

UNIT II NETWORK LAYER PROTOCOLS

Network Layer – IPv4 Addressing – Network Layer Protocols (IP, ICMP and Mobile IP) Unicast and Multicast Routing – Intra domain and Inter domain Routing Protocols – IPv6 Addresses – IPv6 – Datagram Format - Transition from IPv4 to IPv6.

Network Layer

The network layer in the TCP/IP protocol suite is responsible for the host-to-host delivery of datagrams. It provides services to the transport layer and receives services from the data-link layer.

NETWORK-LAYER SERVICES

1. Packetizing

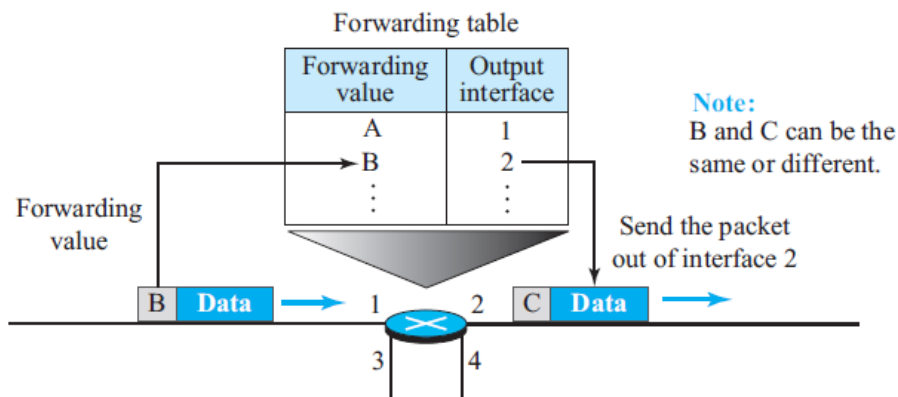
- The first duty of the network layer is **packetizing**: encapsulating the payload (data received from upper layer) in a network-layer packet at the source and decapsulating the payload from the network-layer packet at the destination.
- The source host receives the payload from an upper-layer protocol, adds a header that contains the source and destination addresses and some other information that is required by the network-layer protocol (as discussed later) and delivers the packet to the data-link layer.
- The destination host receives the network-layer packet from its data-link layer, decapsulates the packet, and delivers the payload to the corresponding upper-layer protocol.
- If the packet is fragmented at the source or at routers along the path, the network layer is responsible for waiting until all fragments arrive, reassembling them, and delivering them to the upper-layer protocol.

2. Routing

- The network layer is responsible for routing the packet from its source to the destination.
- There is more than one route from the source to the destination. The network layer is responsible for finding the best one among these possible routes. The network layer needs to have some specific strategies for defining the best route.

3. Forwarding

- Forwarding can be defined as the action applied by each router when a packet arrives at one of its interfaces. The decision-making table a router normally uses for applying this action is sometimes called the forwarding table and sometimes the routing table.
- When a router receives a packet from one of its attached networks, it needs to forward the packet to another attached network (in unicast routing) or to some attached networks (in multicast routing). To make this decision, the router uses a piece of information in the packet header, which can be the destination address or a label, to find the corresponding output interface number in the forwarding table.

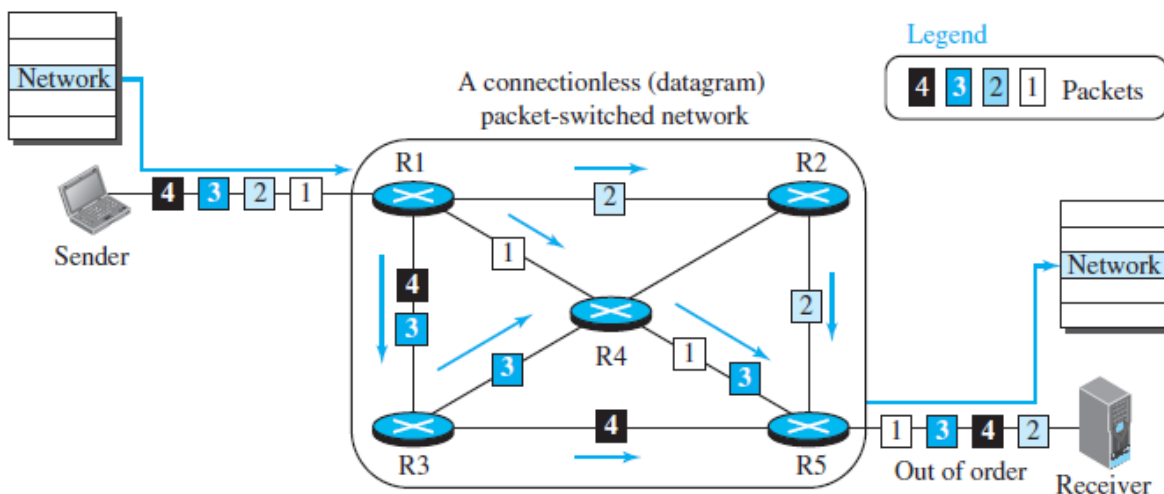


PACKET SWITCHING

- At the network layer, a message from the upper layer is divided into manageable packets and each packet is sent through the network. The source of the message sends the packets one by one; the destination of the message receives the packets one by one.
- The destination waits for all packets belonging to the same message to arrive before delivering the message to the upper layer. The connecting devices in a packet-switched network still need to decide how to route the packets to the final destination.
- Packet-switched network can use two different approaches to route the packets: **the datagram approach and the virtual circuit approach.**

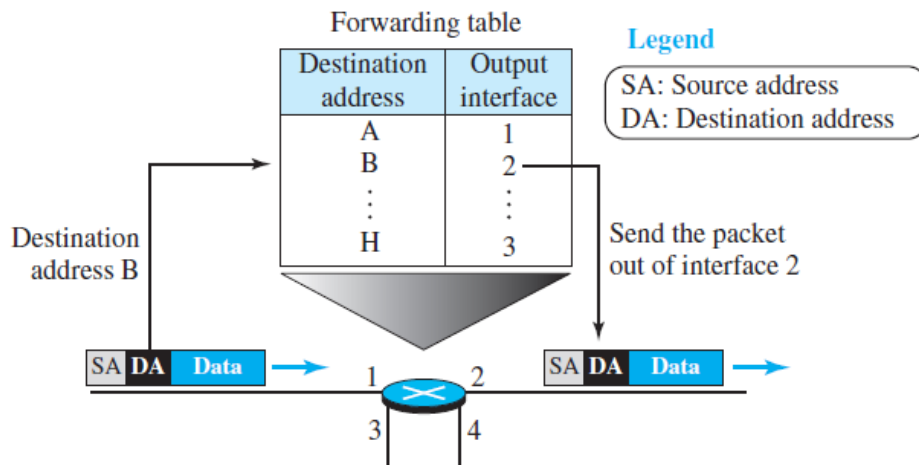
Datagram Approach: Connectionless Service

- When the network layer provides a connectionless service, each packet traveling in the Internet is an independent entity; there is no relationship between packets belonging to the same message.
- The switches in this type of network are called *routers*.
- A packet belonging to a message may be followed by a packet belonging to the same message or to a different message. A packet may be followed by a packet coming from the same or from a different source.



A connectionless packet-switched network

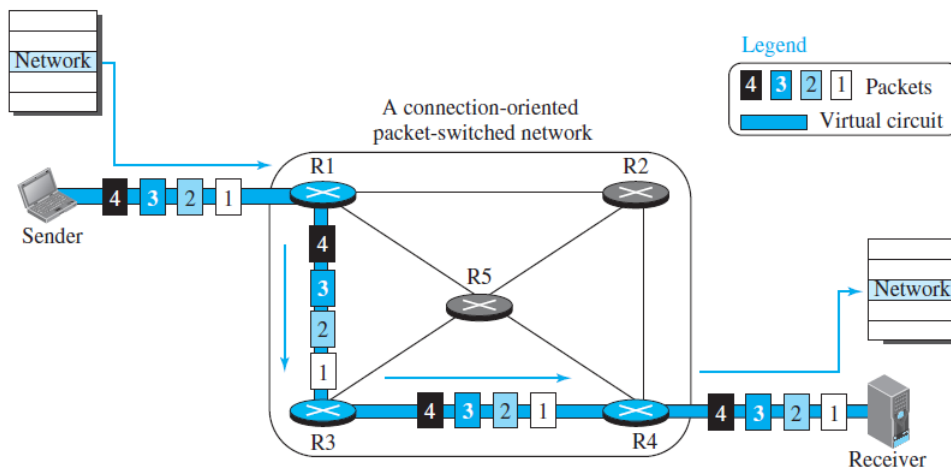
The router in this case routes the packet based only on the destination address. In the datagram approach, the forwarding decision is based on the destination address of the packet.



Forwarding Process

Virtual-Circuit Approach: Connection-Oriented Service

- In a connection-oriented service (also called *virtual-circuit approach*), there is a relationship between all packets belonging to a message.
- Before all datagrams in a message can be sent, a virtual connection should be set up to define the path for the datagrams. After connection setup, the datagrams can all follow the same path.
- In this type of service, not only must the packet contain the source and destination addresses, it must also contain a flow label, a virtual circuit identifier that defines the virtual path the packet should follow.



A virtual-circuit packet-switched network

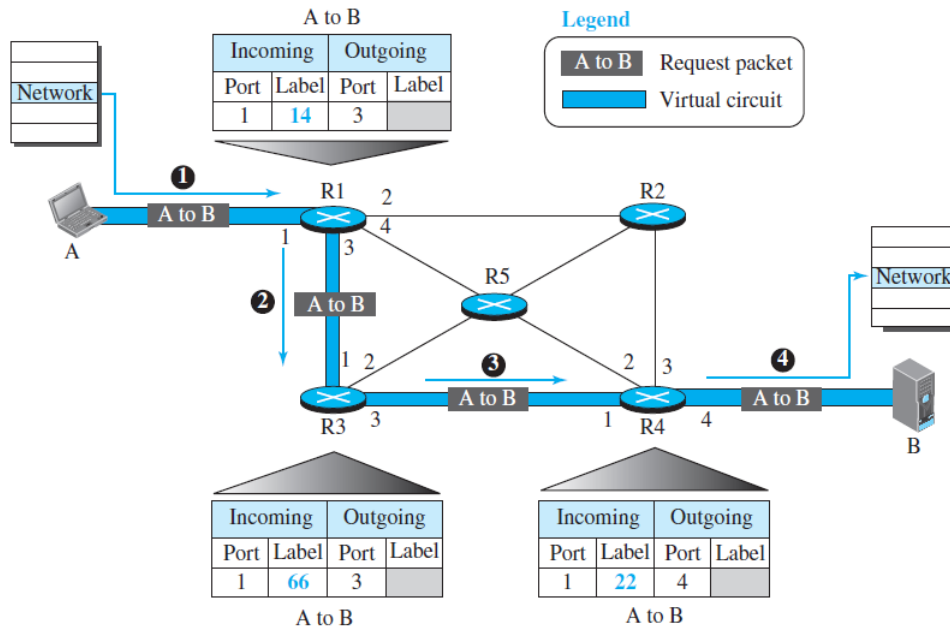
- To create a connection-oriented service, a three-phase process is used: setup, data transfer, and teardown.
- In the setup phase, the source and destination addresses of the sender and receiver are used to make table entries for the connection-oriented service.
- In the teardown phase, the source and destination inform the router to delete the corresponding entries.
- *Data transfer occurs between these two phases.*

Setup Phase

Suppose source A needs to create a virtual circuit to destination B. Two auxiliary packets need to be exchanged between the sender and the receiver: the request packet and the acknowledgment packet.

Request packet

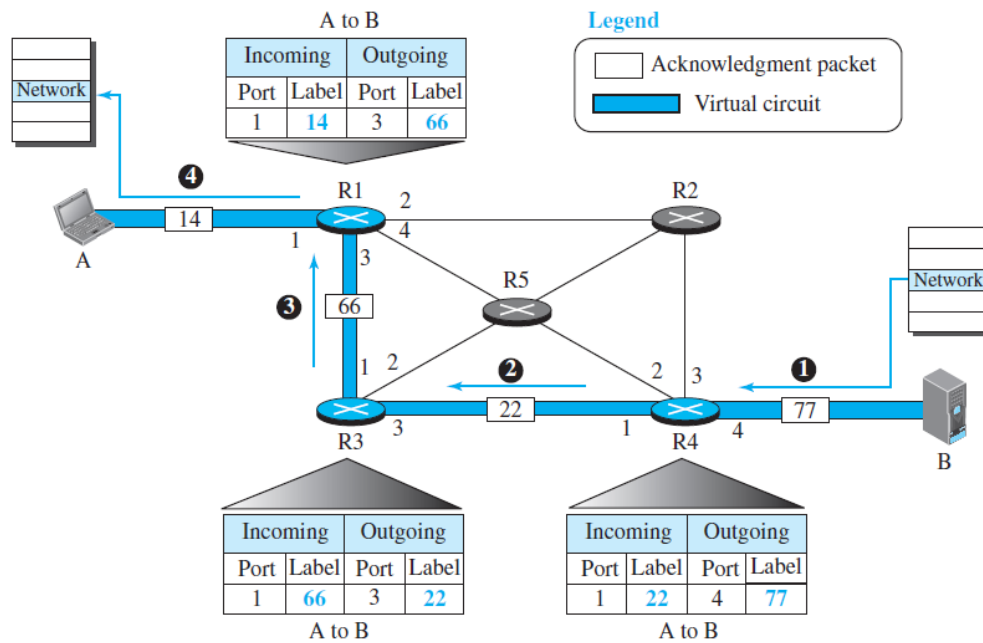
A request packet is sent from the source to the destination.



1. Source A sends a request packet to router R1.
2. Router R1 receives the request packet. It knows that a packet going from A to B goes out through port 3. The router creates an entry in its table for this virtual circuit, but it is only able to fill three of the four columns. The router assigns the incoming port (1) and chooses an available incoming label (14) and the outgoing port (3). It does not yet know the outgoing label, which will be found during the acknowledgment step. The router then forwards the packet through port 3 to router R3.
3. Router R3 receives the setup request packet. The same events happen here as at router R1; three columns of the table are completed: in this case, incoming port (1), incoming label (66), and outgoing port (3).
4. Router R4 receives the setup request packet. Again, three columns are completed: incoming port (1), incoming label (22), and outgoing port (4).
5. Destination B receives the setup packet, and if it is ready to receive packets from A, it assigns a label to the incoming packets that come from A, in this case 77, as shown in Fig. This label lets the destination know that the packets come from A, and not from other sources.

Acknowledgment Packet

A special packet, called the acknowledgment packet, completes the entries in the switching tables. Figure shows the process.



1. The destination sends an acknowledgment to router R4. The acknowledgment carries the global source and destination addresses so the router knows which entry in the table is to be completed. The packet also carries label 77, chosen by the destination as the incoming label for packets from A. Router R4 uses this label to complete the outgoing label column for this entry. Note that 77 is the incoming label for destination B, but the outgoing label for router R4.
2. Router R4 sends an acknowledgment to router R3 that contains its incoming label in the table, chosen in the setup phase. Router R3 uses this as the outgoing label in the table.
3. Router R3 sends an acknowledgment to router R1 that contains its incoming label in the table, chosen in the setup phase. Router R1 uses this as the outgoing label in the table.
4. Finally router R1 sends an acknowledgment to source A that contains its incoming label in the table, chosen in the setup phase.
5. The source uses this as the outgoing label for the data packets to be sent to destination B.

Data-Transfer Phase

The second phase is called the data-transfer phase. After all routers have created their forwarding table for a specific virtual circuit, then the network-layer packets belonging to one message can be sent one after another.

Teardown Phase

In the teardown phase, source A, after sending all packets to B, sends a special packet called a teardown packet. Destination B responds with a confirmation packet. All routers delete the corresponding entries from their tables.

IPv4 ADDRESSES

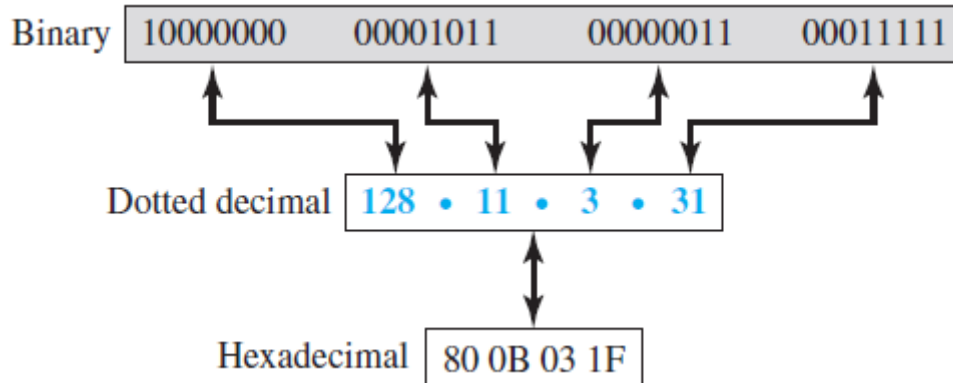
An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a host or a router to the Internet. The IP address is the address of the connection, not the host or the router, because if the device is moved to another network, the IP address may be changed.

Address Space

An **address space** is the total number of addresses used by the protocol. IPv4 uses 32-bit addresses, which means that the address space is 2^{32} or 4,294,967,296 (more than four billion).

Notation

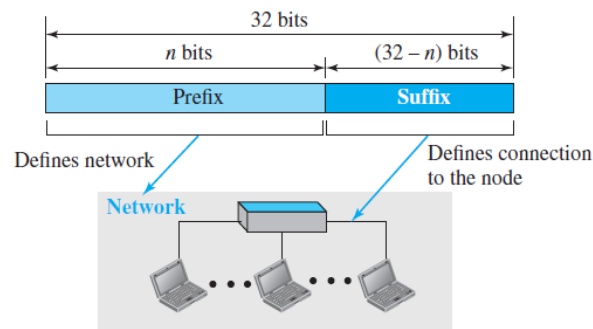
There are three common notations to show an IPv4 address: binary notation (base 2), dotted-decimal notation (base 256), and hexadecimal notation (base 16).



In binary notation, an IPv4 address is displayed as 32 bits. To make the IPv4 address more compact and easier to read, it is usually written in decimal form with a decimal point (dot) separating the bytes. This format is referred to as dotted-decimal notation. Each number in the dotted-decimal notation is between 0 and 255. Each hexadecimal digit is equivalent to four bits. This means that a 32-bit address has 8 hexadecimal digits.

Hierarchy in Addressing

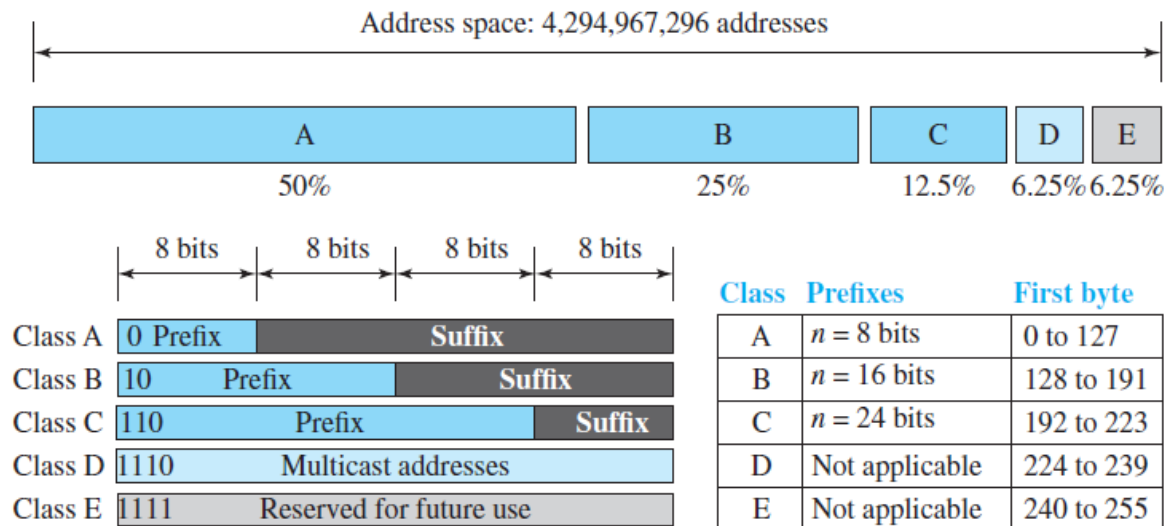
A 32-bit IPv4 address is hierarchical, and divided into two parts. The first part of the address, called the *prefix*, defines the network; the second part of the address, called the *suffix*, defines the node.



A prefix can be fixed length or variable length.

Classful Addressing

The whole address space was divided into five classes (class A, B, C, D, and E), as shown in Figure.



- In class A, the network length is 8 bits, but since the first bit, which is 0, defines the class, we can have only seven bits as the network identifier. This means there are only $2^7 = 128$ networks in the world that can have a class A address.
- In class B, the network length is 16 bits, but since the first two bits, which are $(10)_2$, define the class, we can have only 14 bits as the network identifier. This means there are only $2^{14} = 16,384$ networks in the world that can have a class B address.
- All addresses that start with $(110)_2$ belong to class C. In class C, the network length is 24 bits, but since three bits define the class, we can have only 21 bits as the network identifier. This means there are $2^{21} = 2,097,152$ networks in the world that can have a class C address.
- Class D is not divided into prefix and suffix. It is used for multicast addresses.
- All addresses that start with 1111 in binary belong to class E. As in Class D, Class E is not divided into prefix and suffix and is used as reserve.

Address Depletion

The reason that classful addressing has become obsolete is address depletion. Since the addresses were not distributed properly, the Internet was faced with the problem of the addresses being rapidly used up, resulting in no more addresses available for organizations and individuals that needed to be connected to the Internet.

Subnetting and Supernetting

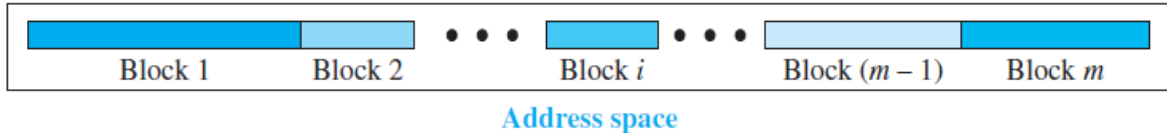
- To alleviate address depletion, two strategies were proposed and, to some extent, implemented: subnetting and supernetting.
- In subnetting, a class A or class B block is divided into several subnets. If all of the addresses in a network are not used, subnetting allows the addresses to be divided among several organizations.
- Supernetting was devised to combine several class C blocks into a larger block to be attractive to organizations that need more than the 256 addresses available in a class C block.

Advantage of Classful Addressing

Although classful addressing had several problems and became obsolete, it had one advantage: Given an address, we can easily find the class of the address and, since the prefix length for each class is fixed, we can find the prefix length immediately.

Classless Addressing

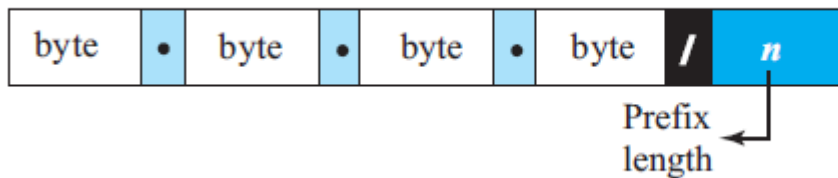
In classless addressing, the whole address space is divided into variable length blocks. The prefix in an address defines the block (network); the suffix defines the node (device). Theoretically, we can have a block of $2^0, 2^1, 2^2, \dots, 2^{32}$ addresses. The number of addresses in a block needs to be a power of 2.



The prefix length in classless addressing is variable. We can have a prefix length that ranges from 0 to 32. The size of the network is inversely proportional to the length of the prefix. A small prefix means a larger network; a large prefix means a smaller network.

Prefix Length: Slash Notation

Since the prefix length is not inherent in the address, we need to separately give the length of the prefix. In this case, the prefix length, n , is added to the address, separated by a slash. The notation is informally referred to as *slash notation* and formally as **classless interdomain routing** or **CIDR** (pronounced cider) strategy.



Examples:
 12.24.76.8/8
 23.14.67.92/12
 220.8.24.255/25

Extracting Information from an Address

If the value of prefix n is given we can easily find the the number of addresses, the first address in the block, and the last address.

1. The number of addresses in the block is found as $N = 2^{32-n}$.
2. To find the first address, we keep the n leftmost bits and set the $(32 - n)$ rightmost bits all to 0s.
3. To find the last address, we keep the n leftmost bits and set the $(32 - n)$ rightmost bits all to 1s.

Example

A classless address is given as 167.199.170.82/27. We can find the above three pieces of information as follows. The number of addresses in the network is $2^{32-n} = 2^5 = 32$ addresses. The first address can be found by keeping the first 27 bits and changing the rest of the bits to 0s.

Address: 167.199.170.82/27	10100111	11000111	10101010	01010010
First address: 167.199.170.64/27	10100111	11000111	10101010	01000000

The last address can be found by keeping the first 27 bits and changing the rest of the bits to 1s.

Address: 167.199.170.82/27	10100111	11000111	10101010	01011111
Last address: 167.199.170.95/27	10100111	11000111	10101010	01011111

Address Mask

Another way to find the first and last addresses in the block is to use the address mask. The address mask is a 32-bit number in which the n leftmost bits are set to 1s and the rest of the bits ($32 - n$) are set to 0s.

1. The number of addresses in the block $N = \text{NOT}(\text{mask}) + 1$.
2. The first address in the block = (Any address in the block) AND (mask).
3. The last address in the block = (Any address in the block) OR [(NOT (mask))].

Example

We repeat same example using the mask. The mask in dotted-decimal notation is 256.256.256.224.

Number of addresses in the block: $N = \text{NOT}(\text{mask}) + 1 = 0.0.0.31 + 1 = 32$ addresses

First address: First = (address) AND (mask) = 167.199.170.82

Last address: Last = (address) OR (NOT mask) = 167.199.170.255

Network Address

Given any address, we can find all information about the block. The first address, the **network address**, is particularly important because it is used in routing a packet to its destination network. The network address is actually the identifier of the network; each network is identified by its network address.

Block Allocation

Block allocation is given to a global authority called the Internet Corporation for Assigned Names and Numbers (ICANN).

It assigns a large block of addresses to an ISP (or a larger organization that is considered an ISP in this case). For the proper operation of the CIDR, two restrictions need to be applied to the allocated block.

1. The number of requested addresses, N , needs to be a power of 2.
2. The requested block needs to be allocated where there is an adequate number of contiguous addresses available in the address space. The first address needs to be divisible by the number of addresses in the block.

Subnetting

More levels of hierarchy can be created using subnetting. An organization (or an ISP) that is granted a range of addresses may divide the range into several subranges and assign each subrange to a subnetwork (or subnet).

Example

An organization is granted a block of addresses with the beginning address 14.24.74.0/24. The organization needs to have 3 subblocks of addresses to use in its three subnets: one subblock of 10 addresses, one subblock of 60 addresses, and one subblock of 120 addresses. Design the subblocks.

Solution

There are $2^{32-24} = 256$ addresses in this block. The first address is 14.24.74.0/24; the last

address is 14.24.74.255/24. To satisfy the third requirement, we assign addresses to subblocks, starting with the largest and ending with the smallest one.

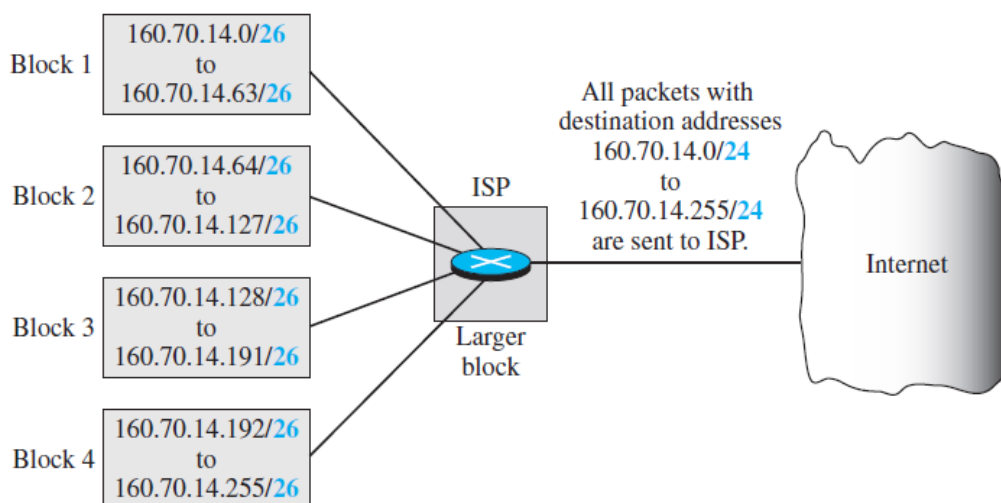
- a. The number of addresses in the largest subblock, which requires 120 addresses, is not a power of 2. We allocate 128 addresses. The subnet mask for this subnet can be found as $n1 = 32 - \log_2 128 = 25$. The first address in this block is 14.24.74.0/25; the last address is 14.24.74.127/25.
- b. The number of addresses in the second largest subblock, which requires 60 addresses, is not a power of 2 either. We allocate 64 addresses. The subnet mask for this subnet can be found as $n2 = 32 - \log_2 64 = 26$. The first address in this block is 14.24.74.128/26; the last address is 14.24.74.191/26.
- c. The number of addresses in the smallest subblock, which requires 10 addresses, is not a power of 2 either. We allocate 16 addresses. The subnet mask for this subnet can be found as $n3 = 32 - \log_2 16 = 28$. The first address in this block is 14.24.74.192/28; the last address is 14.24.74.207/28.
- d. If we add all addresses in the previous subblocks, the result is 208 addresses, which means 48 addresses are left in reserve. The first address in this range is 14.24.74.208. The last address is 14.24.74.255.

Address Aggregation

One of the advantages of the CIDR strategy is **address aggregation** (sometimes called address summarization or route summarization). When blocks of addresses are combined to create a larger block, routing can be done based on the prefix of the larger block. ICANN assigns a large block of addresses to an ISP.

Example

Figure shows how four small blocks of addresses are assigned to four organizations by an ISP. The ISP combines these four blocks into one single block and advertises the larger block to the rest of the world. Any packet destined for this larger block should be sent to this ISP. It is the responsibility of the ISP to forward the packet to the appropriate organization.

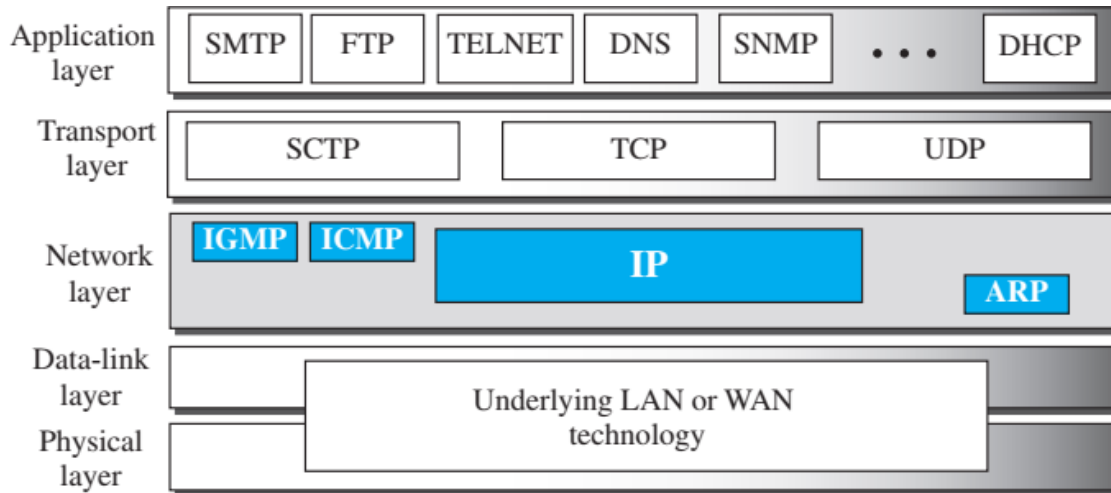


NETWORK LAYER PROTOCOLS (IP, ICMP, MOBILE IP)

There are different protocols used in the network layer for communication over network. The most important among those are: IP, ICMP and Mobile IP.

INTERNET PROTOCOL -IP

Internet Protocol version 4 (IPv4), is responsible for packetizing, forwarding, and delivery of a packet at the network layer. IP layer position is shown in figure below:

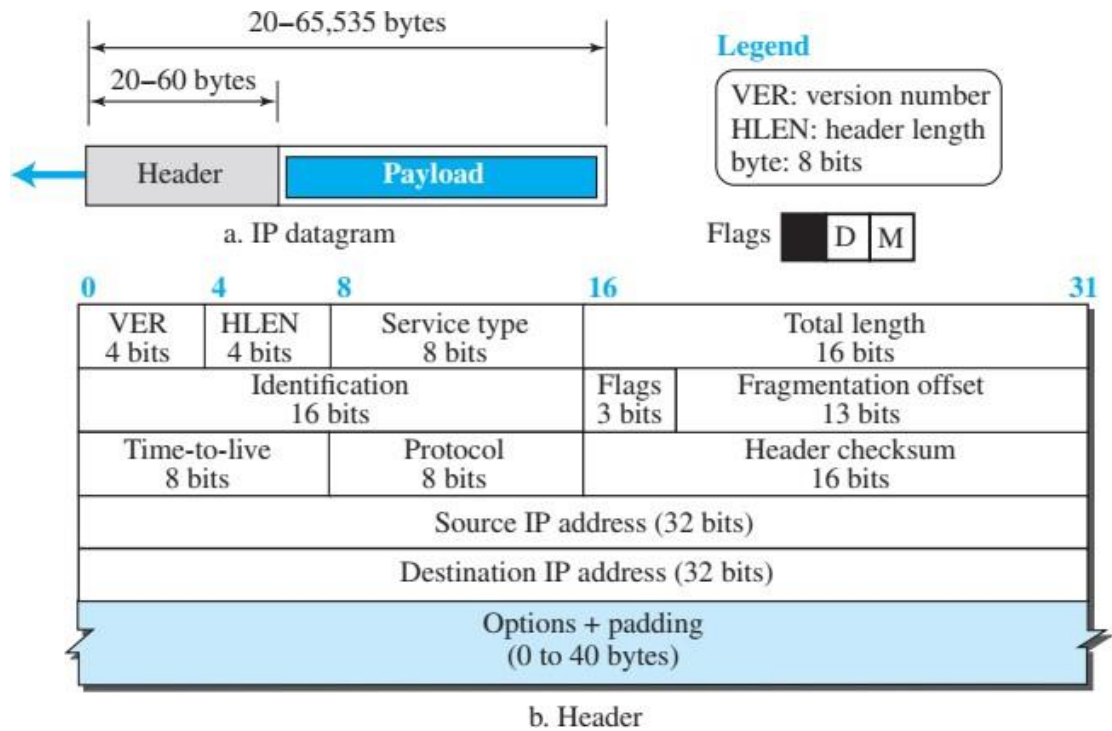


IP layer position in TCP/IP model

- IPv4 is an unreliable datagram protocol—a best-effort delivery service. The term best-effort means that IPv4 packets can be corrupted, be lost, arrive out of order, or be delayed, and may create congestion for the network.
- IPv4 is also a connectionless protocol that uses the datagram approach. This means that each datagram is handled independently, and each datagram can follow a different route to the destination. This implies that datagrams sent by the same source to the same destination could arrive out of order.

Datagram Format

- Packets used by the IP are called datagrams. Figure 19.2 shows the IPv4 datagram format.
- A datagram is a variable-length packet consisting of two parts: header and payload (data).
- The header is 20 to 60 bytes in length and contains information essential to routing and delivery.



IP datagram

Version Number. The 4-bit version number (VER) field defines the version of the IPv4 protocol, which, has the value of 4.

Header Length. The 4-bit header length (HLEN) field defines the total length of the datagram header in 4-byte words. The IPv4 datagram has a variable-length header. When a device receives a datagram, it needs to know when the header stops and the data, which is encapsulated in the packet, starts.

Service Type. In the original design of the IP header, this field was referred to as type of service (TOS), which defined how the datagram should be handled. In the late 1990s, IETF redefined the field to provide differentiated services (DiffServ).

Total Length. This 16-bit field defines the total length (header plus data) of the IP datagram in bytes. A 16-bit number can define a total length of up to 65,535 (when all bits are 1s). The header length can be found by multiplying the value in the HLEN field by 4.

$$\text{Length of data} = \text{total length} - (\text{HLEN}) \times 4$$

Identification, Flags, and Fragmentation Offset. These three fields are related to the fragmentation of the IP datagram when the size of the datagram is larger than the underlying network can carry.

Time-to-live. The time-to-live (TTL) field is used to control the maximum number of hops (routers) visited by the datagram. When a source host sends the datagram, it stores a number in this field. This value is approximately two times the maximum number of routers between any two hosts.

Protocol. In TCP/IP, the data section of a packet, called the payload, carries the whole packet from another protocol. A datagram, for example, can carry a packet belonging to any transport-layer protocol such as UDP or TCP. A datagram can also carry a packet from other protocols that directly use the service of the IP, such as some routing protocols or some auxiliary protocols.

Header checksum. IP is not a reliable protocol; it does not check whether the payload carried by a datagram is corrupted during the transmission. IP puts the burden of error checking of the payload on the protocol that owns the payload, such as UDP or TCP. The datagram header, however, is added by IP, and its error-checking is the responsibility of IP. Errors in the IP header can be a disaster. checksum in the Internet normally uses a 16-bit field, which is the complement of the sum of other fields calculated using 1s complement arithmetic.

Source and Destination Addresses. These 32-bit source and destination address fields define the IP address of the source and destination respectively. The source host should know its IP address. The destination IP address is either known by the protocol that uses the service of IP or is provided by the DNS.

Options. A datagram header can have up to 40 bytes of options. Options can be used for network testing and debugging. Although options are not a required part of the IP header, option processing is required of the IP software.

Payload. Payload, or data, is the main reason for creating a datagram. Payload is the packet coming from other protocols that use the service of IP

Internet Control Message Protocol ICMP

The ICMP stands for Internet Control Message Protocol. It is a network layer protocol. It is used for error handling in the network layer, and it is primarily used on network devices such as routers. As different types of errors can exist in the network layer, so ICMP can be used to report these errors and to debug those errors. The IP protocol does not have any error-reporting or error-correcting mechanism, so it uses a message to convey the information.

ICMP messages are not passed directly to the data-link layer as would be expected. Instead, the messages are first encapsulated inside IP datagrams before going to the lower layer. When an IP datagram encapsulates an ICMP message, the value of the

protocol field in the IP datagram is set to 1 to indicate that the IP payload is an ICMP message.

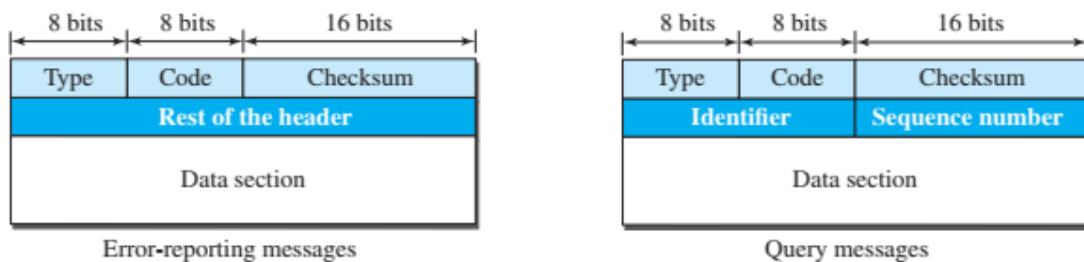
Position of ICMP in the network layer

The ICMP resides in the IP layer, as shown in the below diagram.



Position of ICMP in IP layer

General format of ICMP Messages:



Type and code values

Error-reporting messages

- 03: Destination unreachable (codes 0 to 15)
- 04: Source quench (only code 0)
- 05: Redirection (codes 0 to 3)
- 11: Time exceeded (codes 0 and 1)
- 12: Parameter problem (codes 0 and 1)

Query messages

- 08 and 00: Echo request and reply (only code 0)
- 13 and 14: Timestamp request and reply (only code 0)

Fig.2.50 ICMP message format

- **Type:** It is an 8-bit field. It defines the ICMP message type. The values range from 0 to 127 are defined for ICMPv6, and the values from 128 to 255 are the informational messages.
- **Code:** It is an 8-bit field that defines the subtype of the ICMP message
- **Checksum:** It is a 16-bit field to detect whether the error exists in the message or not.

ICMP MESSAGES:

The ICMP messages are usually divided into two categories:

ICMP messages

Category	Type	Message
Error-reporting messages	3	Destination unreachable
	4	Source quench
	11	Time exceeded
	12	Parameter problem
	5	Redirection
Query messages	8 or 0	Echo request or reply
	13 or 14	Timestamp request or reply

- **Error-reporting messages**

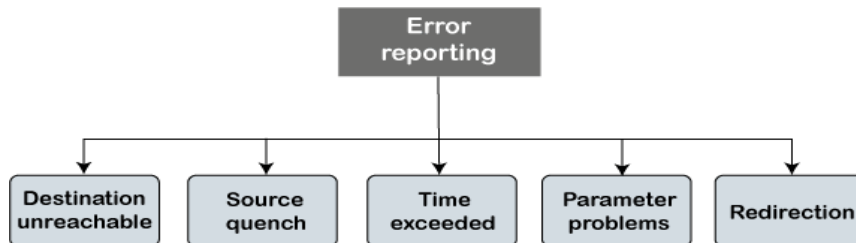
The error-reporting message means that the router encounters a problem when it processes an IP packet then it reports a message.

- **Query messages**

The query messages are those messages that help the host to get the specific information of another host. For example, suppose there are a client and a server, and the client wants to know whether the server is live or not, then it sends the ICMP message to the server.

Types of Error Reporting messages

The error reporting messages are broadly classified into the following categories:



A) Destination unreachable

The destination unreachable error occurs when the packet does not reach the destination. Suppose the sender sends the message, but the message does not reach the destination, then the intermediate router reports to the sender that the destination is unreachable.

Type: 3	Code: 0 to 15	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Fig.2.51 Destination unreachable

Type: It defines the type of message. The number 3 specifies that the destination is unreachable.

Code (0 to 15): It is a 4-bit number which identifies whether the message comes from some intermediate router or the destination itself.

B) Source quench

There is no flow control or congestion control mechanism in the network layer or the IP protocol. The sender is concerned with only sending the packets, and the sender does not think whether the receiver is ready to receive those packets or if there is any congestion in the network layer so that the sender can send a lesser number of packets, so there is no flow control or congestion control mechanism. In this case, ICMP provides feedback, i.e., source quench. Suppose the sender resends the packet at a higher rate, and the router is not able to handle the high data rate. To overcome such a situation, the router sends a source quench message to tell the sender to send the packet at a lower rate.

Type: 4	Code: 0	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Fig.2.52 Source quench

C) Time exceeded

Sometimes the situation arises when there are many routers that exist between the sender and the receiver. When the sender sends the packet, then it moves in a routing loop. The time exceeded is based on the time-to-live value. When the packet traverses through the router, then each router decreases the value of TTL by one. Whenever a router decreases a datagram with a time-to-live value to zero, then the router discards a datagram and sends the time exceeded message to the original source.

Type: 11	Code: 0 or 1	Checksum
Unused (All 0s)		
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Fig.2.53 Time exceeded

The above message format shows that the type of time-exceeded is 11, and the code can be either 0 or 1. The code 0 represents TTL, while code 1 represents fragmentation. In a time-exceeded message, the code 0 is used by the router to show that the time-to-live value is reached to zero.

The code 1 is used by the destination to show that all the fragments do not reach within a set time.

D) Parameter problems

The router and the destination host can send a parameter problem message. This message conveys that some parameters are not properly set.

Type: 12	Code: 0 or 1	Checksum
Pointer	Unused (All 0s)	
Part of the received IP datagram including IP header plus the first 8 bytes of datagram data		

Fig.2.54 Parameter problems

The above diagram shows the message format of the parameter problem. The type of message is 12, and the code can be 0 or 1.

E) Redirection

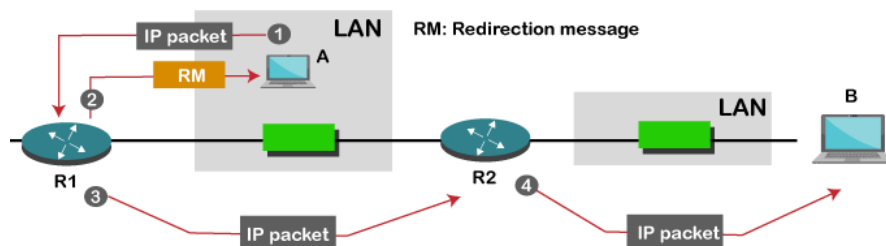


Fig.2.55 Redirection

When the packet is sent, then the routing table is gradually augmented and updated. The tool used to achieve this is the redirection message. For example, A wants to send the packet to B, and there are two routers exist between A and B. First, A sends the data to the router 1. The router 1 sends the IP packet to router 2 and redirection message to A so that A can update its routing table.

ICMP Query Messages

The ICMP Query message is used for error handling or debugging the internet. This message is commonly used to ping a message.

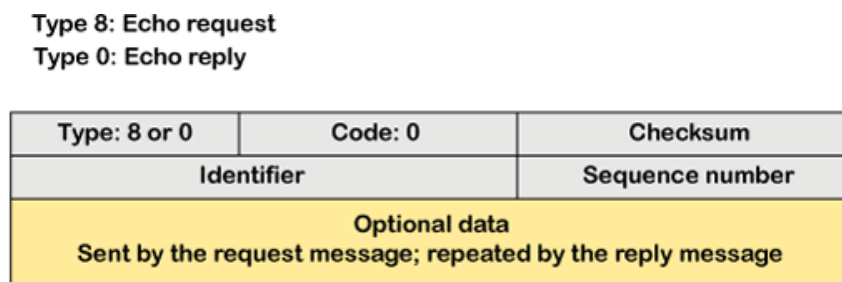
A) Echo-request and echo-reply message

A router or a host can send an echo-request message. It is used to ping a message to another host that "Are you alive". If the other host is alive, then it sends the echo-reply message. An echo-reply message is sent by the router or the host that receives an echo-request message.

Key points of Query messages

1. The echo-request message and echo-reply message can be used by the network managers to check the operation of the IP protocol. Suppose two hosts, i.e., A and B, exist, and A wants to communicate with host B. The A host can communicate to host B if the link is not broken between A and B, and B is still alive.
2. The echo-request message and echo-reply message check the host's reachability, and it can be done by invoking the ping command.

The message format of echo-request and echo-reply message



The above diagram shows the message format of the echo-request and echo-reply message. The type of echo-request is 8, and the request of echo-reply is 0. The code of this message is 0.

B) Timestamp-request and timestamp-reply message

The timestamp-request and timestamp-reply messages are also a type of query messages. Suppose the computer A wants to know the time on computer B, so it sends the timestamp-request message to computer B. The computer B responds with a timestamp-reply message.

Message format of timestamp-request and timestamp-reply

Type 13: request
 Type 14: reply

Type: 13 or 14	Code: 0	Checksum
Identifier		Sequence number
Original timestamp		
Receive timestamp		
Transmit timestamp		

The type of timestamp-request is 13, and the type of timestamp-reply is 14. The code of this type of message is 0.

2.7.2 Mobile IP

Explain the term mobile IP and how it is compared with computer networks. (7) Nov/Dec 2021

Mobile IP is the extension of IP protocol that allows mobile computers to be connected to the Internet at any location where the connection is possible.

Addressing

The main problem that must be solved in providing mobile communication using the IP protocol is addressing.

Two Addresses

The approach that is more feasible is the use of two addresses. The host has its original address, called the **home address**, and a temporary address, called the **care-of address**. The home address is permanent; it associates the host with its **home network**, the network that is the permanent home of the host. The care-of address is temporary. When a host moves from one network to another, the care-of address changes; it is associated with the **foreign network**, the network to which the host moves.

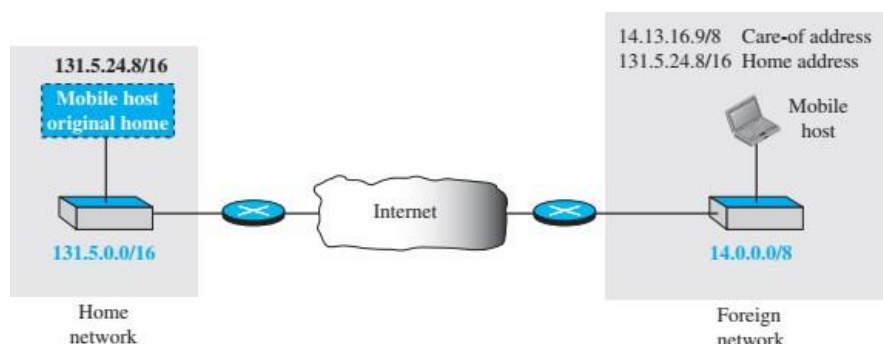


Fig.2.56 Home address and care-of address

Agents

To make the change of address transparent to the rest of the Internet requires a **home agent** and a **foreign agent**. Figure below shows the position of a home agent relative to the home network and a foreign agent relative to the foreign network.

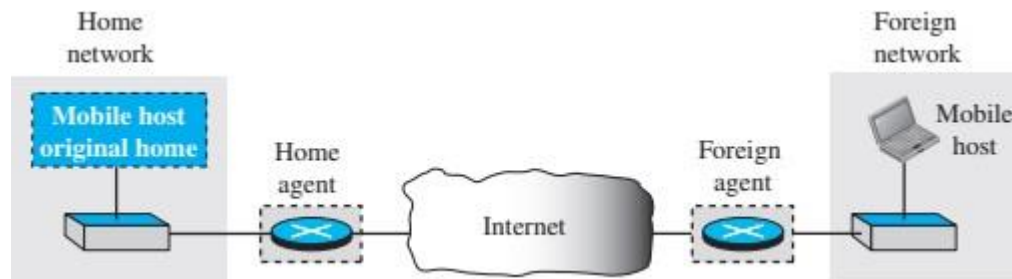


Fig.2.57 Home agent and foreign agent

Home Agent

The home agent is usually a router attached to the home network of the mobile host. The home agent acts on behalf of the mobile host when a remote host sends a packet to the mobile host. The home agent receives the packet and sends it to the foreign agent.

Foreign Agent

The foreign agent is usually a router attached to the foreign network. The foreign agent receives and delivers packets sent by the home agent to the mobile host. The mobile host can also act as a foreign agent. In other words, the mobile host and the foreign agent can be the same. However, to do this, a mobile host must be able to receive a care-of address by itself, which can be done through the use of DHCP. When the mobile host acts as a foreign agent, the care-of address is called a **collocated care-of address**.

Three Phases

To communicate with a remote host, a mobile host goes through three phases: **agent discovery, registration, and data transfer**.

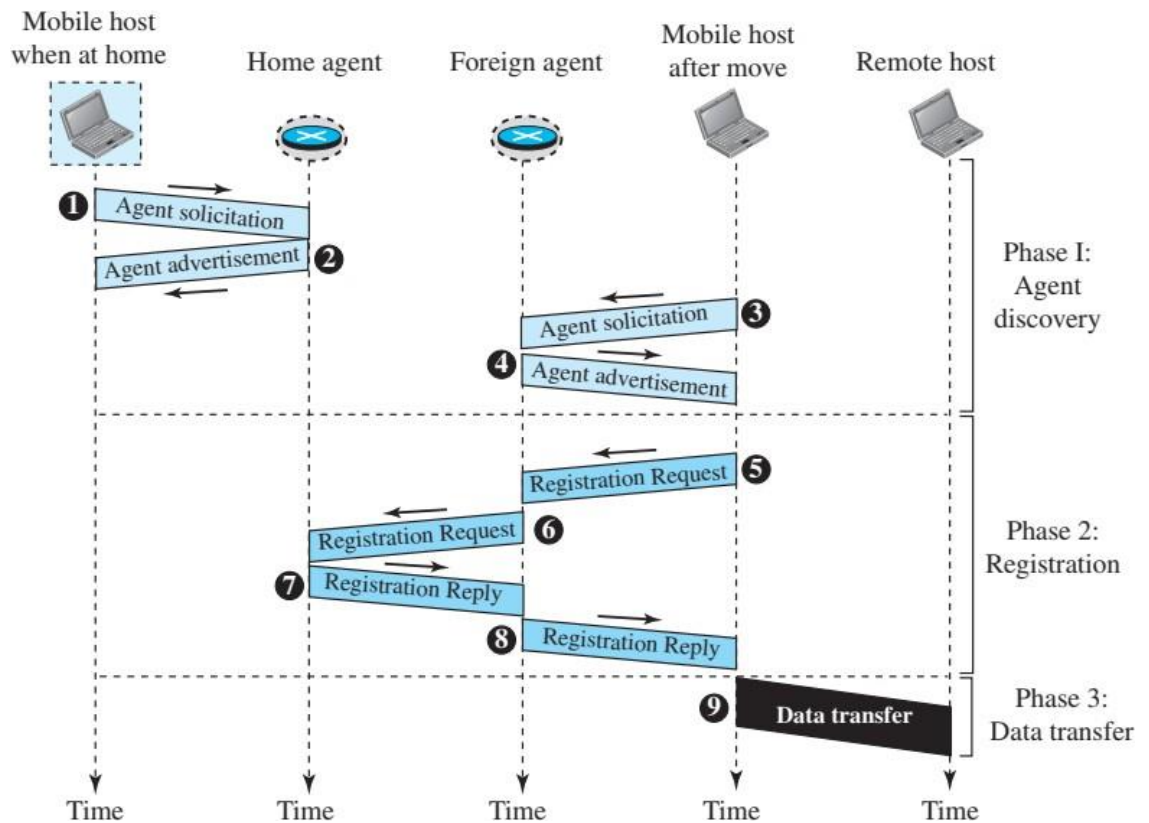


Fig.2.58 Remote host and mobile host communication

A) Agent Discovery

The first phase in mobile communication, agent discovery, consists of two subphases. A mobile host must discover (learn the address of) a home agent before it leaves its home network. A mobile host must also discover a foreign agent after it has moved to a foreign network. This discovery consists of learning the care-of address as well as the foreign agent's address. The discovery involves two types of messages: advertisement and solicitation.

Agent Advertisement

When a router advertises its presence on a network using an ICMP router advertisement, it can append an agent advertisement to the packet if it acts as an agent.

ICMP Advertisement message			
Type	Length	Sequence number	
Lifetime		Code	Reserved
Care-of addresses (foreign agent only)			

The field descriptions are as follows:

- ❑ **Type.** The 8-bit type field is set to 16.
- ❑ **Length.** The 8-bit length field defines the total length of the extension message (not the length of the ICMP advertisement message).
- ❑ **Sequence number.** The 16-bit sequence number field holds the message number. The recipient can use the sequence number to determine if a message is lost.
- ❑ **Lifetime.** The lifetime field defines the number of seconds that the agent will accept requests. If the value is a string of 1s, the lifetime is infinite.
- ❑ **Code.** The code field is an 8-bit flag in which each bit is set (1) or unset (0).
- ❑ **Care-of Addresses.** This field contains a list of addresses available for use as care-of addresses. The mobile host can choose one of these addresses. The selection of this care-of address is announced in the registration request. Note that this field is used only by a foreign agent.

Agent

Solicitation

When a mobile host has moved to a new network and has not received agent advertisements, it can initiate an agent solicitation. It can use the ICMP solicitation message to inform an agent that it needs assistance.

Registration

The second phase in mobile communication is registration. After a mobile host has moved to a foreign network and discovered the foreign agent, it must register. There are four aspects of registration:

1. The mobile host must register itself with the foreign agent.
2. The mobile host must register itself with its home agent. This is normally done by the foreign agent on behalf of the mobile host.
3. The mobile host must renew registration if it has expired.
4. The mobile host must cancel its registration (deregistration) when it returns home.

Request and Reply

To register with the foreign agent and the home agent, the mobile host uses a registration request and a registration reply as shown in Figure below.

Registration Request A registration request is sent from the mobile host to the foreign agent to register its care-of address and also to announce its home address and home agent address. The foreign agent, after receiving and registering the request, relays the message to the home agent.

Type	Flag	Lifetime
Home address		
Home agent address		
Care-of address		
Identification		
Extensions ...		

Fig.2.59 Registration request format

Registration Reply A registration reply is sent from the home agent to the foreign agent and then relayed to the mobile host. The reply confirms or denies the registration request. Figure shows the format of the registration reply. The fields are similar to those of the registration request with the following exceptions. The value of the type field is 3. The code field replaces the flag field and shows the result of the registration request (acceptance or denial). The care-of address field is not needed.

Type	Code	Lifetime
Home address		
Home agent address		
Identification		
Extensions ...		

Fig.2.60 Registration reply format

B) Data Transfer

After agent discovery and registration, a mobile host can communicate with a remote host.

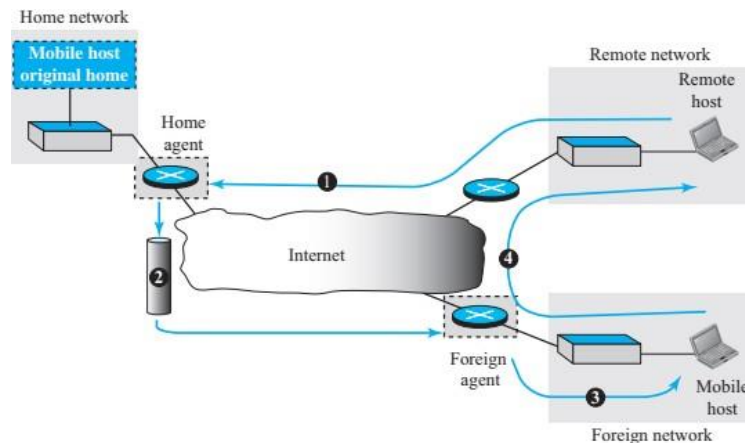


Fig.2.61 Data Transfer

Step 1: From Remote Host to Home Agent

When a remote host wants to send a packet to the mobile host, it uses its address as the source address and the home address of the mobile host as the destination address. In other words, the remote host sends a packet as though the mobile host is at its home network.

Step 2: From Home Agent to Foreign Agent

After receiving the packet, the home agent sends the packet to the foreign agent, using the tunneling concept. The home agent encapsulates the whole IP packet inside another IP packet using its address as the source and the foreign agent's address as the destination.

Step 3: From Foreign Agent to Mobile Host

When the foreign agent receives the packet, it removes the original packet. As the destination address is the home address of the mobile host, the foreign agent consults a registry table to find the care-of address of the mobile host. The packet is then sent to the care-of address.

Step 4: From Mobile Host to Remote Host

When a mobile host wants to send a packet to a remote host (for example, a response to the packet it has received), it sends as it does normally. The mobile host prepares a packet with its home address as the source, and the address of the remote host as the destination.

Inefficiency in Mobile IP

a) Double Crossing

Double crossing occurs when a remote host communicates with a mobile host that has moved to the same network (or site) as the remote host.

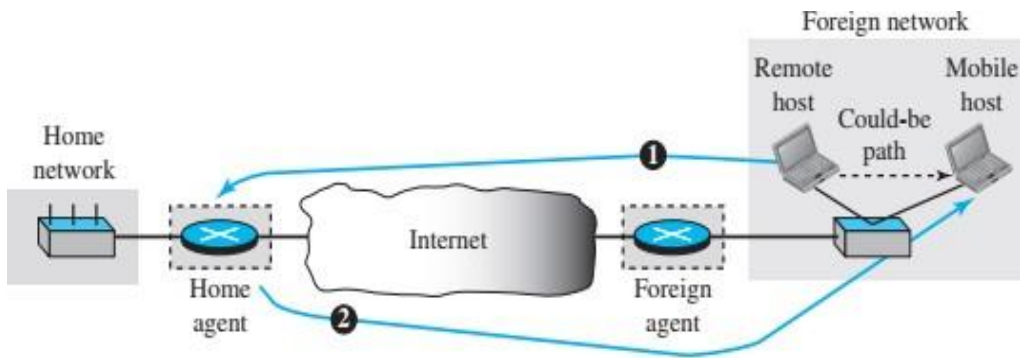


Fig.2.62 Double Crossing

b) Triangle Routing

Triangle routing, the less severe case, occurs when the remote host communicates with a mobile host that is not attached to the same network (or site) as the mobile host.

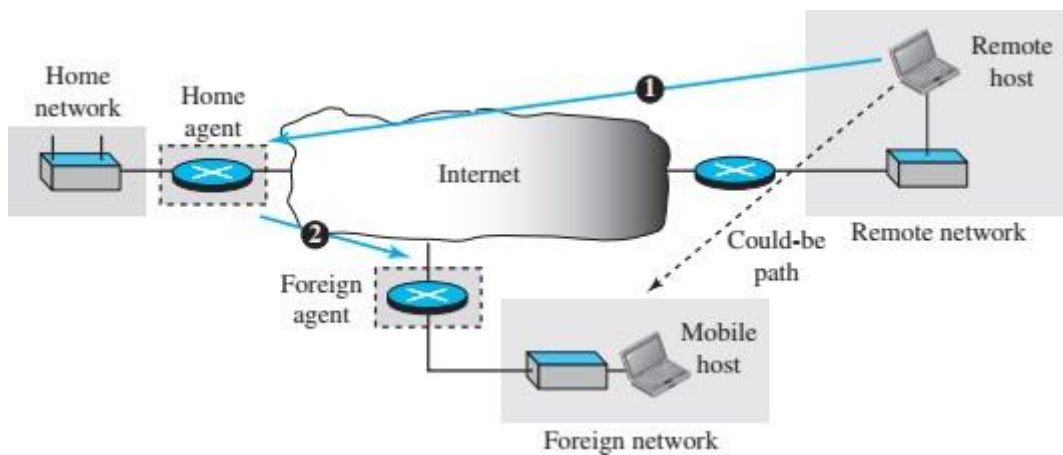


Fig.2.63 Triangle Routing

Routing

- In an internet, the goal of the network layer is to deliver a datagram from its source to its destination or destinations. If a datagram is destined for only one destination (one-to-one delivery), we have unicast routing. If the datagram is destined for several destinations (one-to-many delivery), we have multicast routing.
- The routing can be possible if a router has a forwarding table to forward a packet to the appropriate next node on its way to the final destination or destinations. To make the forwarding tables of the router, the Internet needs routing protocols that will be active all the time in the background and update the forwarding tables.

Unicast Routing

In unicast routing, a packet is routed, hop by hop, from its source to its destination by the help of forwarding tables. The source host needs no forwarding table because it delivers its packet to the default router in its local network. The destination host needs no forwarding table either because it receives the packet from its default router in its local network. This means that only the routers that glue together the networks in the internet need forwarding tables.

Routing a packet from its source to its destination means routing the packet from a source

router (the default router of the source host) to a destination router (the router connected to the destination network).

Autonomous Systems

Name and compare three different types of Autonomous system (2 marks nov/dec 2020)
Each ISP is an autonomous system when it comes to managing networks and routers under its control. The autonomous systems, however, are not categorized according to their size; they are categorized according to the way they are connected to another ASs.

Stub AS. A stub AS has only one connection to another AS. The data traffic can be either initiated or terminated in a stub AS; the data cannot pass through it. A good example of a stub AS is the customer network, which is either the source or the sink of data.

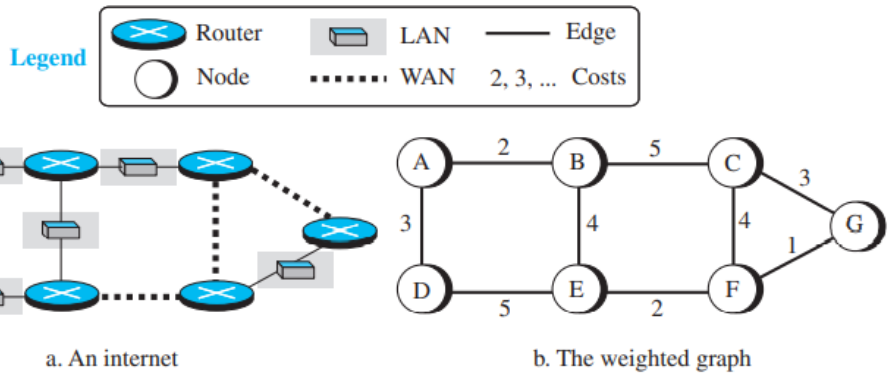
Multihomed AS. A multihomed AS can have more than one connection to other ASs, but it does not allow data traffic to pass through it. A good example of such an AS is some of the customer ASs that may use the services of more than one provider network, but their policy does not allow data to be passed through them.

Transient AS. A transient AS is connected to more than one other AS and also allows the traffic to pass through. The provider networks and the backbone are good examples of transient ASs.

- The routing protocol run in each AS is referred to as intra-AS routing protocol, intradomain routing protocol, or interior gateway protocol (IGP); the global routing protocol is referred to as inter-AS routing protocol, interdomain routing protocol, or exterior gateway protocol (EGP).
- The two common intradomain routing protocols are RIP and OSPF; the only interdomain routing protocol is BGP.
- The Routing Information Protocol (RIP) is one of the most widely used intradomain routing protocols based on the distance-vector routing algorithm.
- Open Shortest Path First (OSPF) is also an intradomain routing protocol like RIP, but it is based on the link-state routing protocol.
- Border Gateway Protocol version 4 (BGP4) is the only interdomain routing protocol used in the Internet today. BGP4 is based on the path-vector algorithm.

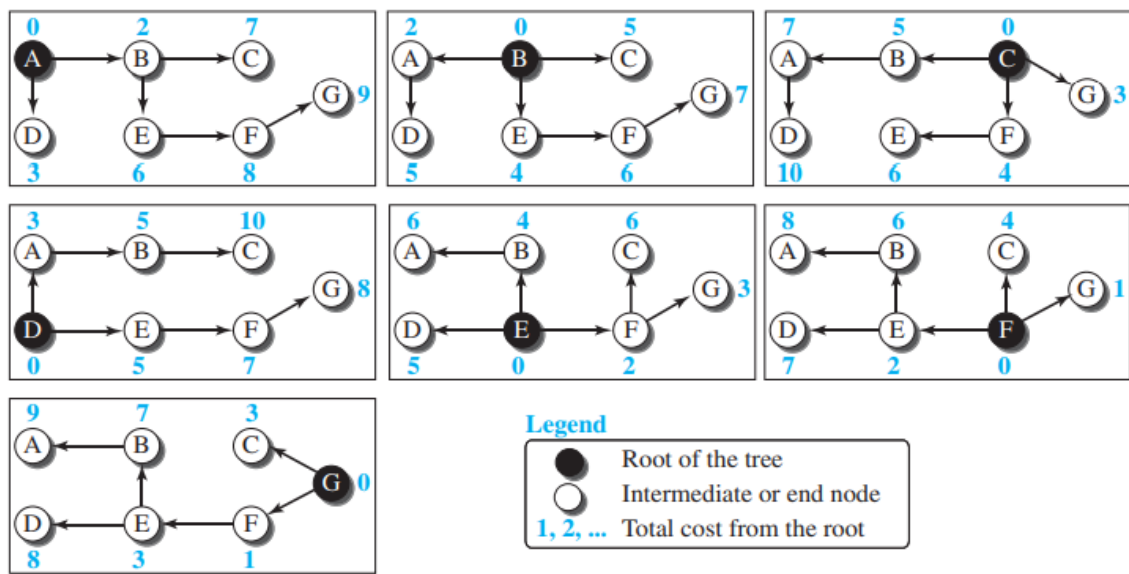
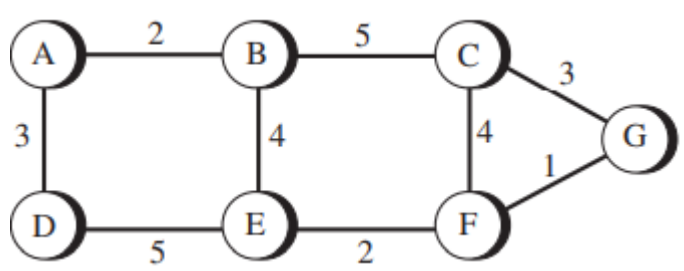
Least-Cost Routing

- When an internet is modeled as a weighted graph, one of the ways to interpret the best route from the source router to the destination router is to find the least cost between the two.
- In other words, the source router chooses a route to the destination router in such a way that the total cost for the route is the least cost among all possible routes.
- In Figure, the best route between A and E is A-B-E, with the cost of 6. This means that each router needs to find the least-cost route between itself and all the other routers to be able to route a packet using these criteria.



Least-Cost Trees

- A least-cost tree is a tree with the source router as the root that spans the whole graph (visits all other nodes) and in which the path between the root and any other node is the shortest.
- In this way, we can have only one shortest-path tree for each node; we have N least-cost trees for the whole internet.



Least cost tree

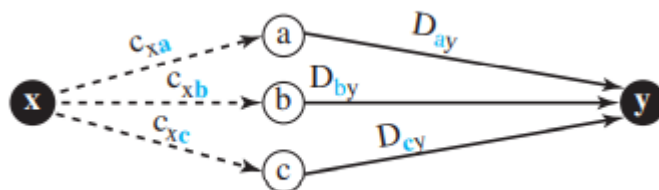
ROUTING ALGORITHMS

Compare distance vector routing and link state routing (nov/dec 2019,2020 13 marks)

1. Distance-Vector Routing

In distance-vector routing, the first thing each node creates is its own least-cost tree with the rudimentary information it has about its immediate neighbours. The incomplete trees are exchanged between immediate neighbours to make the trees more and more complete and to represent the whole internet.

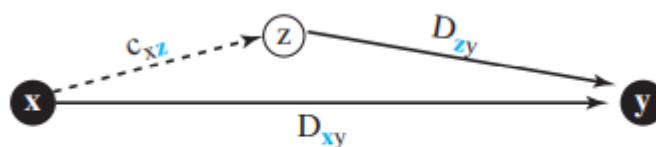
Bellman-Ford Equation



a. General case with three intermediate nodes

$$D_{xy} = \min \{ (c_{xa} + D_{ay}), (c_{xb} + D_{by}), (c_{xc} + D_{cy}), \dots \}$$

D_{ij} is the shortest distance and c_{ij} is the cost between nodes i and j .



b. Updating a path with a new route

$$D_{xy} = \min \{ D_{xy}, (c_{xz} + D_{zy}) \}$$

The Bellman-Ford equation enables us to build a new least-cost path from previously established least-cost paths.

$(a \rightarrow y)$, $(b \rightarrow y)$, and $(c \rightarrow y)$ as previously established least-cost paths and $(x \rightarrow y)$ as the new least-cost path.

Distance Vectors

In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node.

Initialization

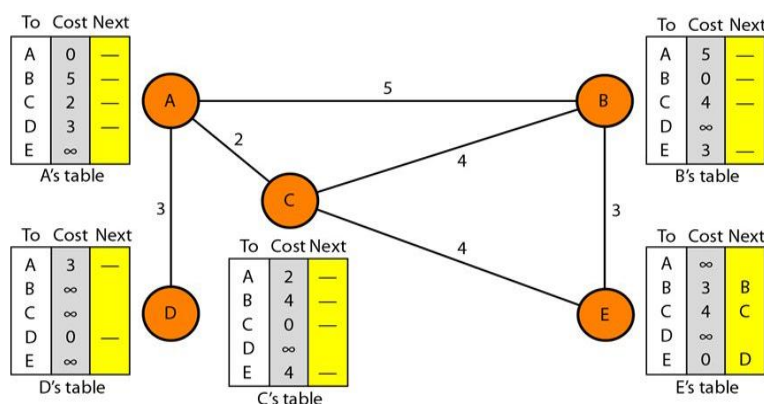


Figure shows the initial tables for each node. The distance for any entry that is not a neighbour is marked as infinite (unreachable).

Sharing

The whole idea of distance vector routing is the sharing of information between neighbors. Although

node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other. sharing here means sharing only the first two columns.

Updating

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

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

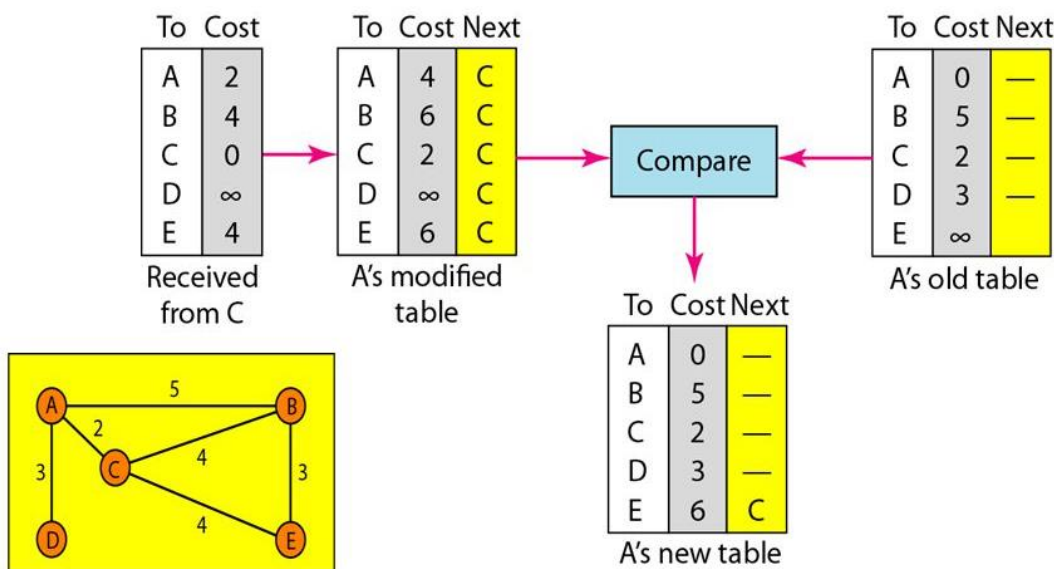


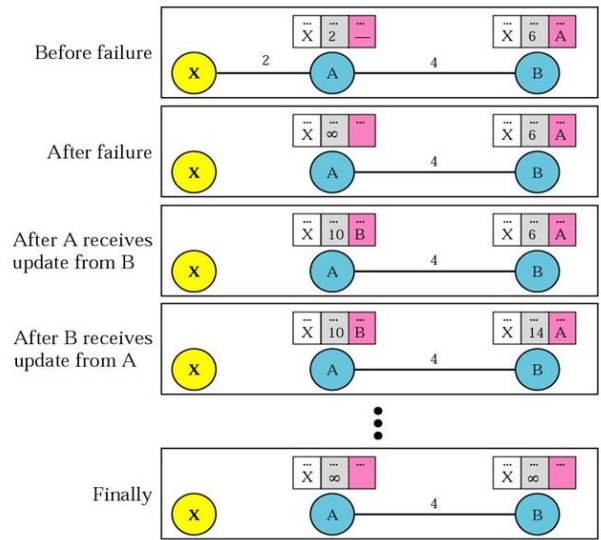
Fig. Sharing; Updating in distance vector routing

Periodic Update A node sends its routing table, normally every 30 s, in a periodic update. The period depends on the protocol that is using distance vector routing.

Triggered Update A node sends its two-column routing table to its neighbors anytime there is a change in its routing table. This is called a triggered update. The change can result from the following.

1. A node receives a table from a neighbor, resulting in changes in its own table after updating.
2. A node detects some failure in the neighboring links which results in a distance change to infinity.

Two-Node Loop Instability



At the beginning, both nodes A and B know how to reach node X. But suddenly, the link between A and X fails. Node A changes its table.

If A can send its table to B immediately, everything is fine. However, the system becomes unstable if B sends its routing table to A before receiving A's routing table.

Node A receives the update and, assuming that B has found a way to reach X, immediately updates its routing table. Based on the triggered update strategy, A sends its new update to B. Now B thinks that something has been changed around A and updates its routing table. The cost of reaching X increases gradually until it reaches infinity.

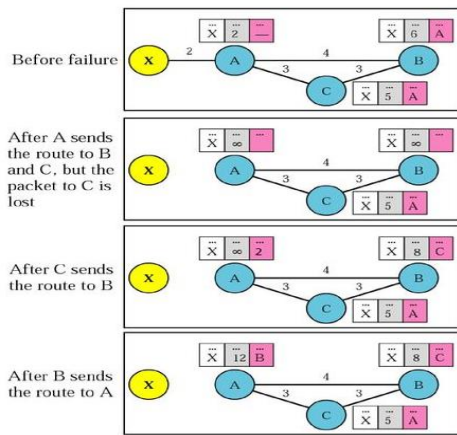
A few solutions have been proposed for instability of this kind.

Defining Infinity The first obvious solution is to redefine infinity to a smaller number, such as 100. Most implementations of the distance vector protocol define the distance between each node to be I and define 16 as infinity. However, this means that the distance vector routing cannot be used in large systems. The size of the network, in each direction, cannot exceed 15 hops.

Split Horizon Another solution is called split horizon. In this strategy, instead of flooding the table through each interface, each node sends only part of its table through each interface. If, according to its table, node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A.

Split Horizon and Poison Reverse Using the split horizon strategy has one drawback. Normally, the distance vector protocol uses a timer, and if there is no news about a route, the node deletes the route from its table. When node B in the previous scenario eliminates the route to X from its advertisement to A, node A cannot guess that this is due to the split horizon strategy (the source of information was A) or because B has not received any news about X recently. The split horizon strategy can be combined with the poison reverse strategy. Node B can still advertise the value for X, but if the source of information is A, it can replace the distance with infinity.

Three-Node Instability



The two-node instability can be avoided by using the split horizon strategy combined with poison reverse. However, if the instability is between three nodes, stability cannot be guaranteed. Suppose, after finding that X is not reachable, node A sends a packet to B and C to inform them of the situation. Node B immediately updates its table, but the packet to C is lost in the network and never reaches C. Node C remains in the dark and still thinks that there is a route to X via A with a distance of 5. After a while, node C sends to B its routing table, which includes the route to X. Node B is totally fooled here. It receives information on the route to X from C, and according to the algorithm, it updates its table, showing the route to X via C with a cost of 8. This information has come from C, not from A, so after a while node B may advertise this route to A. Now A is fooled and updates its table to show that A can reach X via B with a cost of 12. Of course, the loop continues; now A advertises the route to X to C, with increased cost, but not to B. Node C then advertises the route to B with an increased cost. Node B does the same to A. And so on. The loop stops when the cost in each node reaches infinity.

2. Link State Routing

Link state routing has a different philosophy from that of distance vector routing. In link state routing, if each node in the domain has the entire topology of the domain the list of nodes and links, how they are connected including the type, cost (metric), and condition of the links (up or down)-the node can use Dijkstra's algorithm to build a routing table.

Building Routing Tables

In link state routing, four sets of actions are required to ensure that each node has the routing table showing the least-cost node to every other node.

1. Creation of the states of the links by each node, called the link state packet (LSP).
2. Dissemination of LSPs to every other router, called flooding, in an efficient and reliable way.
3. Formation of a shortest path tree for each node.
4. Calculation of a routing table based on the shortest path tree.

Creation of Link State Packet (LSP)

Link state packet consists of the node identity, the list of links, a sequence number, and age. The first two, node identity and the list of links, are needed to make the topology. The third, sequence number, facilitates flooding and distinguishes new LSPs from old ones. The fourth, age, prevents old LSPs from remaining in the domain for a long time.

LSPs are generated on two occasions:

1. When there is a change in the topology of the domain.
2. On a periodic basis. The period in this case is much longer compared to distance vector routing. The timer set for periodic dissemination is normally in the range of 60 min or 2 h based on the implementation

Flooding of LSPs

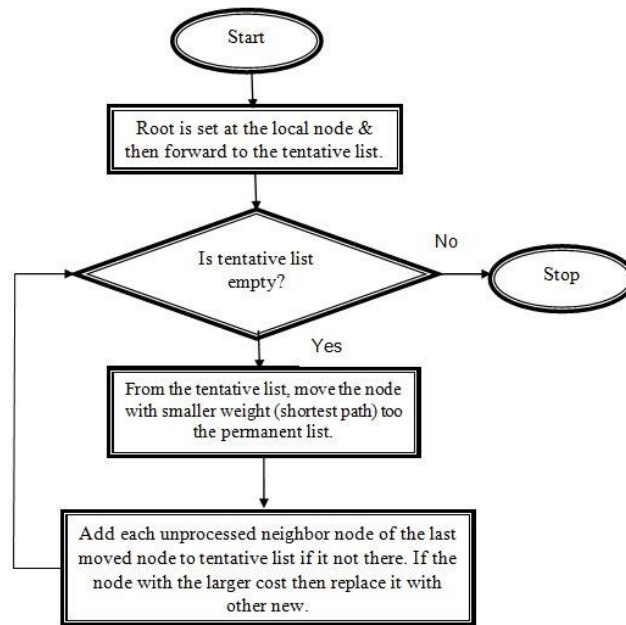
After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbours. The process is called flooding and based on the following:

1. Creating node sends a copy of the LSP out of each interface.
2. A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following:
 - a. It discards the old LSP and keeps the new one.
 - b. It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

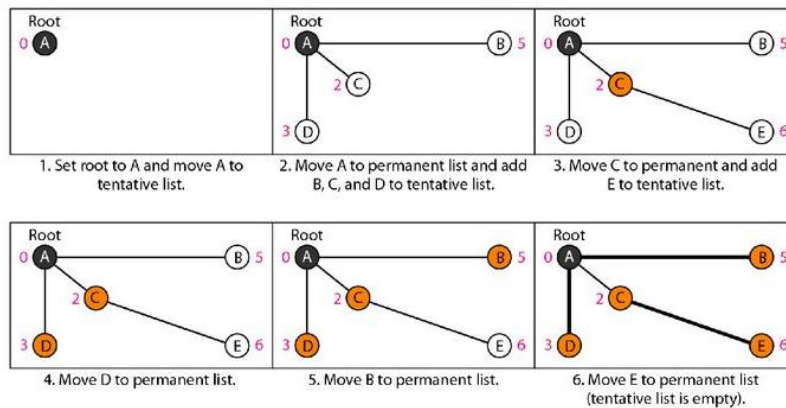
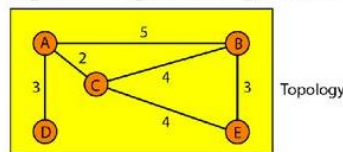
Formation of Shortest Path Tree: Dijkstra Algorithm

- After receiving all LSPs, each node will have a copy of the whole topology. In order to find shortest path shortest path tree is needed. A shortest path tree is a tree in which the path between the root and every other node is the shortest.

- The Dijkstra algorithm creates a shortest path tree from a graph. The algorithm divides the nodes into two sets: tentative and permanent. It finds the neighbours of a current node, makes them tentative, examines them, and if they pass the criteria, makes them permanent.



Let us apply the algorithm to node A of our sample graph in Figure
Example of Dijkstra Algorithm



- We make node A the root of the tree and move it to the tentative list. Our two lists are **Permanent list: empty** **Tentative list: A(0)**
- Node A has the shortest cumulative cost from all nodes in the tentative list. We move A to the permanent list and add all neighbours of A to the tentative list. Our new lists are **Permanent list: A(0)** **Tentative list: B(5), C(2), D(3)**
- Node C has the shortest cumulative cost from all nodes in the tentative list. We move C to the permanent list. Node C has three neighbors, but node A is already processed, which makes the unprocessed neighbors just B and E. However, B is already in the tentative list with a cumulative cost of 5. Node A could also reach node B through C with a cumulative cost of 6. Since 5 is less than 6, we keep node B with a cumulative cost of 5 in the tentative list and do not replace it. Our new lists are **Permanent list: A(0), C(2)** **Tentative list: B(5), D(3), E(6)**
- Node D has the shortest cumulative cost of all the nodes in the tentative list. We move D to the permanent list. Node D has no unprocessed neighbor to be added to the tentative list. Our new lists

are

Permanent list: A(0), C(2), D(3)

Tentative list: B(5), E(6)

5. Node B has the shortest cumulative cost of all the nodes in the tentative list. We move B to the permanent list. We need to add all unprocessed neighbors of B to the tentative list (this is just node E). However, E(6) is already in the list with a smaller cumulative cost. The cumulative cost to node E, as the neighbor of B, is 8. We keep node E(6) in the tentative list. Our new lists are

Permanent list: A(0), B(5), C(2), D(3)

Tentative list: E(6)

6. Node E has the shortest cumulative cost from all nodes in the tentative list. We move E to the permanent list. Node E has no neighbor. Now the tentative list is empty. We stop; our shortest path tree is ready. The final lists are

Permanent list: A(0), B(5), C(2), D(3), E(6)

Tentative list: empty

Routing table for A

<i>Node</i>	<i>Cost</i>	<i>Next Router</i>
A	0	-
B	5	-
C	2	-
D	3	-
E	6	C

Distance Vector Routing	Link State Routing
No flooding, small packets and local sharing require less bandwidth.	More bandwidth required to facilitate flooding and sending large link state packets.
Uses Bellman-Ford algorithm.	Uses Dijkstra's algorithm.
Less traffic.	More network traffic when compared to Distance Vector Routing.
Updates table based on information from neighbours, thus uses local knowledge.	It has knowledge about the entire network, thus it uses global knowledge.
Persistent looping problem exists.	Only transient loop problems.
Based on least hops.	Based on least cost.
Updation of full routing tables.	Updation of only link states.
Less CPU utilisation.	High CPU utilisation.
Uses broadcast for updates.	Uses multicast for updates.
Moderate convergence time.	Low convergence time.

3. Path-Vector Routing

Both link-state and distance-vector routing are based on the least-cost goal. However, there are instances where this goal is not the priority. For example, a router may belong to an organization that does not provide enough security or it may belong to a commercial rival of the sender which might inspect the packets for obtaining information. Least-cost routing does not prevent a packet from passing through an area when that area is in the least-cost path. To respond to these demands, a third routing algorithm, called path-vector (PV) routing has been devised.

Path-vector routing is not based on least-cost routing. The best route is determined by the source

using the policy it imposes on the route.

Spanning Trees

In path-vector routing, the path from a source to all destinations is also determined by the best spanning tree. The best spanning tree, however, is not the least-cost tree; it is the tree determined by the source when it imposes its own policy. If there is more than one route to a destination, the source can choose the route that meets its policy best.

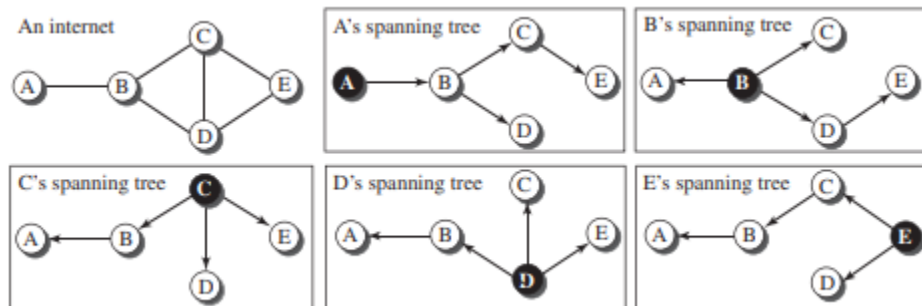
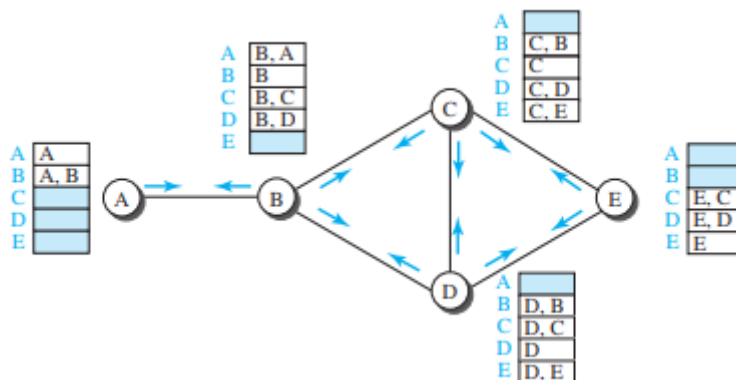


Figure shows a small internet with only five nodes. Each source has created its own spanning tree that meets its policy. The policy imposed by all sources is to use the minimum number of nodes to reach a destination. The spanning tree selected by A and E is such that the communication does not pass through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

Creation of Spanning Trees

The spanning trees are made, gradually and asynchronously, by each node. When a node is booted, it creates a path vector based on the information it can obtain about its immediate neighbor. A node sends greeting messages to its immediate neighbors to collect these pieces of information.

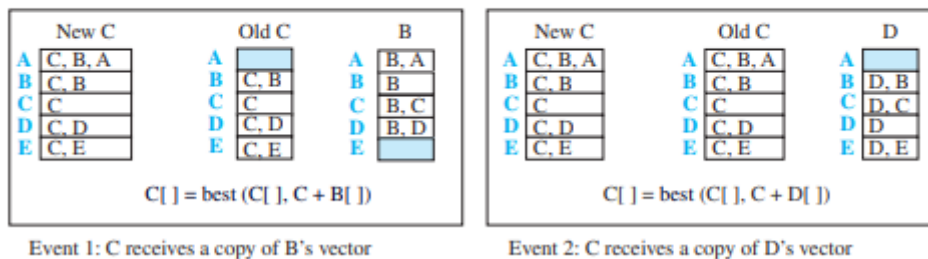


Each node, after the creation of the initial path vector, sends it to all its immediate neighbors. Each node, when it receives a path vector from a neighbor, updates its path vector using an equation similar to the Bellman-Ford, but applying its own policy instead of looking for the least cost

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

Updating Path Vectors

Note:
 X []: vector X
 Y: node Y



In the first event, node C receives a copy of B's vector, which improves its vector: now it knows how to reach node A. In the second event, node C receives a copy of D's vector, which does not change its vector.

UNICAST ROUTING PROTOCOLS

1. Routing Information Protocol (RIP)

The Routing Information Protocol (RIP) is one of the most widely used intradomain routing protocols based on the distance-vector routing algorithm.

Hop Count

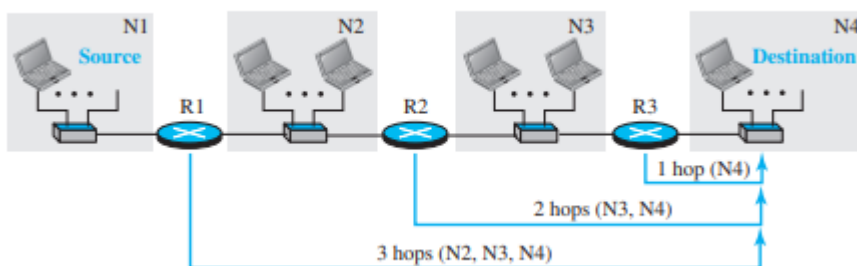


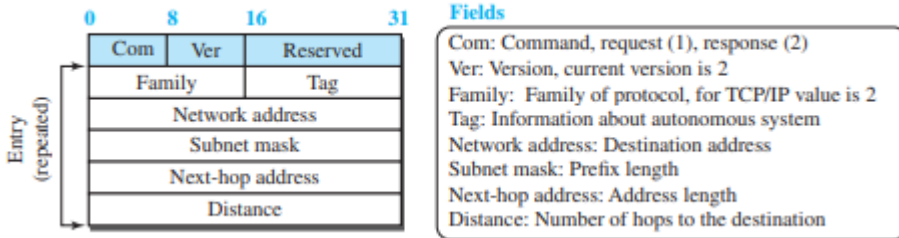
Figure shows the concept of hop count advertised by three routers from a source host to a destination host. In RIP, the maximum cost of a path can be 15, which means 16 is considered as infinity (no connection). For this reason, RIP can be used only in autonomous systems in which the diameter of the AS is not more than 15 hops.

Forwarding Tables

Forwarding table for R1			Forwarding table for R2			Forwarding table for R3		
Destination network	Next router	Cost in hops	Destination network	Next router	Cost in hops	Destination network	Next router	Cost in hops
N1	—	1	N1	R1	2	N1	R2	3
N2	—	1	N2	—	1	N2	R2	2
N3	R2	2	N3	—	1	N3	—	1
N4	R2	3	N4	R3	2	N4	—	1

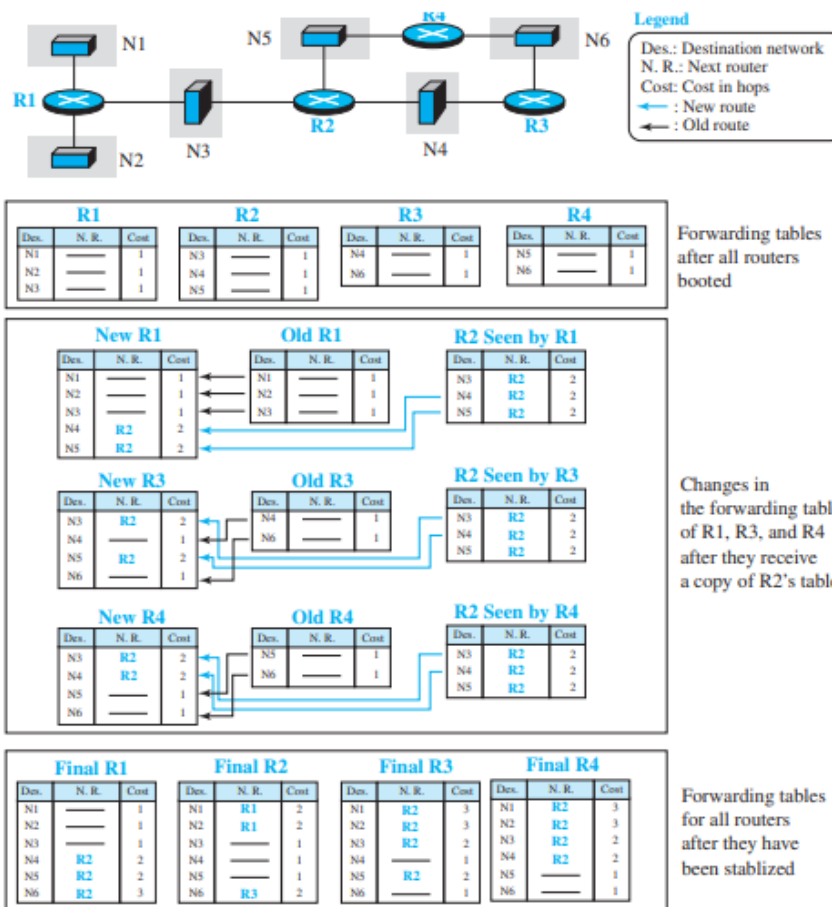
A forwarding table in RIP is a three-column table in which the first column is the address of the destination network, the second column is the address of the next router to which the packet should be forwarded, and the third column is the cost (the number of hops) to reach the destination network.

RIP Messages



RIP Algorithm

- Instead of sending only distance vectors, a router needs to send the whole contents of its forwarding table in a response message.
- The receiver adds one hop to each cost and changes the next router field to the address of the sending router. We call each route in the modified forwarding table the received route and each route in the old forwarding table the old route.



The received router selects the old routes as the new ones except in the following three cases:

1. If the received route does not exist in the old forwarding table, it should be added to the route.
2. If the cost of the received route is lower than the cost of the old one, the received route should be selected as the new one.
3. If the cost of the received route is higher than the cost of the old one, but the value of the next router is the same in both routes, the received route should be selected as the new one. This is the case where the route was actually advertised by the same router in the past, but now the situation has been changed. For example, suppose a neighbor has previously advertised a route to a destination with cost 3, but now there is no path between this neighbor and that destination. The neighbor advertises this destination with cost value infinity (16 in RIP). The receiving router must not ignore this value even though its old route has a lower cost to the same destination.

- The new forwarding table needs to be sorted according to the destination route (mostly using the longest prefix first).

Timers in RIP

The periodic timer

Each router has one periodic timer that is randomly set to a number between 25 and 35 seconds. The timer counts down; when zero is reached, the update message is sent, and the timer is randomly set once again.

The expiration timer

When a router receives update information for a route, the expiration timer is set to 180 seconds for that particular route. Every time a new update for the route is received, the timer is reset. If there is a problem on an internet and no update is received within the allotted 180 seconds, the route is considered expired and the hop count of the route is set to 16, which means the destination is unreachable.

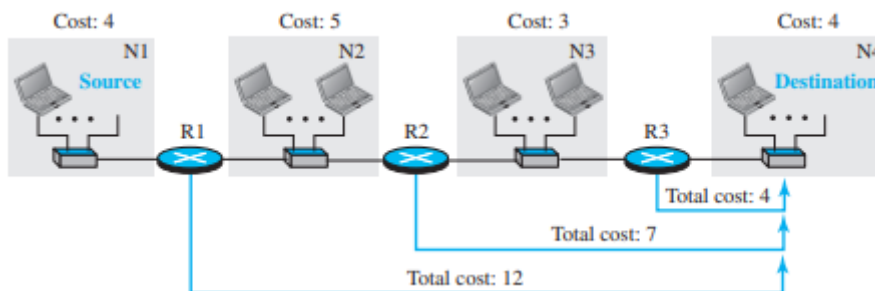
The garbage collection timer is used to purge a route from the forwarding table. When the information about a route becomes invalid, the router does not immediately purge that route from its table. Instead, it continues to advertise the route with a metric value of 16. At the same time, a garbage collection timer is set to 120 seconds for that route. When the count reaches zero, the route is purged from the table.

2. Open Shortest Path First (OSPF)

Open Shortest Path First (OSPF) is also an intradomain routing protocol like RIP, but it is based on the link-state routing protocol.

Metric

In OSPF, like RIP, the cost of reaching a destination from the host is calculated from the source router to the destination network. However, each link (network) can be assigned a weight based on the throughput, round-trip time, reliability, and so on. An administration can also decide to use the hop count as the cost.



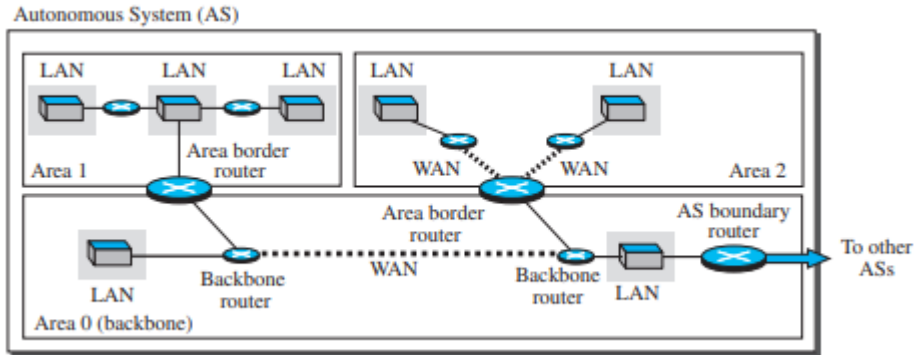
Forwarding Tables

Each OSPF router can create a forwarding table after finding the shortest-path tree between itself and the destination using Dijkstra's algorithm.

Forwarding table for R1			Forwarding table for R2			Forwarding table for R3		
Destination network	Next router	Cost	Destination network	Next router	Cost	Destination network	Next router	Cost
N1	—	4	N1	R1	9	N1	R2	12
N2	—	5	N2	—	5	N2	R2	8
N3	R2	8	N3	—	3	N3	—	3
N4	R2	12	N4	R3	7	N4	—	4

Areas

Flooding in OSPF may not create a problem in a small AS, it may have created a huge volume of traffic in a large AS. To prevent this, the AS needs to be divided into small sections called areas.



One of the areas in the AS is designated as the backbone area, responsible for gluing the areas together. The routers in the backbone area are responsible for passing the information collected by each area to all other areas. In this way, a router in an area can receive all LSPs generated in other areas. For the purpose of communication, each area has an area identification. The area identification of the backbone is zero.

Link-State Advertisement

There are five types of link state advertisements in OSPF -router link, network link, summary link to network, summary link to AS border router, and external link.

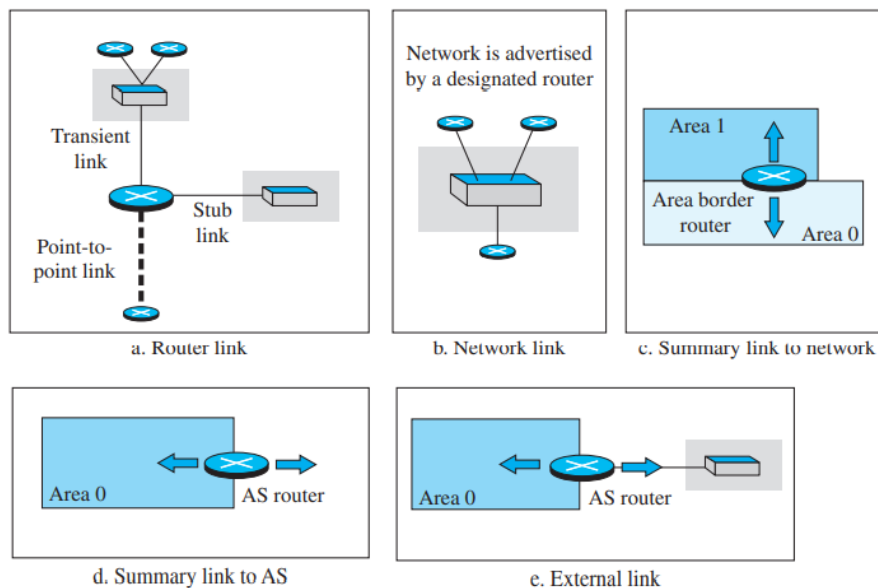
1. Router link

A router link advertises the existence of a router as a node. A transient link announces a link to a transient network, a network that is connected to the rest of the networks by one or more routers. A stub link advertises a link to a stub network, a network that is not a through network. A point-to-point link should define the address of the router at the end of the point-to-point line.

2. Network link.

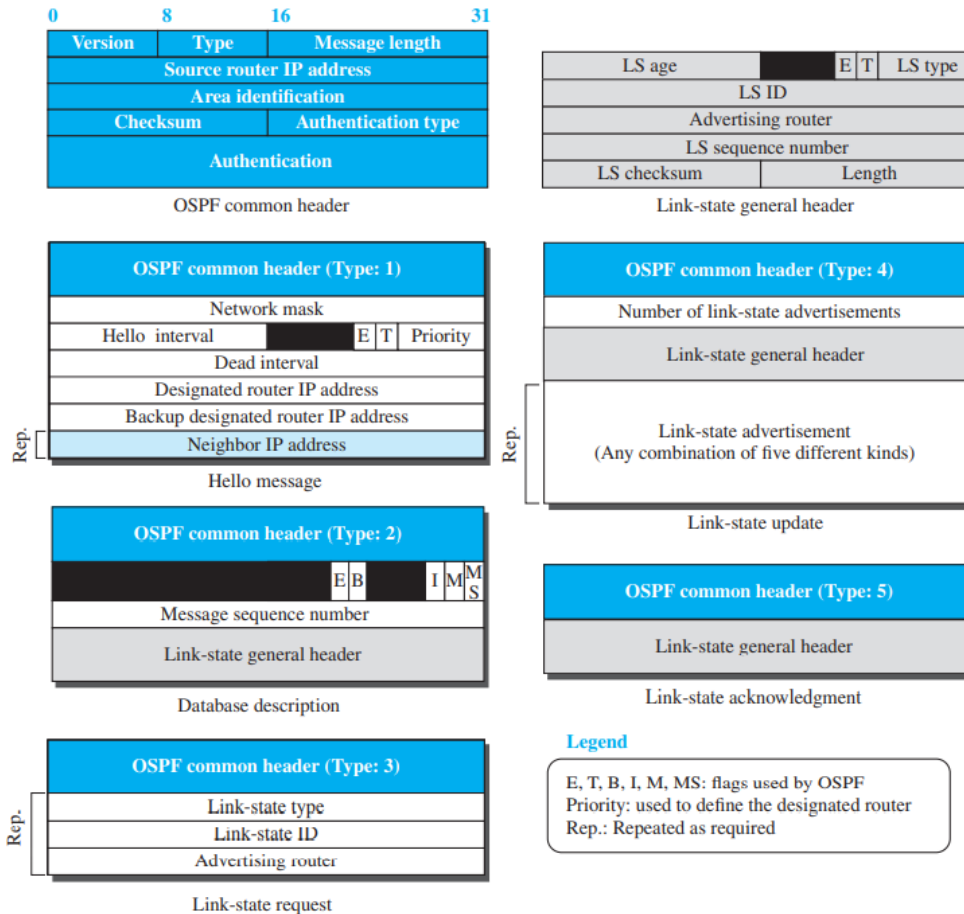
A network link advertises the network as a node. However, since a network cannot do announcements itself (it is a passive entity), one of the routers is assigned as the designated router and does the advertising.

3. **Summary link to network.** This is done by an area border router; it advertises the summary of links collected by the backbone to an area or the summary of links.
4. **Summary link to AS.** This is done by an AS router that advertises the summary links from other ASs to the backbone area of the current AS, information which later can be disseminated to the areas so that they will know about the networks in another ASs.
5. **External link.** This is also done by an AS router to announce the existence of a single network outside the AS to the backbone area to be disseminated into the areas.



OSPF Messages;

The **hello message** (type 1) is used by a router to introduce itself to the neighbors and announce all neighbors that it already knows. The **database description message** (type 2) is normally sent in response to the hello message to allow a newly joined router to acquire the full LSDB. The **link state request message** (type 3) is sent by a router that needs information about a specific LS. The **link-state update message** (type 4) is the main OSPF message used for building the LSDB. The **link-state acknowledgment message** (type 5) is used to create reliability in OSPF; each router that receives a link-state update message needs to acknowledge it.



OSPF Algorithm

OSPF implements the link-state routing algorithm.

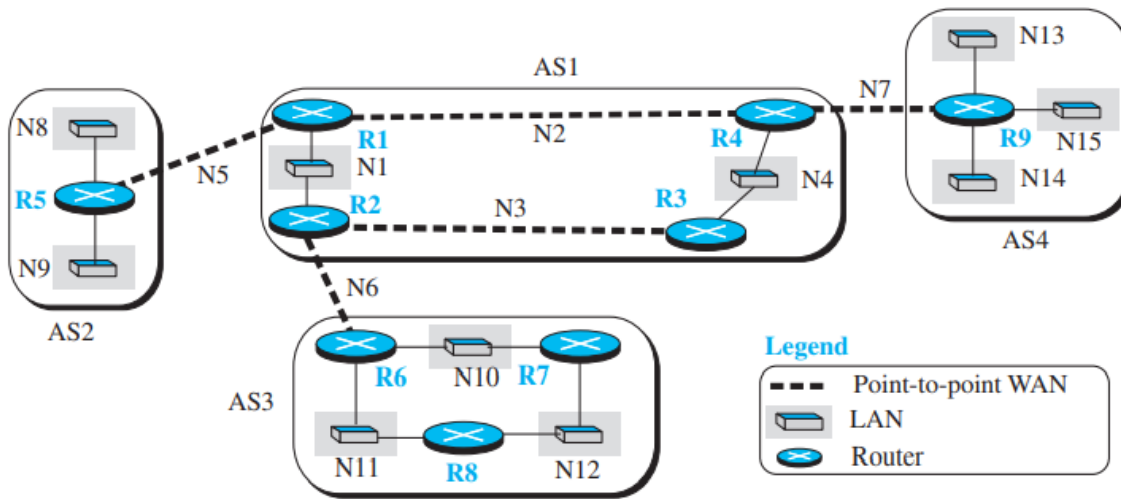
- After each router has created the shortest-path tree, the algorithm needs to use it to create the corresponding routing algorithm.
- The algorithm needs to be augmented to handle sending and receiving all five types of messages.

3. Border Gateway Protocol Version 4 (BGP4)

Explain few characteristics of BGP (13 marks nov/Dec 2019)

Give a routing method used for interdomain communication. (5 marks nov/dec 2021)

The Border Gateway Protocol version 4 (BGP4) is the only interdomain routing protocol used in the Internet today. BGP4 is based on the path-vector algorithm.

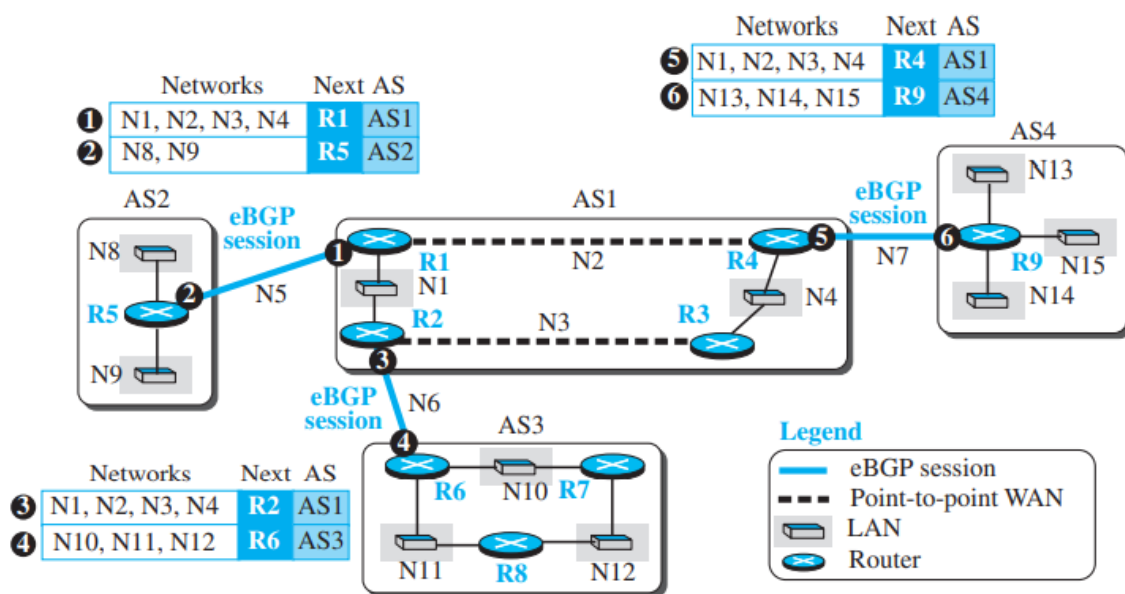


- Figure shows an example of an internet with four autonomous systems. AS2, AS3, and AS4 are stub autonomous systems; AS1 is a transit one. In this example, data exchange between AS2, AS3, and AS4 should pass through AS1.
- Each autonomous system in this figure uses one of the two common intradomain protocols, RIP or OSPF. Each router in each AS knows how to reach a network that is in its own AS, but it does not know how to reach a network in another AS.
- To enable each router to route a packet to any network in the internet, we first install a variation of BGP4, called external BGP (eBGP), on each border router.
- We then install the second variation of BGP, called internal BGP (iBGP), on all routers.
- This means that the border routers will be running three routing protocols (intradomain, eBGP, and iBGP), but other routers are running two protocols (intradomain and iBGP).

Operation of External BGP (eBGP)

- The two routers that run the BGP processes are called BGP peers or BGP speakers.
- The eBGP variation of BGP allows two physically connected border routers in two different ASs to form pairs of eBGP speakers and exchange messages.

The routers that are eligible in our example in Figure form three pairs: R1-R5, R2-R6, and R4- R9. Each logical connection is referred to as a session.



For example, message number 1 is sent by router R1 and tells router R5 that N1, N2, N3, and N4 can be reached through router R1. Router R5 can now add these pieces of information at the end of its forwarding table. When R5 receives any packet destined for these four networks, it can use its

forwarding table and find that the next router is R1.

There are two problems that need to be addressed:

1. Some border routers do not know how to route a packet destined for non neighbor ASs. For example, R5 does not know how to route packets destined for networks in AS3 and AS4.
2. None of the non-border routers know how to route a packet destined for any networks in other ASs.

To address the above two problems, we need to allow all pairs of routers (border or nonborder) to run the second variation of the BGP protocol, iBGP.

Operation of Internal BGP (iBGP)

- It creates a session between any possible pair of routers inside an autonomous system.
- If an AS has only one router, there cannot be an iBGP session. For example, we cannot create an iBGP session inside AS2 or AS4 in our internet.
- If there are n routers in an autonomous system, there should be $[n \times (n - 1) / 2]$ iBGP sessions in that autonomous system (a fully connected mesh) to prevent loops in the system.

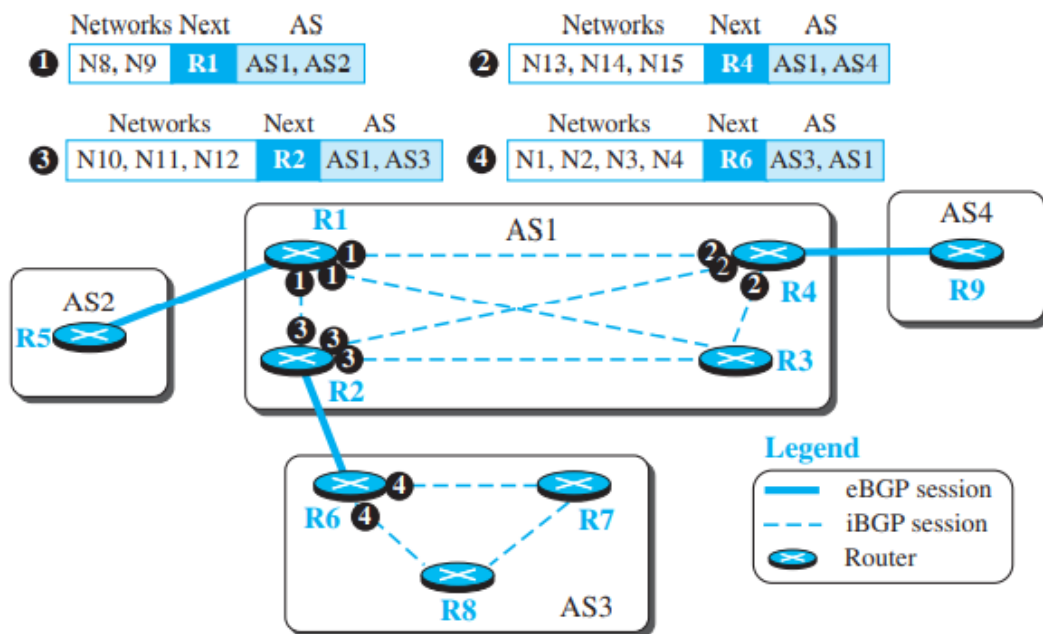


Figure shows the combination of eBGP and iBGP sessions in our internet.

- The first message (numbered 1) is sent by R1 announcing that networks N8 and N9 are reachable through the path AS1-AS2, but the next router is R1. This message is sent, through separate sessions, to R2, R3, and R4. Routers R2, R4, and R6 do the same thing but send different messages to different destinations.
- The updating process does not stop here. For example, after R1 receives the update message from R2, it combines the reachability information about AS3 with the reachability information it already knows about AS1 and sends a new update message to R5.
- At this time, each router combines the information received from eBGP and iBGP and creates a path table.

Networks	Next	Path	Networks	Next	Path	Networks	Next	Path
N8, N9	R5	AS1, AS2	N8, N9	R1	AS1, AS2	N8, N9	R2	AS1, AS2
N10, N11, N12	R2	AS1, AS3	N10, N11, N12	R6	AS1, AS3	N10, N11, N12	R2	AS1, AS3
N13, N14, N15	R4	AS1, AS4	N13, N14, N15	R1	AS1, AS4	N13, N14, N15	R4	AS1, AS4

Path table for R1 Path table for R2 Path table for R3

Networks	Next	Path	Networks	Next	Path	Networks	Next	Path
N8, N9	R1	AS1, AS2	N1, N2, N3, N4	R1	AS2, AS1	N1, N2, N3, N4	R2	AS3, AS1
N10, N11, N12	R1	AS1, AS3	N10, N11, N12	R1	AS2, AS1, AS3	N8, N9	R2	AS3, AS1, AS2
N13, N14, N15	R9	AS1, AS4	N13, N14, N15	R1	AS2, AS1, AS4	N13, N14, N15	R2	AS3, AS1, AS4

Path table for R4 Path table for R5 Path table for R6

Networks	Next	Path	Networks	Next	Path	Networks	Next	Path
N1, N2, N3, N4	R6	AS3, AS1	N1, N2, N3, N4	R6	AS3, AS1	N1, N2, N3, N4	R4	AS4, AS1
N8, N9	R6	AS3, AS1, AS2	N8, N9	R6	AS3, AS1, AS2	N8, N9	R4	AS4, AS1, AS2
N13, N14, N15	R6	AS3, AS1, AS4	N13, N14, N15	R6	AS3, AS1, AS4	N10, N11, N12	R4	AS4, AS1, AS3

Path table for R7 Path table for R8 Path table for R9

Fig. Finalized BGP Path Tables

Injection of Information into Intradomain Routing

The path tables collected and organized by BPG are not used, per se, for routing packets; they are injected into intradomain forwarding tables (RIP or OSPF) for routing packets.

Des.	Next	Cost	Des.	Next	Cost	Des.	Next	Cost	Des.	Next	Cost
N1	—	1	N1	—	1	N1	R2	2	N1	R1	2
N4	R4	2	N4	R3	2	N4	—	1	N4	—	1
N8	R5	1	N8	R1	2	N8	R2	3	N8	R1	2
N9	R5	1	N9	R1	2	N9	R2	3	N9	R1	2
N10	R2	2	N10	R6	1	N10	R2	2	N10	R3	3
N11	R2	2	N11	R6	1	N11	R2	2	N11	R3	3
N12	R2	2	N12	R6	1	N12	R2	2	N12	R3	3
N13	R4	2	N13	R3	3	N13	R4	2	N13	R9	1
N14	R4	2	N14	R3	3	N14	R4	2	N14	R9	1
N15	R4	2	N15	R3	3	N15	R4	2	N15	R9	1

Table for R1 Table for R2 Table for R3 Table for R4

Des.	Next	Cost	Des.	Next	Cost	Des.	Next	Cost	Des.	Next	Cost
N8	—	1	N10	—	1	N10	—	1	N13	—	1
N9	—	1	N11	—	1	N11	R6	2	N14	—	1
0	R1	1	N12	R7	2	N12	—	1	N15	—	1
			0	R2	1	0	R6	2	0	R4	1

Table for R5 Table for R6 Table for R7 Table for R8 Table for R9

Fig Forwarding table after injection from BGP

Path Attributes

Interdomain routing needs more information about how to reach the final destination. In BGP these pieces are called path attributes.

Path attributes are divided into two broad categories: well-known and optional. A well-known attribute must be recognized by all routers; an optional attribute need not be. A well-known attribute can be mandatory, which means that it must be present in any BGP update message. An optional attribute can be either transitive, which means it can pass to the next AS, or intransitive, which means it cannot.

There are seven attributes:

ORIGIN (type 1). This is a well-known mandatory attribute, which defines the source of the routing information. This attribute can be defined by one of the three values: 1, 2, and 3. Value 1 means that

the information about the path has been taken from an intradomain protocol (RIP or OSPF). Value 2 means that the information comes from BGP. Value 3 means that it comes from an unknown source.

AS-PATH (type 2). This is a well-known mandatory attribute, which defines the list of autonomous systems through which the destination can be reached.

NEXT-HOP (type 3). This is a well-known mandatory attribute, which defines the next router to which the data packet should be forwarded.

MULT-EXIT-DISC (type 4). The multiple-exit discriminator is an optional intransitive attribute, which discriminates among multiple exit paths to a destination. The value of this attribute is normally defined by the metric in the corresponding intradomain protocol.

LOCAL-PREF (type 5). The local preference attribute is a well-known discretionary attribute. It is normally set by the administrator, based on the organization policy. The routes the administrator prefers are given a higher local preference value.

ATOMIC-AGGREGATE (type 6). This is a well-known discretionary attribute, which defines the destination prefix as not aggregate; it only defines a single destination network.

AGGREGATOR (type 7). This is an optional transitive attribute, which emphasizes that the destination prefix is an aggregate.

Route Selection

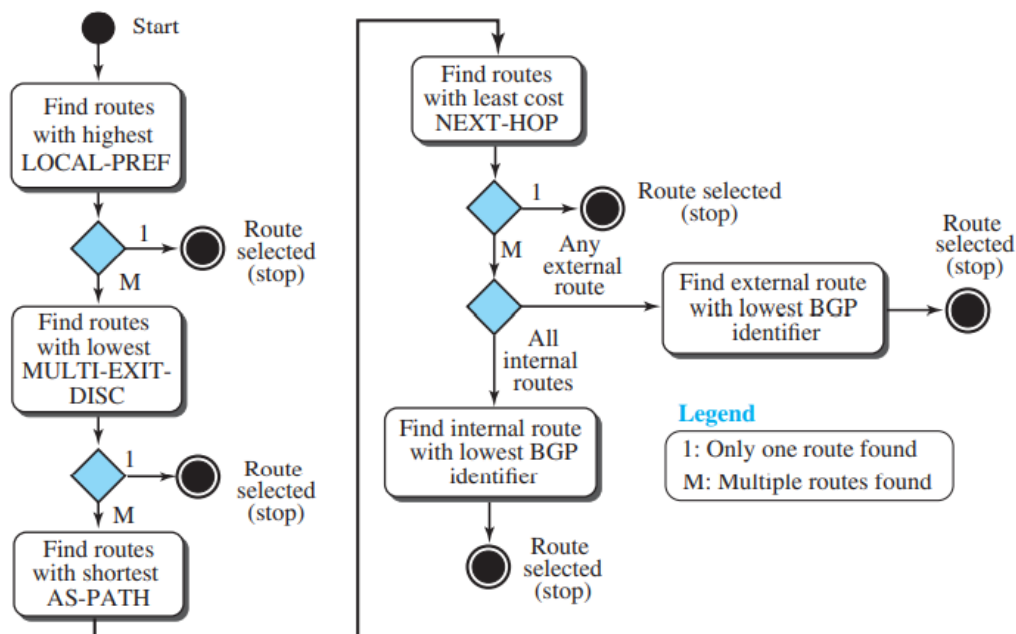


Fig Flow diagram of route selection

Messages

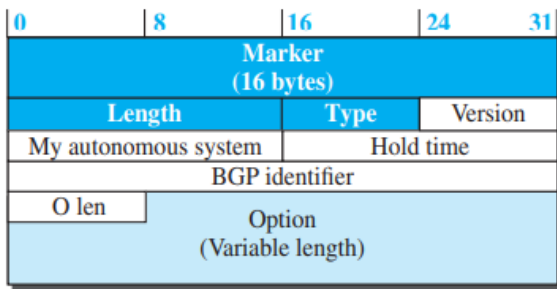
BGP uses four types of messages for communication between the BGP speakers across the ASs and inside an AS: open, update, keepalive, and notification. All BGP packets share the same common header.

Open Message. To create a neighborhood relationship, a router running BGP opens a TCP connection with a neighbor and sends an open message.

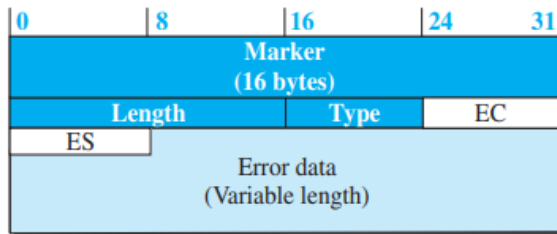
Update Message. The update message is the heart of the BGP protocol. It is used by a router to withdraw destinations that have been advertised previously, to announce a route to a new destination, or both.

Keepalive Message. The BGP peers that are running exchange keepalive messages regularly (before their hold time expires) to tell each other that they are alive.

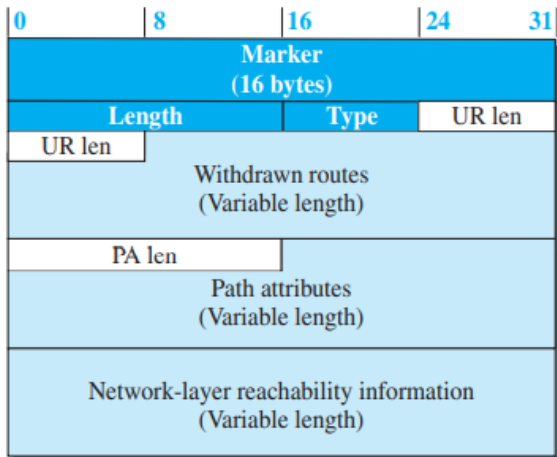
Notification. A notification message is sent by a router whenever an error condition is detected or a router wants to close the session.



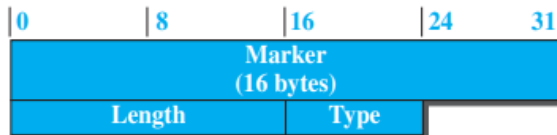
Open message (type 1)



Notification message (type 3)



Update message (type 2)



Keepalive message (type 4)

Fields in common header

Marker: Reserved for authentication
 Length: Length of total message in bytes
 Type: Type of message (1 to 4)

Abbreviations

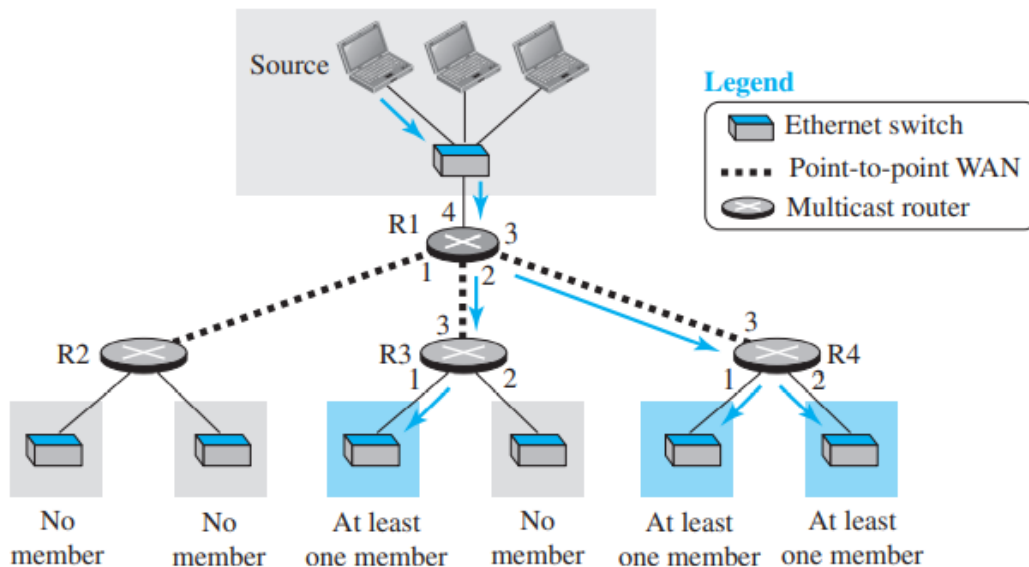
O len: Option length
 EC: Error code
 ES: Error subcode
 UR len: Unfeasible route length
 PA len: Path attribute length

Multicast Routing

Mention the need for multicasting (2 marks nov/dec 2021)

Discuss the role of multicast routing and its relative merits (8 marks nov/dec 2021)

In multicasting, there is one source and a group of destinations. The relationship is one to many. In this type of communication, the source address is a unicast address, but the destination address is a group address, a group of one or more destination networks in which there is at least one member of the group that is interested in receiving the multicast datagram.

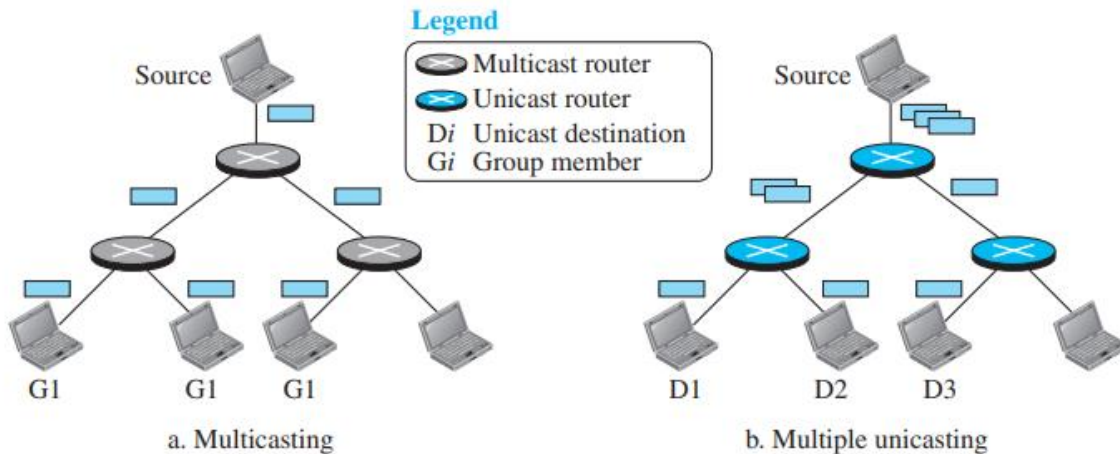


In multicasting, a multicast router may have to send out copies of the same datagram through more than one interface. In Figure router R1 needs to send out the datagram through interfaces 2 and 3. Similarly, router R4 needs to send out the datagram through both its interfaces. Router R3, however, knows that there is no member belonging to this group in the area reached by interface 2; it only

sends out the datagram through interface 1.

Multicasting versus Multiple Unicasting

Explain the difference between multicasting and multiple unicasting (2 marks nov/dec 2020)



Multicasting starts with a single packet from the source that is duplicated by the routers. The destination address in each packet is the same for all duplicates. Note that only a single copy of the packet travels between any two routers.

In **multiple unicasting**, several packets start from the source. If there are three destinations, for example, the source sends three packets, each with a different unicast destination address. Note that there may be multiple copies traveling between two routers. For example, when a person sends an e-mail message to a group of people, this is multiple unicasting. The e-mail application software creates replicas of the message, each with a different destination address, and sends them one by one.

Advantages of Multicasting:

1. Multicasting is more efficient than multiple unicasting. multicasting requires less bandwidth than multiple unicasting. In multiple unicasting, some of the links must handle several copies.
2. In multiple unicasting, the packets are created by the source with a relative delay between packets. If there are 1,000 destinations, the delay between the first and the last packet may be unacceptable. In multicasting, there is no delay because only one packet is created by the source.

MULTICASTING BASICS

1. Multicast Addresses:

In multicast communication, the sender is only one, but the receiver is many, sometimes thousands or millions spread all over the world. It should be clear that we cannot include the addresses of all recipients in the packet. The destination addresses of a packet, as described in the Internet Protocol (IP) should be only one. For this reason, we need multicast addresses.

A multicast address defines a group of recipients, not a single one. The source address of a packet in multicast communication can be a unicast address that uniquely defines the sender, but the destination address can be the multicast address that defines a group. In this way, a host, which is a member of n groups, actually has $(n + 1)$ addresses: one unicast address that is used for source or destination address in unicast communication and n multicast addresses that are used only for destination addresses to receive messages sent to a group.

2. Collecting Information about Groups:

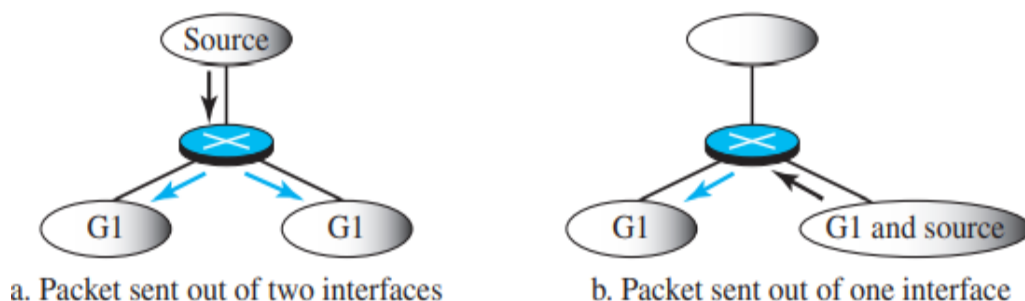
Creation of forwarding tables in both unicast and multicast routing involves two steps:

1. A router needs to know to which destinations it is connected.
2. Each router needs to propagate information obtained in the first step to all other routers so that each router knows to which destination each other router is connected.

- In unicast routing, the collection of the information in the first step is automatic; each router knows to which network it is connected. The routing protocols distance-vector or link-state are responsible for propagating these automatically collected pieces of information to each other router in the internet.
- In multicast routing, the collection of information in the first step is not automatic for two reasons. First, a router does not know which host in the attached network is a member of a particular group. If a host is a member of a group, it has a separate multicast address related to that group. Second, the membership is not a fixed attribute of a host; a host may join some new groups and leave some others even in a short period of time. For this reason, a router needs help to find out which groups are active in each of its interfaces. After collecting these pieces of information, a router can propagate the membership to any other router using a multicast routing protocol.
- In other words, for unicast routing, we need only the routing protocol inside each domain to propagate the information about a router link; in multicasting we need two protocols: one to collect these pieces of information and the second to propagate them.

3. Multicast Forwarding:

- In unicast communication, the destination address of the packet defines one single destination. The packet needs to be sent only out of one of the interfaces. In multicast communication, the destination of the packet defines one group, but that group may have more than one member in the internet. To reach all of the destinations, the router may have to send the packet out of more than one interface.
- Forwarding decisions in unicast communication depend only on the destination address of the packet. Forwarding decisions in multicast communication depend on both the destination and the source address of the packet.



- In part a of the figure, the source is in a section of the internet where there is no group member. In part b, the source is in a section where there is a group member. In part a, the router needs to send out the packet from two interfaces; in part b, the router should send the packet only from one interface to avoid sending a second copy of the packet from the interface it has arrived at. In other words, in part b of the figure, the member or members of the group G1 have already received a copy of the packet when it arrives at the router; sending out the packet in that direction does not help, but creates more traffic. This shows that the forwarding in multicast communication depends on both source and destination addresses.

4. Two Approaches to Multicasting:

Two different approaches in multicast routing have been developed: routing using source-based trees and routing using group-shared trees.

Source-Based Tree Approach

In the source-based tree approach to multicasting, each router needs to create a separate tree for each source-group combination. In other words, if there are m groups and n sources in the internet, a router needs to create $(m \times n)$ routing trees. In each tree, the corresponding source is the root, the members of the group are the leaves, and the router itself is somewhere on the tree.

Group-Shared Tree Approach

In the group-shared tree approach, the designated router, which is called the core router or the rendezvous point router, acts as the representative for the group. Any source that has a packet to send to a member of that group sends it to the core center (unicast communication) and the core center is responsible for multicasting. The core center creates one single routing tree with itself as the root and any routers with active members in the group as the leaves. In this approach, there are m core routers (one for each group) and each core router has a routing tree, for the total of m trees. This means that the number of routing trees is reduced from $(m \times n)$ in the source-based tree approach to m in this approach.

INTRADOMAIN MULTICAST PROTOCOLS

Explain the steps used by DVMRP router to create a source-based tree
(7 marks nov/dec 2020)

1. Multicast Distance Vector (DVMRP)

The Distance Vector Multicast Routing Protocol (DVMRP) is the extension of the Routing Information Protocol (RIP) which is used in unicast routing. It uses the source-based tree approach to multicasting. Each router in this protocol that receives a multicast packet to be forwarded implicitly creates a source-based multicast tree in three steps:

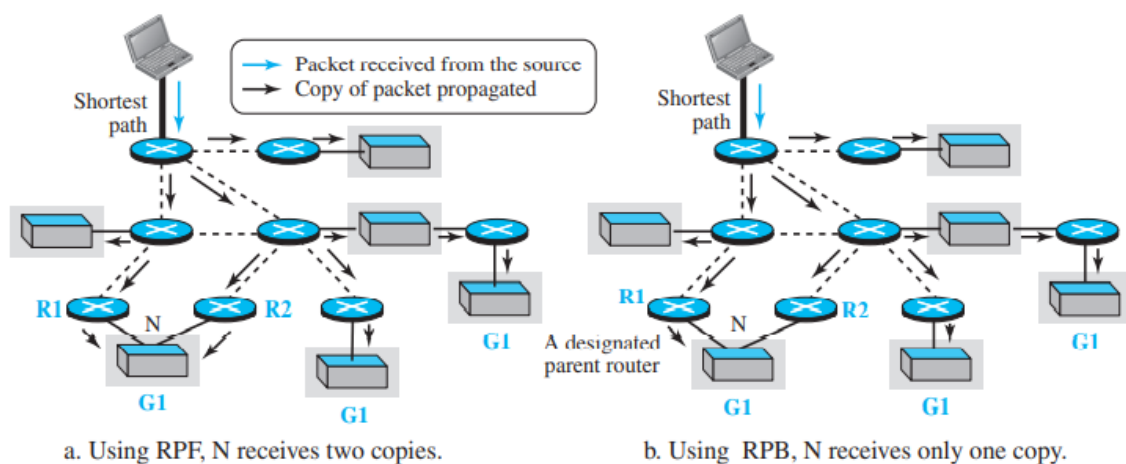
1. The router uses an algorithm called reverse path forwarding (RPF) to simulate creating part of the optimal source-based tree between the source and itself.
2. The router uses an algorithm called reverse path broadcasting (RPB) to create a broadcast (spanning) tree whose root is the router itself and whose leaves are all networks in the internet.
3. The router uses an algorithm called reverse path multicasting (RPM) to create a multicast tree by cutting some branches of the tree that end in networks with no member in the group.

Reverse Path Forwarding (RPF)

- The first algorithm, reverse path forwarding (RPF), forces the router to forward a multicast packet from one specific interface: the one which has come through the shortest path from the source to the router.
- The router does not know the shortest path from the source to itself, but it can find which is the next router in the shortest path from itself to the source (reverse path). The router simply consults its unicast forwarding table, pretending that it wants to send a packet to the source.
- The router uses this information to accept a multicast packet only if it arrives from this interface.
- The RPF algorithm helps a router to forward only one copy received from a source and drop the rest.

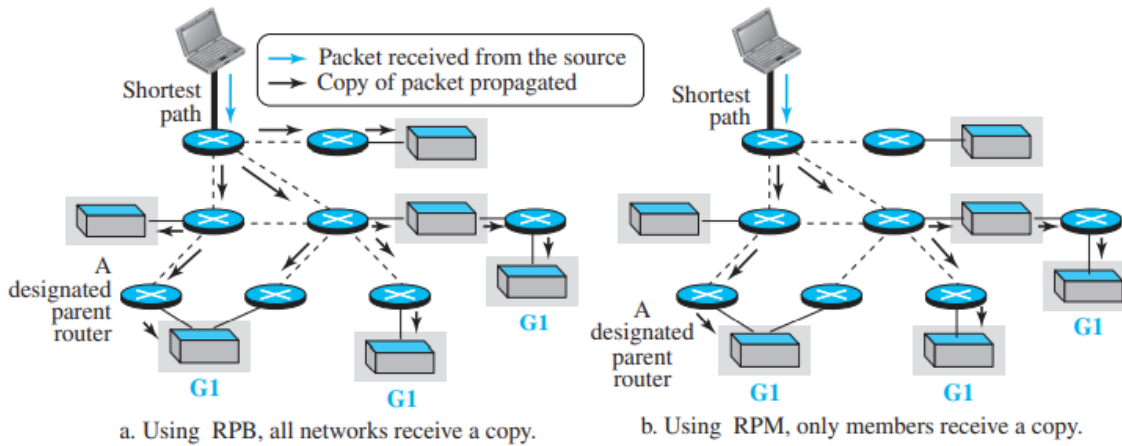
Reverse Path Broadcasting (RPB)

- If a network is connected to more than one router, it may receive a copy of the packet from each router. RPF cannot help here, because a network does not have the intelligence to apply the RPF algorithm.
- we need to allow only one of the routers attached to a network to pass the packet to the network. One way to do so is to designate only one router as the parent of a network related to a specific source. When a router that is not the parent of the attached network receives a multicast packet, it simply drops the packet.
- one way is to select the router that has the shortest path to the source (using the unicast forwarding table, again in the reverse direction). If there is a tie in this case, the router with the smaller IP address can be selected.
- Reverse path broadcasting (RPB) actually creates a broadcast tree from the graph that has been created by the RPF algorithm. RPB has cut those branches of the tree that cause cycles in the graph.



Reverse Path Multicasting (RPM)

- RPB does not multicast the packet, it broadcasts it. This is not efficient. To increase efficiency, the multicast packet must reach only those networks that have active members for that particular group. This is called reverse path multicasting (RPM).
- To change the broadcast shortest-path tree to a multicast shortest-path tree, each router needs to prune (make inactive) the interfaces that do not reach a network with active members corresponding to a particular source-group combination. This step can be done bottom-up, from the leaves to the root.
- At the leaf level, the routers connected to the network collect the membership information using the IGMP protocol. The parent router of the network can then disseminate this information upward using the reverse shortest-path tree from the router to the source, the same way as the distance vector messages are passed from one neighbour to another. When a router receives all of these membership-related messages, it knows which interfaces need to be pruned.



2. Multicast Link State (MOSPF)

- Multicast Open Shortest Path First (MOSPF) is the extension of the Open Shortest Path First (OSPF) protocol, which is used in unicast routing.
- It also uses the source-based tree approach to multicasting.

A router goes through the following steps to forward a multicast packet received from source S and to be sent to destination G (a group of recipients):

1. The router uses the Dijkstra algorithm to create a shortest-path tree with S as the root and all destinations in the internet as the leaves. Note that this shortest-path tree is different from the one the router normally uses for unicast forwarding, in which the root of the tree is the router itself. In this case, the root of the tree is the source of the packet defined in the source address of the packet.
2. The router finds itself in the shortest-path tree created in the first step. In other words, the router creates a shortest-path subtree with itself as the root of the subtree.
3. The shortest-path subtree is actually a broadcast subtree with the router as the root and all networks as the leaves. The router now uses a strategy similar to the one we describe in the case of DVMRP to prune the broadcast tree and to change it to a multicast tree.
4. The router can now forward the received packet out of only those interfaces that correspond to the branches of the multicast tree. We need to make certain that a copy of the multicast packet reaches all networks that have active members of the group and that it does not reach those networks that do not.

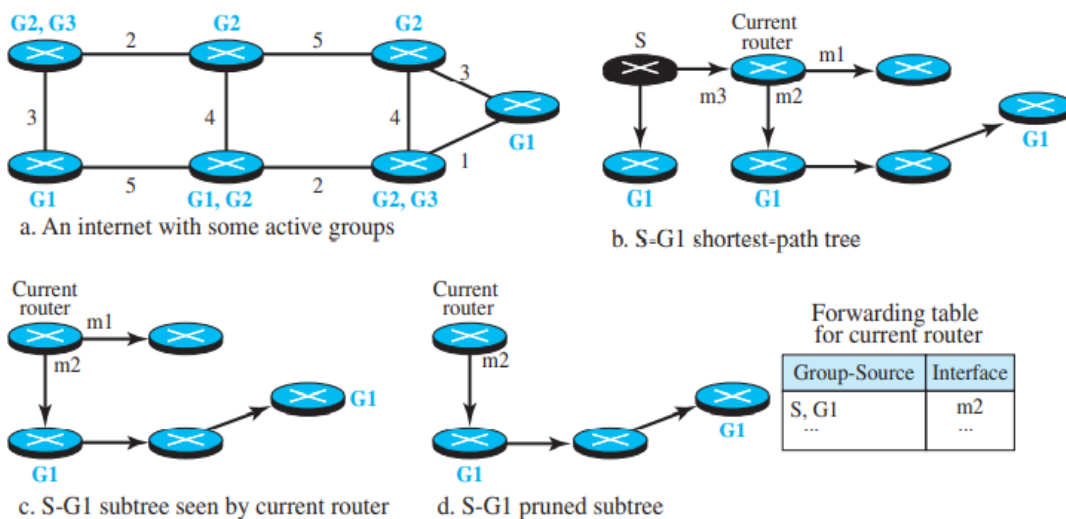


Fig Example of tree formation in MOSPF.

3. Protocol Independent Multicast (PIM)

- Protocol Independent Multicast (PIM) use the forwarding table of a unicast routing protocol to find the next router in a path to the destination, but it does not matter how the forwarding table is created.
- PIM can work in two different modes: dense and sparse. The term dense here means that the number of active members of a group in the internet is large. The term sparse, on the other hand, means that only a few routers in the internet have active members in the group.
- When the protocol is working in the dense mode, it is referred to as PIM-DM; when it is working in the sparse mode, it is referred to as PIM-SM.

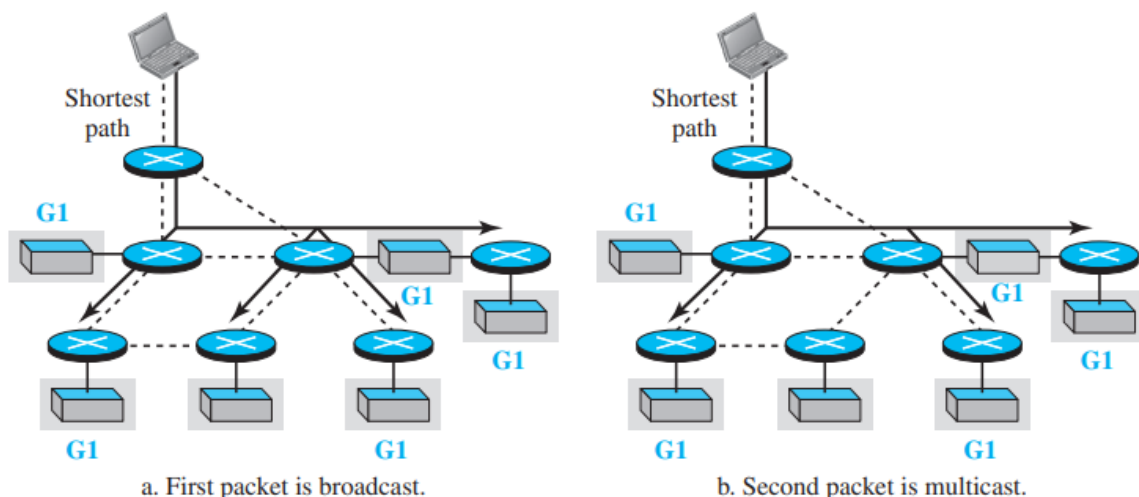
Protocol Independent Multicast-Dense Mode (PIM-DM)

- When the number of routers with attached members is large relative to the number of routers in the internet, PIM works in the dense mode and is called PIM-DM.
- In this mode, the protocol uses a source-based tree approach.

The two steps used in PIM-DM are:

1. A router that has received a multicast packet from the source S destined for the group G first uses the RPF strategy to avoid receiving a duplicate of the packet. It consults the forwarding table of the underlying unicast protocol to find the next router if it wants to send a message to the source S (in the reverse direction). If the packet has not arrived from the next router in the reverse direction, it drops the packet and sends a prune message in that direction to prevent receiving future packets related to (S, G).
2. If the packet in the first step has arrived from the next router in the reverse direction, the receiving router forwards the packet from all its interfaces except the one from which the packet has arrived and the interface from which it has already received a prune message related to (S, G). Note that this is actually a broadcasting instead of a multicasting if the packet is the first packet from the source S to group G.

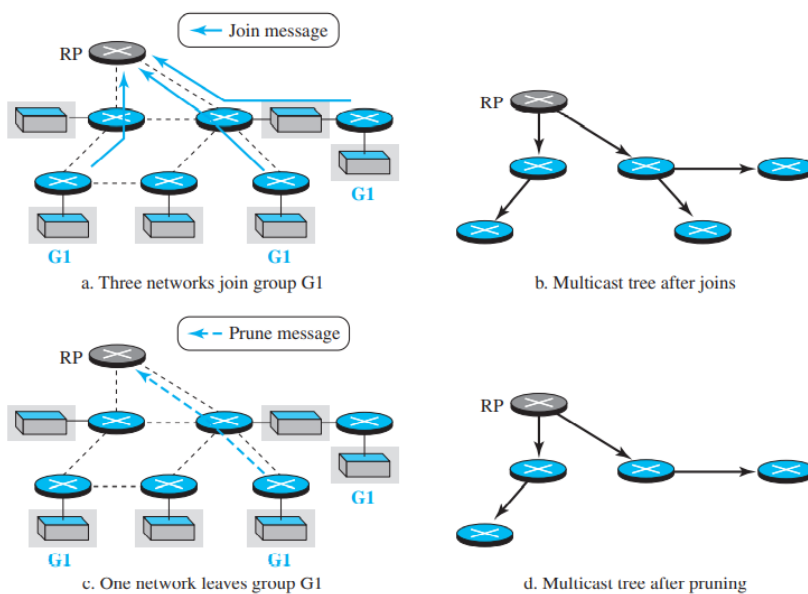
However, each router downstream that receives an unwanted packet sends a prune message to the router upstream, and eventually the broadcasting is changed to multicasting.



Protocol Independent Multicast-Sparse Mode (PIM-SM)

- When the number of routers with attached members is small relative to the number of routers in the internet, PIM works in the sparse mode and is called PIM-SM.
- PIM-SM uses a group-shared tree approach to multicasting. The core router in PIM-SM is called the rendezvous point (RP).

- Multicast communication is achieved in two steps. Any router that has a multicast packet to send to a group of destinations first encapsulates the multicast packet in a unicast packet (tunneling) and sends it to the RP. The RP then decapsulates the unicast packet and sends the multicast packet to its destination.
- PIM-SM uses a complex algorithm to select one router among all routers in the internet as the RP for a specific group. This means that if we have m active groups, we need m RPs, although a router may serve more than one group.
- After the RP for each group is selected, each router creates a database and stores the group identifier and the IP address of the RP for tunneling multicast packets to it.
- PIM-SM uses a spanning multicast tree rooted at the RP with leaves pointing to designated routers connected to each network with an active member.
- To create a multicast tree rooted at the RP, PIM-SM uses join and prune messages. Figure shows the operation of join and prune messages in PIM-SM. First, three networks join group G1 and form a multicast tree. Later, one of the networks leaves the group and the tree is pruned.



When a designated router finds out that a network has a new member in the corresponding group (via IGMP), it sends a join message in a unicast packet destined for the RP. The packet travels through the unicast shortest-path tree to reach the RP. Any router in the path receives and forwards the packet, but at the same time, the router adds two pieces of information to its multicast forwarding table. The number of the interface through which the join message has arrived is marked (if not already marked) as one of the interfaces through which the multicast packet destined for the group should be sent out in the future. The number of the interface through which the join message was sent to the RP is marked (if not already marked) as the only interface through which the multicast packet destined for the same group should be received.

To avoid sending multicast packets to networks with no members, PIM-SM uses the prune message. Each designated router that finds out (via IGMP) that there is no active member in its network, sends a prune message to the RP. When a router receives a prune message, it decrements the join count for the interface through which the message has arrived and forwards it to the next router. When the join count for an interface reaches zero, that interface is not part of the multicast tree anymore.

INTERDOMAIN MULTICAST PROTOCOLS

- When the members of the groups are spread among different domains (ASs), we need an interdomain multicast routing protocol.
- One common protocol for interdomain multicast routing is called **Multicast Border Gateway Protocol (MBGP)**, which is the extension of BGP.

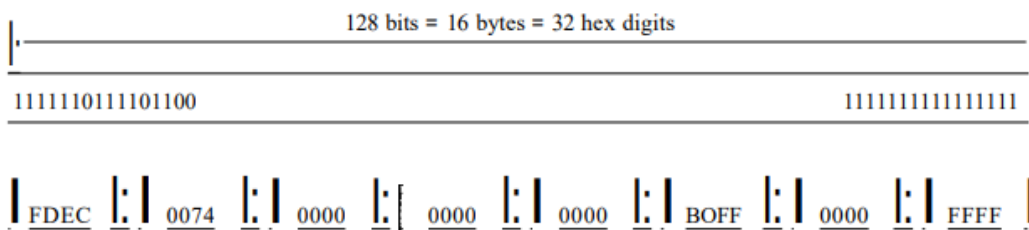
- MBGP provides two paths between ASs: one for unicasting and one for multicasting. Information about multicasting is exchanged between border routers in different ASs.
- MBGP is a shared-group multicast routing protocol in which one router in each AS is chosen as the rendezvous point (RP).
- The problem with MBGP protocol is that it is difficult to inform an RP about the sources of groups in other ASs.
- The **Multicast Source Discovery Protocol (MSDP)** is a new suggested protocol that assigns a source representative router in each AS to inform all RPs about the existence of sources in that AS.
- Another protocol is **Border Gateway Multicast Protocol (BGMP)**, which allows construction of shared group trees with a single root in one of the ASs. In other words, for each group, there is only one shared tree, with leaves in different ASs, but the root is located in one of the ASs.

IPv6 address

Explain IPV6 datagram format with suitable diagram (7 marks nov/dec 2020)

Define the term IPV6 and its advantages (2 marks nov/dec 2021)

- An IPv6 address consists of 16 bytes (octets); it is 128 bits long.
- To make addresses more readable, IPv6 specifies hexadecimal colon notation. In this notation, 128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon



- Although the IP address, even in hexadecimal format, is very long, many of the digits are zeros. In this case, we can abbreviate the address. The leading zeros of a section (four digits between two colons) can be omitted. Only the leading zeros can be dropped, not the trailing zeros.

Original

FDEC : 0074 : 0000 : 0000 : 0000 : BOFF : 0000 : FFF0

Abbreviated FDEC : 74 : 0 : 0 : 0 : BOFF : 0 : FFF0

More abbreviated FDEC : 74 : : BOFF : 0 : FFF0

↑
Gap

Address Space

IPv6 has a much larger address space; 2¹²⁸ addresses are available.

Three Address Types

In IPv6, a destination address can belong to one of three categories: unicast, anycast, and

multicast.

Unicast Address A unicast address defines a single interface (computer or router). The packet sent to a unicast address will be routed to the intended recipient.

Anycast Address An anycast address defines a group of computers that all share a single address. A packet with an anycast address is delivered to only one member of the group, the most reachable one.

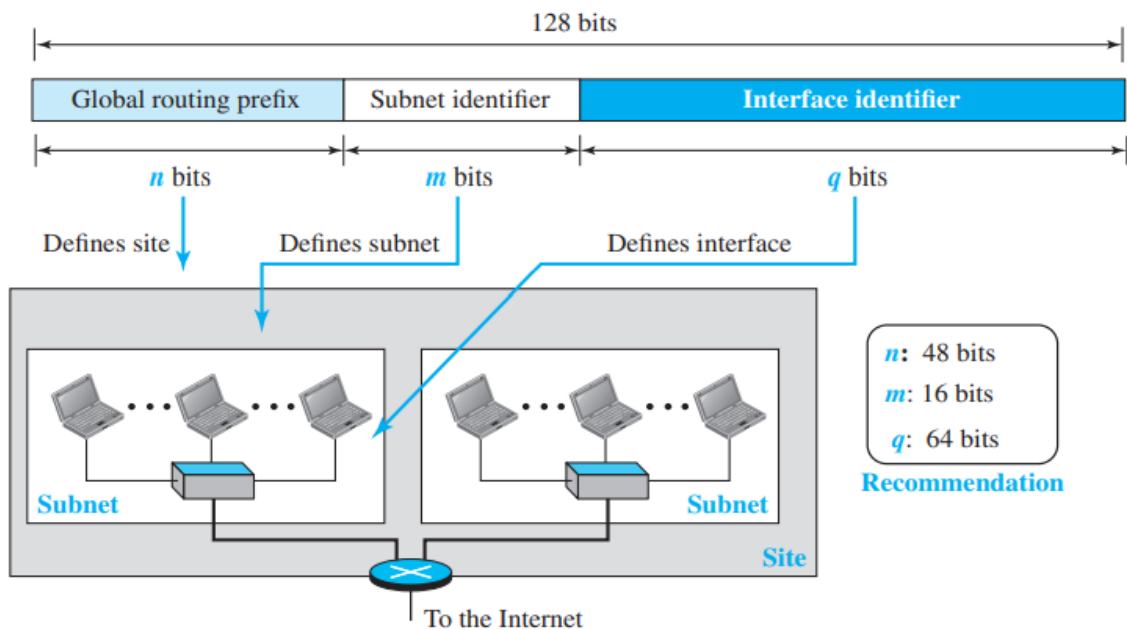
Multicast Address A multicast address also defines a group of computers. However, there is a difference between anycasting and multicasting. In anycasting, only one copy of the packet is sent to one of the members of the group; in multicasting each member of the group receives a copy.

Address Space Allocation

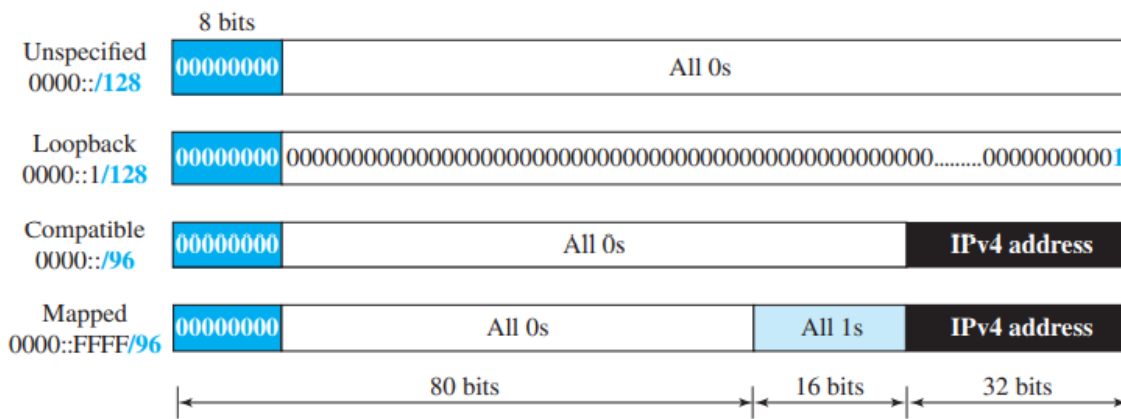
The address space of IPv6 is divided into several blocks of varying size and each block is allocated for a special purpose.

Block prefix	CIDR	Block assignment	Fraction
0000 0000	0000:: 8</td <td>Special addresses</td> <td>1/256</td>	Special addresses	1/256
001	2000::<!--3</b-->	Global unicast	1/8
1111 110	FC00:: 7</td <td>Unique local unicast</td> <td>1/128</td>	Unique local unicast	1/128
1111 1110 10	FE80:: 10</td <td>Link local addresses</td> <td>1/1024</td>	Link local addresses	1/1024
1111 1111	FF00:: 8</td <td>Multicast addresses</td> <td>1/256</td>	Multicast addresses	1/256

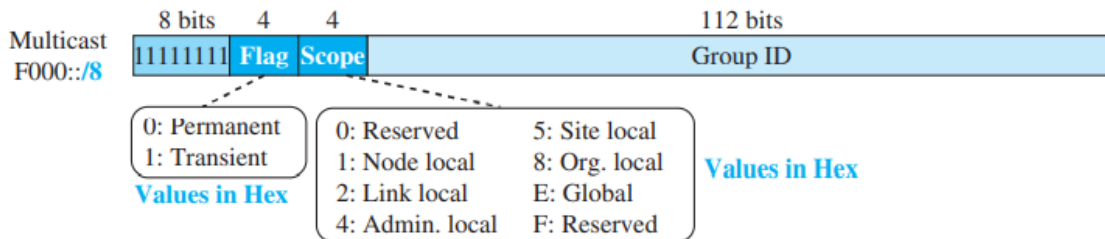
Global Unicast Addresses



Special Addresses



Multicast Address

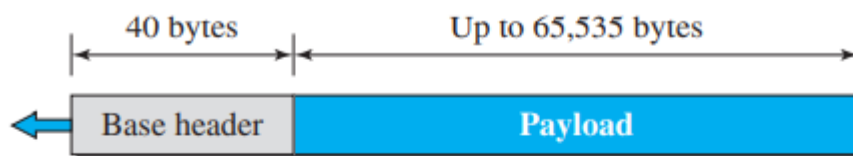


Autoconfiguration

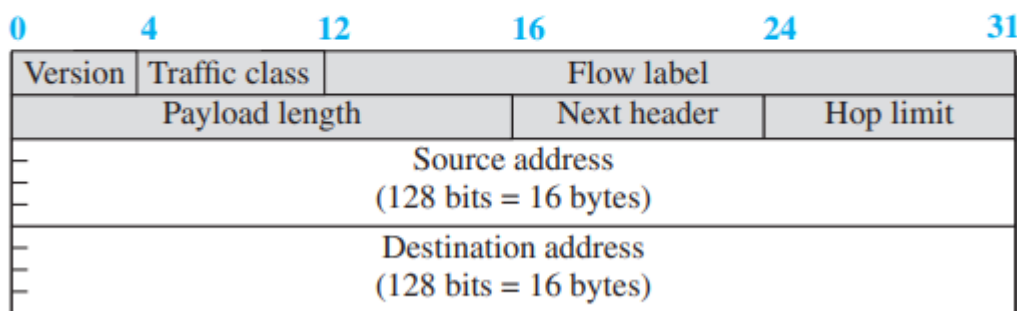
When a host in IPv6 joins a network, it can configure itself using the following process:

1. The host first creates a link local address for itself. This is done by taking the 10-bit link local prefix (1111 1110 10), adding 54 zeros, and adding the 64-bit interface identifier, which any host knows how to generate from its interface card. The result is a 128-bit link local address.
2. The host then tests to see if this link local address is unique and not used by other hosts. Since the 64-bit interface identifier is supposed to be unique, the link local address generated is unique with a high probability.
3. If the uniqueness of the link local address is passed, the host stores this address as its link local address (for private communication), but it still needs a global unicast address. The host then sends a router solicitation message to a local router.

IPv6 Packet Format



a. IPv6 packet



b. Base header

Version. The 4-bit version field defines the version number of the IP. For IPv6, the value is 6.

Traffic class. The 8-bit traffic class field is used to distinguish different payloads with different delivery requirements.

Flow label. The flow label is a 20-bit field that is designed to provide special handling for a particular flow of data.

Payload length. The 2-byte payload length field defines the length of the IP datagram excluding the header.

Next header. The next header is an 8-bit field defining the type of the first extension header (if present) or the type of the data that follows the base header in the datagram.

Hop limit. The 8-bit hop limit field serves the same purpose as the TTL field in IPv4.

Source and destination addresses. The source address field is a 16-byte (128-bit) Internet address that identifies the original source of the datagram. The destination address field is a 16-byte (128-bit) Internet address that identifies the destination of the datagram.

payload in IPv6 means a combination of zero or more extension headers (options) followed by the data from other protocols (UDP, TCP, and so on).

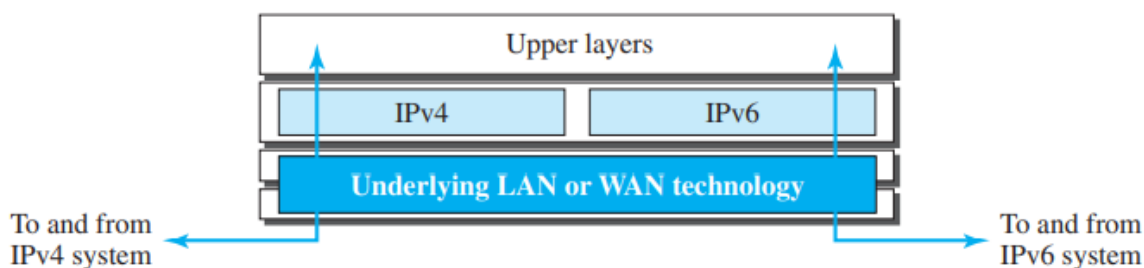
TRANSITION FROM IPv4 TO IPv6

Discuss the merits of IPv4 and IPv6 and needs for transition to IPv6 (13 marks nov/dec 2021)

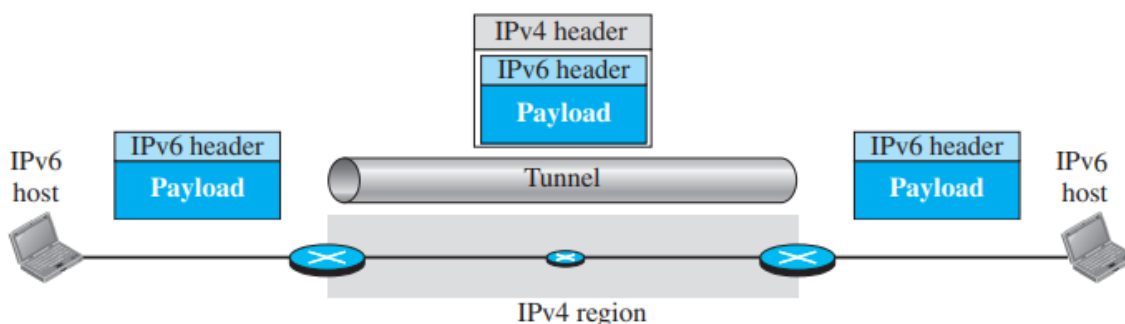
The transition from IPv4 to IPv6 cannot happen suddenly. It will take a considerable amount of time before every system in the Internet can move from IPv4 to IPv6. The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.

Three strategies have been devised for transition: dual stack, tunneling, and header translation.

Dual Stack It is recommended that all hosts, before migrating completely to version 6, have a dual stack of protocols during the transition. In other words, a station must run IPv4 and IPv6 simultaneously until all the Internet uses IPv6.

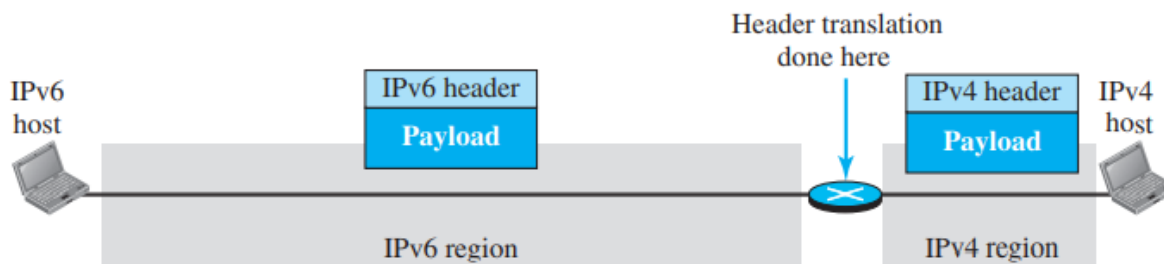


Tunneling Tunneling is a strategy used when two computers using IPv6 want to communicate with each other and the packet must pass through a region that uses IPv4. To pass through this region, the packet must have an IPv4 address. So the IPv6 packet is encapsulated in an IPv4 packet when it enters the region, and it leaves its capsule when it exits the region. It seems as if the IPv6 packet enters a tunnel at one end and emerges at the other end.



Header Translation Header translation is necessary when the majority of the Internet has moved to IPv6 but some systems still use IPv4. The sender wants to use IPv6, but the receiver does not understand IPv6. Tunneling does not work in this situation because the packet must be in the IPv4 format to be understood by the receiver. In this case, the header format must be totally changed

through header translation. The header of the IPv6 packet is converted to an IPv4 header.



IPv4	IPv6
IPv4 has a 32-bit address length	IPv6 has a 128-bit address length
It Supports Manual and DHCP address configuration	It supports Auto and renumbering address configuration
In IPv4 end to end, connection integrity is Unachievable	In IPv6 end to end, connection integrity is Achievable
It can generate 4.29×10^9 address space	Address space of IPv6 is quite large it can produce 3.4×10^{38} address space
The Security feature is dependent on application	IPSEC is an inbuilt security feature in the IPv6 protocol
Address representation of IPv4 is in decimal	Address Representation of IPv6 is in hexadecimal
Fragmentation performed by Sender and forwarding routers	In IPv6 fragmentation performed only by the sender
In IPv4 Packet flow identification is not available	In IPv6 packet flow identification are Available and uses the flow label field in the header
In IPv4 checksum field is available	In IPv6 checksum field is not available
It has broadcast Message Transmission Scheme	In IPv6 multicast and anycast message transmission scheme is available
In IPv4 Encryption and Authentication facility not provided	In IPv6 Encryption and Authentication are provided
IPv4 has a header of 20-60 bytes.	IPv6 has header of 40 bytes fixed

IPv4	IPv6
IPv4 consist of 4 fields which are separated by dot (.)	IPv6 consist of 8 fields, which are separated by colon (:)
IPv4's IP addresses are divided into five different classes. Class A , Class B, Class C , Class D , Class E.	IPv6 does not have any classes of IP address.
IPv4 supports VLSM(Variable Length subnet mask).	IPv6 does not support VLSM.
Example of IPv4: 66.94.29.13	Example of IPv6: 2001:0000:3238:DFE1:0063:0000:0000:FEFB