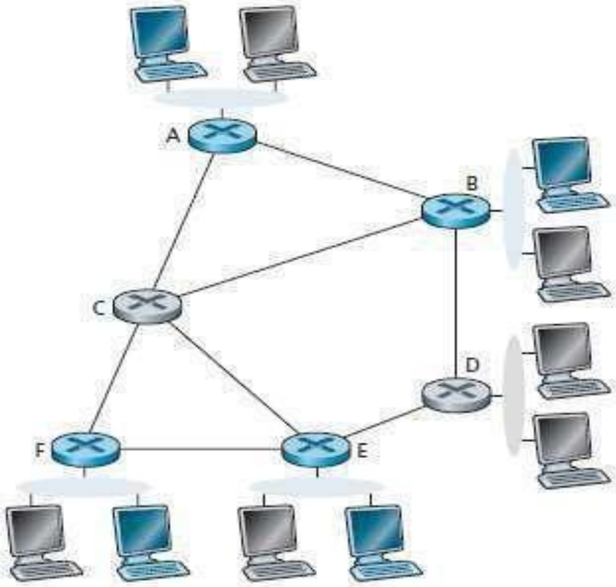


Figure 3.37: Multicast hosts, their attached routers, and other routers

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.