The Internet has thrived by providing better quality of service to support these applications. To keep the network layer untouched, these provisions are mostly implemented in the upper layer.

Security

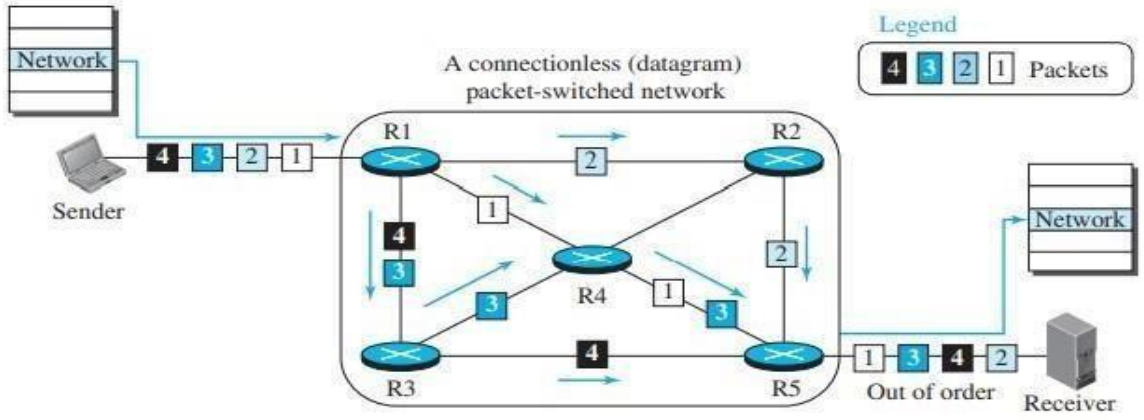
Security was not a concern when the Internet was originally designed because it was used by a small number of users at universities for research activities; other people had no access to the Internet.
The network layer was designed with no security provision. Today, security is a big concern.
To provide security for a connectionless network layer, we need to have another virtual level that changes the connectionless service to a connection-oriented service.

Packet Switching

From the routing and forwarding, we infer that a kind of switching occurs at the network layer. A router, in fact, is a switch that creates a connection between an input port and an output port.
In data communication switching techniques are divided into two broad categories, circuit switching and packet switching, only packet switching is used at the network layer because the unit of data at this layer is a packet.
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 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

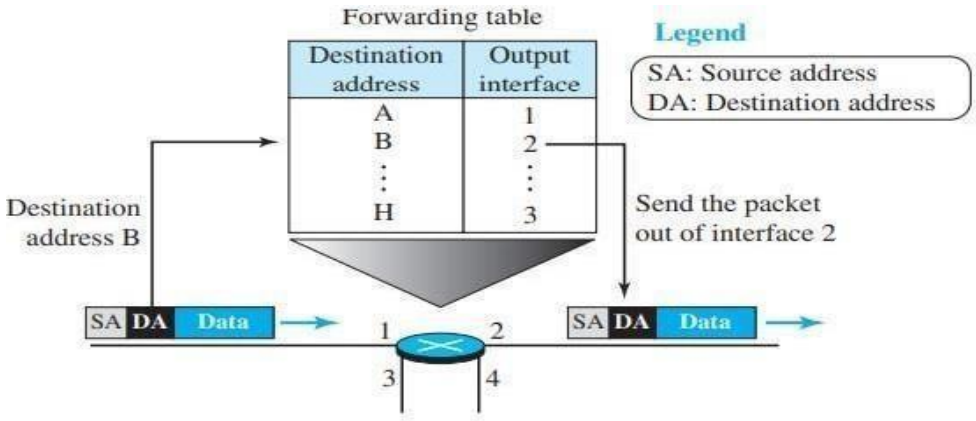
When the Internet started, the network layer was designed to provide a connectionless service in which the network-layer protocol treats each packet independently, with each packet having no relationship to any other packet.
The idea was that the network layer is only responsible for delivery of packets from the source to the destination. In this approach, the packets in a message may or may not travel the same path to their destination.



A connectionless packet-switched network

Each packet is routed based on the information contained in its header: source and destination addresses. The destination address defines where it should go; the source address defines where it comes from.

The router routes the packet based only on the destination address. The source address may be used to send an error message to the source if the packet is discarded. The figure shows the forwarding process in a router in this case.



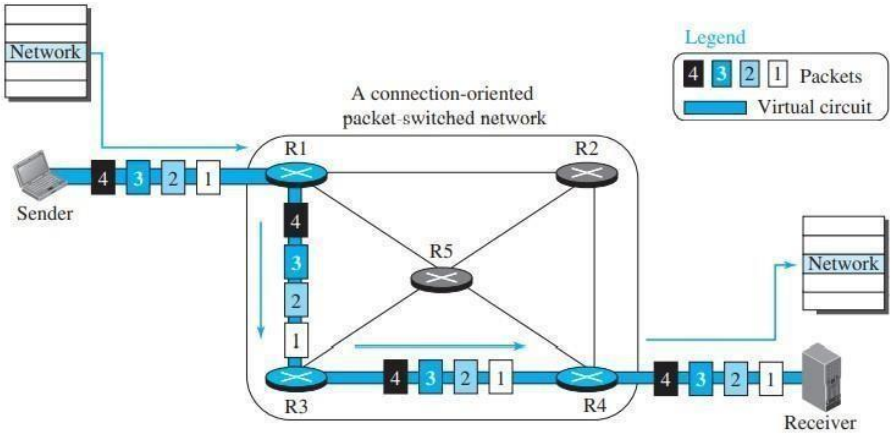
Forwarding process in a router when used in a connectionless network

Virtual-CircuitApproach: Connection-Oriented Service

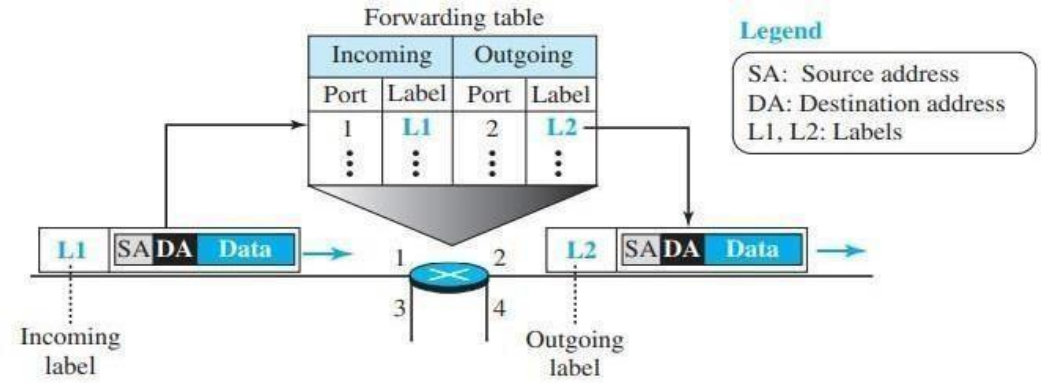
In a connection-oriented service (also called virtual-circuit approach), there is a relationship between all packets belonging to a message. Before all datagrams in a message can be sent, a virtual connection should be set up to define the path for the datagrams. After connection setup, the datagrams 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. Although it looks as though the use of the label may make the source and destination addresses unnecessary during the data transfer phase, parts of the Internet at the network layer keep these addresses.

One reason is that part of the packet path may still be using the connectionless service. Another reason is that the protocol at the network layer is designed with these addresses, and it may take a while before they can be changed. Figure shows the concept of connection-oriented service.



Each packet is forwarded based on the label in the packet. To follow the idea of connection-oriented design to be used in the Internet, we assume that the packet has a label when it reaches the router.



Forwarding process in a router when used in a virtual-circuit network

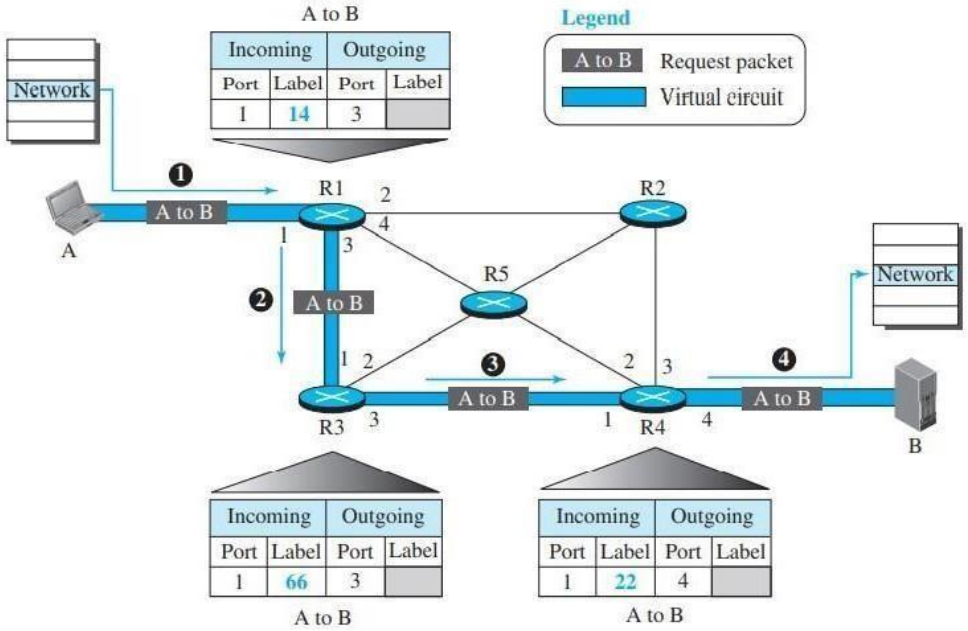
In this case, the forwarding decision is based on the value of the label, or virtual circuit identifier, as it is sometimes called.

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 address 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

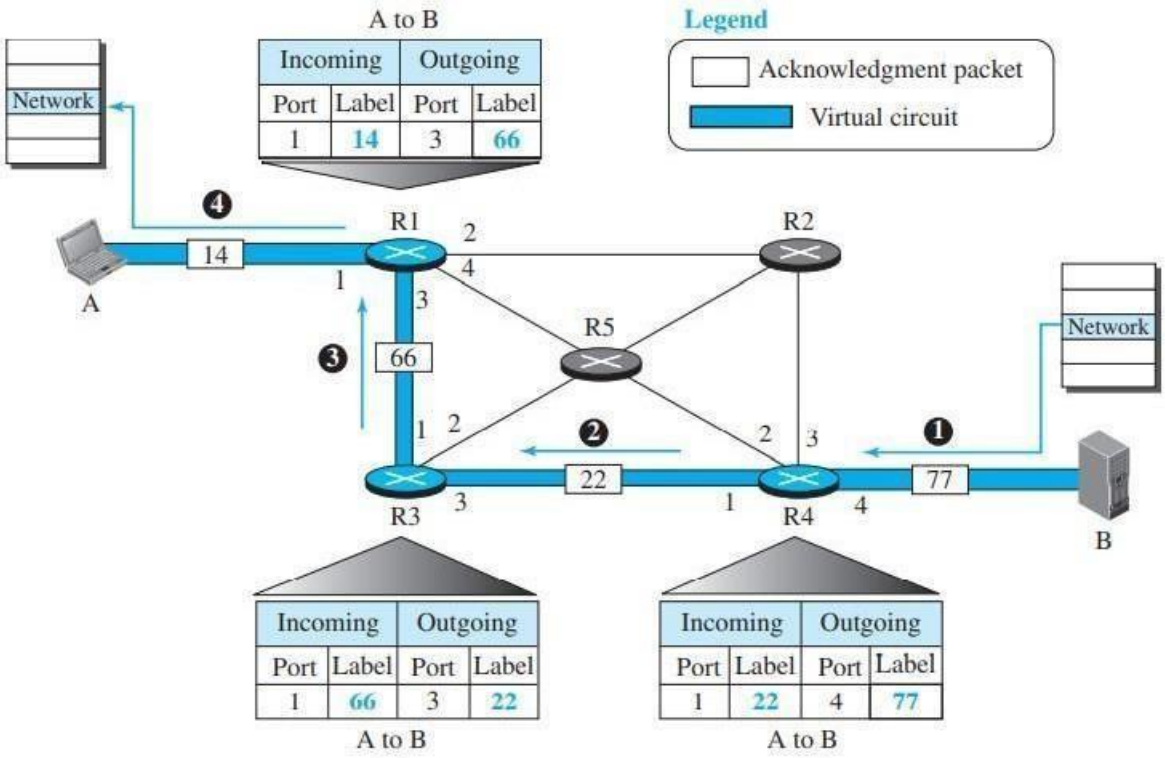
In the setup phase, a router creates an entry for a virtual circuit. For example, suppose source A needs to create a virtual circuit to destination B. Two auxiliary packets need to be exchanged between the sender and the receiver: the request packet and the acknowledgment packet.

Request packet: A request packet is sent from the source to the destination. This auxiliary packet carries the source and destination addresses. The figure shows this process.



- Source A sends a request packet to router R1.

- Router R1 receives the request packet. It knows that a packet going from A to B goes out through port 3. How the router has obtained this information is a point covered later. For the moment, assume that it knows the output port. 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.
- 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).
- Router R4 receives the setup request packet. Again, three columns are completed: incoming port (1), incoming label (22), and outgoing port (4).
- 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 Figure. 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.



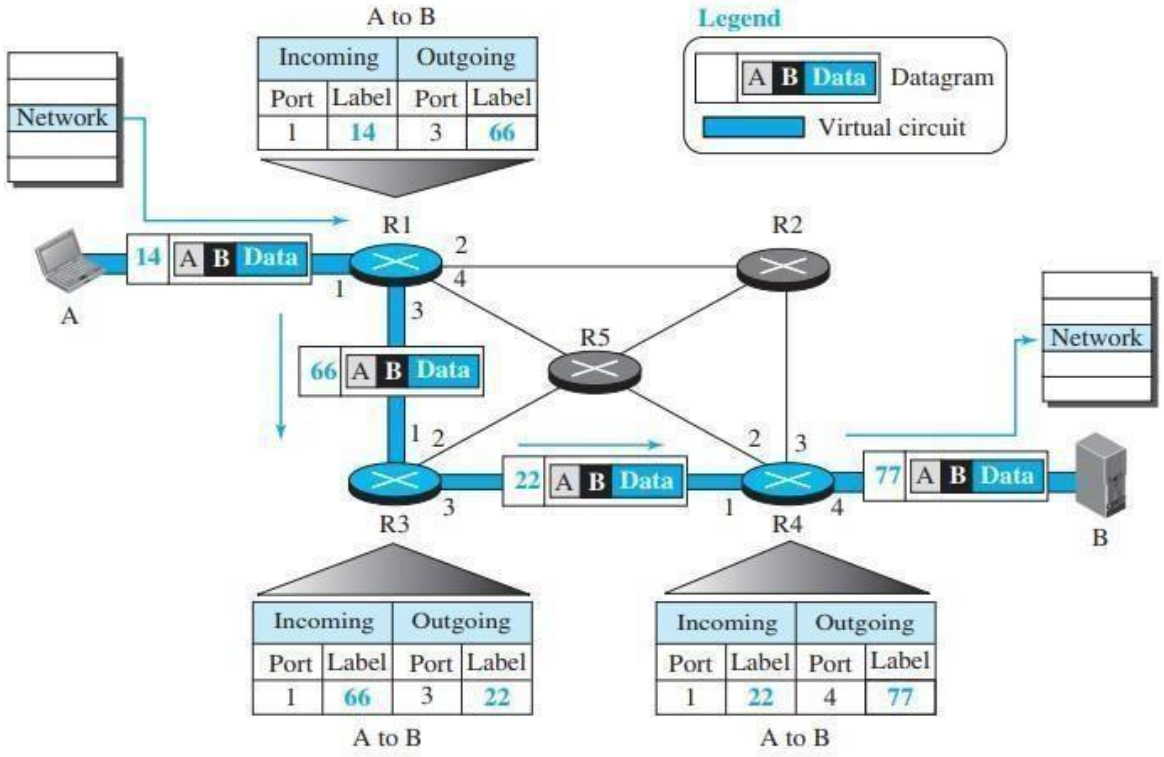
Sending acknowledgments in a virtual-circuit network

- 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.
- 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.

- 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.
- Finally, router R1 sends an acknowledgment to source A that contains its incoming label in the table, chosen in the setup phase.
- 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. The figure shows the flow of a single packet, but the process is the same for 1, 2, or 100 packets. The source computer uses the label 14, which it has received from router R1 in the setup phase. Router R1 forwards the packet to router R3 but changes the label to 66. Router R3 forwards the packet to router R4 but changes the label to 22. Finally, router R4 delivers the packet to its destination with the label 77. All the packets in the message follow the same sequence of labels, and the packets arrive in order at the destination.



Flow of one packet in an established virtual circuit

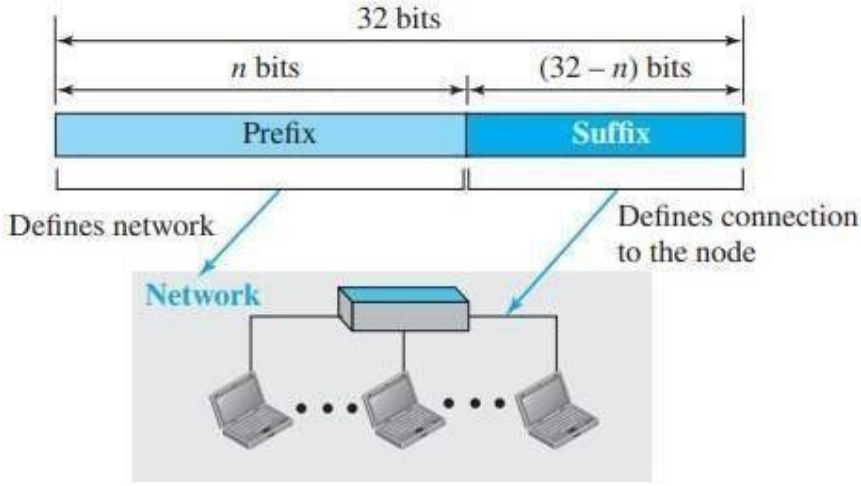
Teardown Phase

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

The identifier used in the IP layer of the TCP/IP protocol suite to identify the connection of each device to the Internet is called the Internet address or IP address.

- In a postal network, the postal address (mailing address) includes the country, state, city, street, house number, and the name of the mail recipient.
- Similarly, a telephone number is divided into the country code, area code, local exchange, and the connection.
- A 32-bit IPv4 address is also hierarchical but divided only 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 (connection of a device to the Internet).
- Figure shows the prefix and suffix of a 32-bit IPv4 address. The prefix length is n bits, and the suffix length is (32 − n) bits.



Hierarchy in addressing

- A prefix can be fixed length or variable length. The network identifier in the IPv4 was first designed as a fixed-length prefix. This scheme, which is now obsolete, is referred to as classful addressing.
- The new scheme, which is referred to as classless addressing, uses a variable-length network prefix. First, we briefly discuss classful addressing; then we concentrate on classless addressing.

Classful Addressing

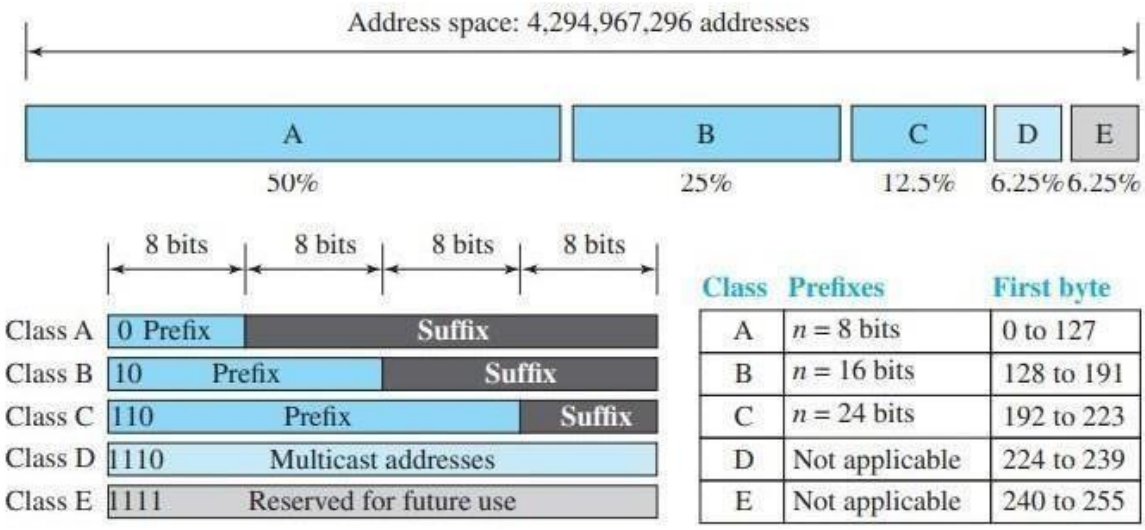
IPv4 address was designed with a fixed-length prefix, but to accommodate both small and large networks, three fixed-length prefixes were designed instead of one (n = 8, n = 16, and n = 24).

The whole address space was divided into five classes (class A, B, C, D, and E), as shown in Figure. This scheme is referred to as classful addressing.

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	Prefixes	First byte
A	$n = 8$ bits	0 to 127
B	$n = 16$ bits	128 to 191
C	$n = 24$ bits	192 to 223
D	Not applicable	224 to 239
E	Not applicable	240 to 255

Occupation of the address space in classful addressing

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.

1. 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.
- To understand the problem, let us think about class A. This class can be assigned to only 128 organizations in the world, but each organization needs to have a single network (seen by the rest of the world) with 16,777,216 nodes (computers in this single network).
- Since there may be only a few organizations that are this large, most of the addresses in this class were wasted (unused).
- Class B addresses were designed for midsize organizations, but many of the addresses in this class also remained unused.
- Class C addresses have a completely different flaw in design. The number of addresses that can be used in each network (256) was so small that most companies were not comfortable using a block in this address class.
- Class E addresses were almost never used, wasting the whole class.

2. 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. Each subnet has a larger prefix length than the original network.

- For example, if a network in class A is divided into four subnets, each subnet has a prefix of $n_{sub} = 10$. At the same time, if all the addresses in a network are not used, subnetting allows the addresses to be divided among several organizations.
- This idea did not work because most large organizations were not happy about dividing the block and giving some of the unused addresses to smaller organizations.
- While subnetting was devised to divide a large block into smaller ones, 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. This idea did not work either because it makes the routing of packets more difficult.

3. Advantage of Classful Addressing:

- 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.
- The prefix length in classful addressing is inherent in the address; no extra information is needed to extract the prefix and the suffix.

Classless Addressing

Subnetting and supernetting in classful addressing did not really solve the address depletion problem. With the growth of the Internet, it was clear that a larger address space was needed as a long-term solution.

The larger address space requires that the length of IP addresses also be increased, which means the format of the IP packets needs to be changed.

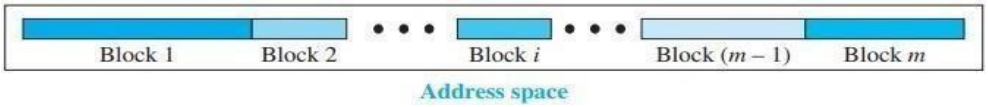
Although the long-range solution has already been devised and is called IPv6, a short-term solution was also devised to use the same address space but to change the distribution of addresses to provide a fair share to each organization.

The short-term solution still uses IPv4 addresses, but it is called classless addressing. The class privilege was removed from the distribution to compensate for the address depletion.

In classless addressing, variable-length blocks are used that belong to no classes. We can have a block of 1 address, 2 addresses, 4 addresses, 128 addresses, and so on.

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).

Figure shows the division of the whole address space into nonoverlapping blocks.



Variable-length blocks in classless addressing

Unlike classful addressing, 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.

We need to emphasize that the idea of classless addressing can be easily applied to classful addressing. An address in class A can be thought of as a classless address in which the prefix length is 8.

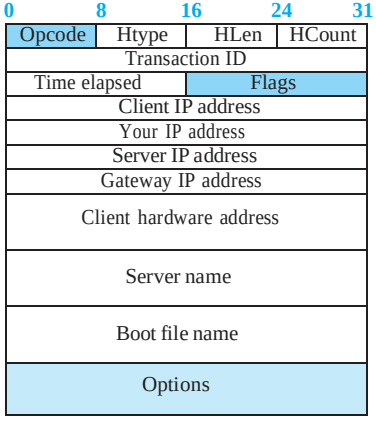
Dynamic Host Configuration Protocol (DHCP)

- We have seen that a large organization or an ISP can receive a block of addresses directly from ICANN and a small organization can receive a block of addresses from an ISP.
- After a block of addresses are assigned to an organization, the network administration can manually assign addresses to the individual hosts or routers. However, address assignment in an organization can be done automatically using the **Dynamic Host Configuration Protocol (DHCP).**
- DHCP is an application-layer program, using the client-server paradigm, that actually helps TCP/IP at the network layer.
- DHCP has found such widespread use in the Internet that it is often called a *plug- and-play protocol*.
- A network manager can configure DHCP to assign permanent IP addresses to the host and routers.
- DHCP can also be configured to provide temporary, on demand, IP addresses to hosts. The second capability can provide a temporary IP address to a traveller to connect her laptop to the Internet while she is staying in the hotel.
- It also allows an ISP with 1000 granted addresses to provide services to 4000 households, assuming not more than one-forth of customers use the Internet at the same time.

DHCP Message Format

- DHCP is a client-server protocol in which the client sends a request message and the server returns a response message. Before we discuss the operation of DHCP, let us show the general format of the DHCP message in Figure 18.25.
- Most of the fields are explained in the figure, but we need to discuss the option field, which plays a very important role in DHCP.
- The 64-byte option field has a dual purpose. It can carry either additional information or some specific vendor information.
- The server uses a number, called a **magic cookie,** in the format of an IP address with the value of 99.130.83.99.
- When the client finishes reading the message, it looks for this magic cookie. If present, the next 60 bytes are options.

Figure 18.25 *DHCP message format*

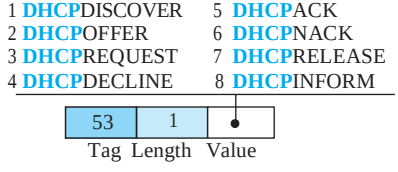


Fields:

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

- An option is composed of three fields: a 1-byte tag field, a 1-byte length field, and a variable-length value field.
- There are several tag fields that are mostly used by vendors. If the tag field is 53, the value field defines one of the 8 message types shown in Figure 18.26.

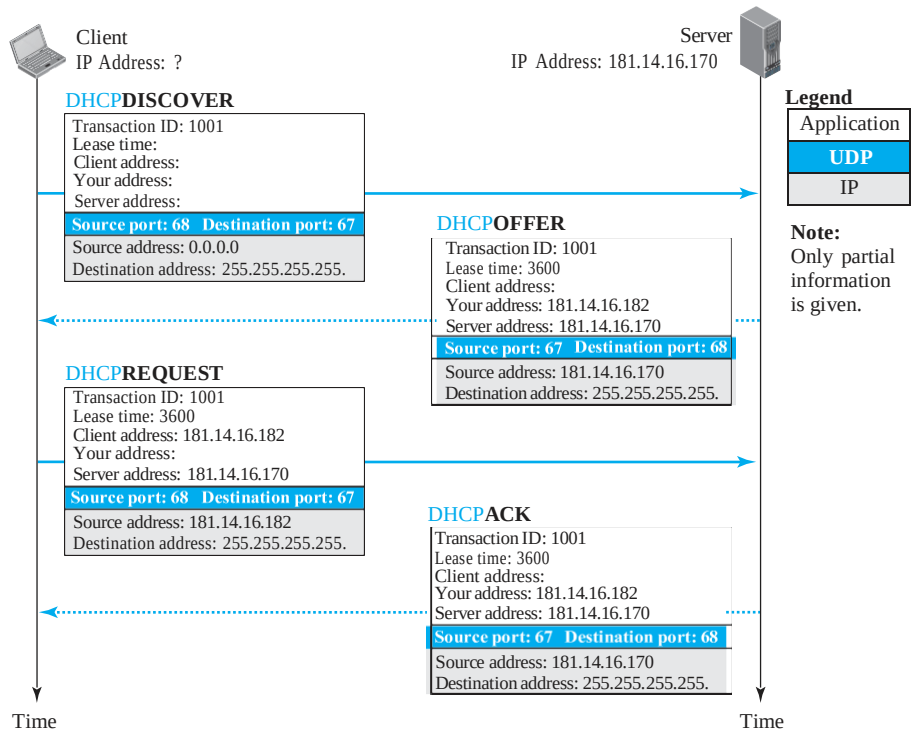
Figure 18.26 *Option format*



DHCP Operation

Figure 18.27 shows a simple scenario.

Figure 18.27 *Operation of DHCP*



1. The joining host creates a **DHCP**DISCOVER message in which only the transaction-ID field is set to a random number. No other field can be set because the host has no knowledge with which to do so. This message is encapsulated in a UDP user datagram with the source port set to 68 and the destination port set to 67. We will discuss the reason for using two well-known port numbers later. The user datagram is encapsulated in an IP datagram with the source address set to **0.0.0.0** ("this host") and the destination address set to **255.255.255.255** (broadcast address). The reason is that the joining host knows neither its own address nor the server address.
2. The DHCP server or servers (if more than one) responds with a **DHCP**OFFER message in which the your address field defines the offered IP address for the joining host and the server address field includes the IP address of the server. The message also includes the lease time for which the host can keep the IP address. This message is encapsulated in a user datagram with the same port numbers, but in the reverse order. The user datagram in turn is encapsulated in a datagram with the server address as the source IP address, but the destination address is a broadcast address, in which the server allows other DHCP servers to receive the offer and give a better offer if they can.

3. The joining host receives one or more offers and selects the best of them. The joining host then sends a **DHCP**REQUEST message to the server that has given the best offer. The fields with known value are set. The message is encapsulated in a user datagram with port numbers as the first message. The user datagram is encapsulated in an IP datagram with the source address set to the new client address, but the destination address still is set to the broadcast address to let the other servers know that their offer was not accepted.
4. Finally, the selected server responds with a **DHCP**ACK message to the client if the offered IP address is valid. If the server cannot keep its offer (for example, if the address is offered to another host in between), the server sends a **DHCP**NACK message and the client needs to repeat the process. This message is also broadcast to let other servers know that the request is accepted or rejected.

Two Well-Known Ports

- We said that the DHCP uses two well-known ports (68 and 67) instead of one well-known and one ephemeral.
- The reason for choosing the well-known port 68 instead of an ephemeral port for the client is that the response from the server to the client is broadcast.
- Remember that an IP datagram with the limited broadcast message is delivered to every host on the network. Now assume that a DHCP client and a DAYTIME client, for example, are both waiting to receive a response from their corresponding server and both have accidentally used the same temporary port number (56017, for example).
- Both hosts receive the response message from the DHCP server and deliver the message to their cli- ents. The DHCP client processes the message; the DAYTIME client is totally confused with a strange message received. Using a well-known port number prevents this problem from happening.
- The response message from the DHCP server is not delivered to the DAYTIME client, which is running on the port number 56017, not 68.
- The temporary port numbers are selected from a different range than the well-known port numbers.

Using FTP

The server does not send all of the information that a client may need for joining the network. In the **DHCP**ACK message, the server defines the pathname of a file in which the client can find complete information such as the address of the DNS server. The client can then use a file transfer protocol to obtain the rest of the needed information.

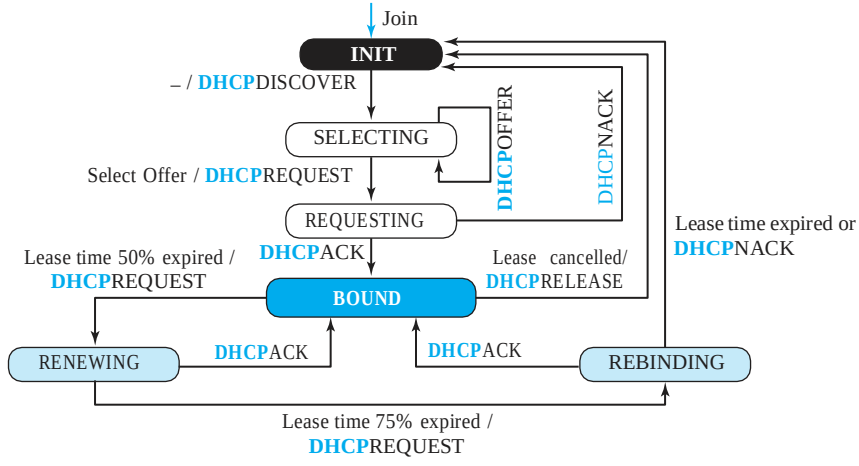
Error Control

DHCP uses the service of UDP, which is not reliable. To provide error control, DHCP uses two strategies. First, DHCP requires that UDP use the checksum. As we will see in Chapter 24, the use of the checksum in UDP is optional. Second, the DHCP client uses timers and a retransmission policy if it does not receive the DHCP reply to a request. However, to prevent a traffic jam when several hosts need to retransmit a request (for example, after a power failure), DHCP forces the client to use a random number to set its timers.

Transition States

The previous scenarios we discussed for the operation of the DHCP were very simple. To provide dynamic address allocation, the DHCP client acts as a state machine that performs transitions from one state to another depending on the messages it receives or sends. Figure 18.28 shows the transition diagram with the main states.

Figure 18.28 *FSM for the DHCP client*



- When the DHCP client first starts, it is in the INIT state (initializing state). The client broadcasts a discover message.
- When it receives an offer, the client goes to the SELECTING state. While it is there, it mayreceive more offers.
- After it selects an offer, it sends a request message and goes to the REQUESTING state. If an ACK arrives while the client is in this state, it goes to the BOUND state and uses the IP address. When the lease is 50 percent expired, the client tries to renew it by moving to the RENEWING state. If the server renews the lease, the client moves to the BOUND state again. If the lease is not renewed and the lease time is 75 percent expired, the client moves to the REBINDING state.
- If the server agrees with the lease (ACK message arrives), the client moves to the BOUND state and continues using the IP address; otherwise, the client moves to the INIT state and requests another IP address.Note that the client can use the IP address only when it is in the BOUND, RENEWING, or REBINDING state. The above procedure requires that the client uses three timers: *renewal timer* (set to 50 percent of the lease time), *rebind- ing timer* (set to 75 percent of the lease time), and *expiration timer* (set to the lease time).

UNICAST ROUTING PROTOCOLS

Hierarchical Routing in Internet

The Internet today is made of a huge number of networks and routers that connect them. Routing in the Internet cannot be done using a single protocol for two reasons:
A scalability problem and an administrative issue.

Scalability problem means that the size of the forwarding tables becomes huge, searching for a destination in a forwarding table becomes time-consuming, and updating creates a huge amount of traffic.

Hierarchical routing means considering each ISP as an autonomous system (AS).Each AS can run a routing protocol that meets its needs, but the global Internet runs a global protocol to join and connect all ASs together.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).

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

A router in this protocol implements the distance-vector routing algorithm. First, since a router in an AS needs to know how to forward a packet to different networks(subnets) in an AS, RIP routers advertise the cost of reaching different networks instead of reaching other nodes in a theoretical graph.

The cost is defined between a router and the network in which the destination host is located. Second, to make the implementation of the cost simpler (independent from performance factors of the routers and links, such as delay, bandwidth, and so on), the cost is defined as the number of hops, which means the number of networks (subnets)a packet needs to travel through from the source router to the final destination host.

Note that the network in which the source host is connected is not counted in this calculation because the source host does not use a forwarding table; the packet is delivered to the default router.

Figure 3.3.1 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.

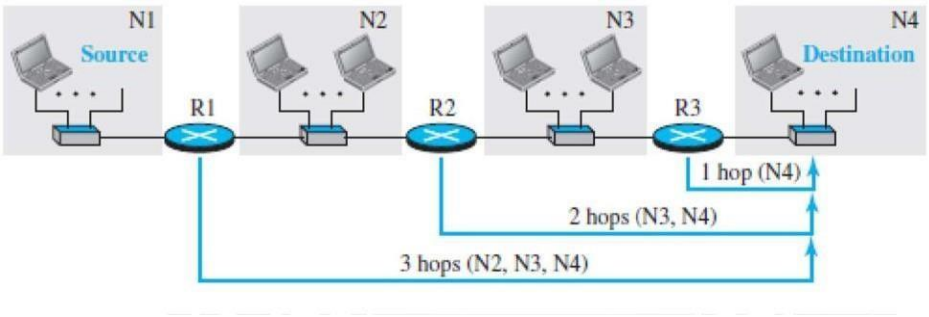


Fig3.3.1: Hop counts in RIP.
[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-613]

Forwarding Table

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.Figure3.3.2 shows the three forwarding tables for the routers in Figure (above).

Note that the first and the third columns together convey the same information as does a distance vector, but the cost shows the number of hops to the destination networks.

Forwarding table for R1

Destination network	Next router	Cost in hops
N1	——	1
N2	——	1
N3	R2	2
N4	R2	3

Forwarding table for R2

Destination network	Next router	Cost in hops
N1	R1	2
N2	——	1
N3	——	1
N4	R3	2

Forwarding table for R3

Destination network	Next router	Cost in hops
N1	R2	3
N2	R2	2
N3	——	1
N4	——	1

Fig3.3.2:Forwarding tables in RIP.
[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-614]

For example**,** R1 defines that the next router for the path to N4 is R2; R2 defines that the next router to N4 isR3; R3 defines that there is no next router for this path. The tree is then R1 -R2-R3-N4.

The third column is not needed for forwarding the packet, but it is needed for updating the forwarding table when there is a change in the route.

RIP Implementation

RIP is implemented as a process that uses the service of UDP on the port number 520. RIP is a routing protocol to help IP route its datagrams through the AS,the RIP messages are encapsulated inside UDP user datagrams, which in turn are encapsulated inside IP datagrams.

That is, RIP runs at the application layer, but creates forwarding tables for IP at the network layer.

RIP Messages

Two RIP processes, a client and a server, need to exchange messages. RIP-2 defines the format of the message, as shown in figure3.3.3.The message Entry, can be repeated as needed in a message. Each entry carries the information related to one line in the forwarding table of the router that sends the message.

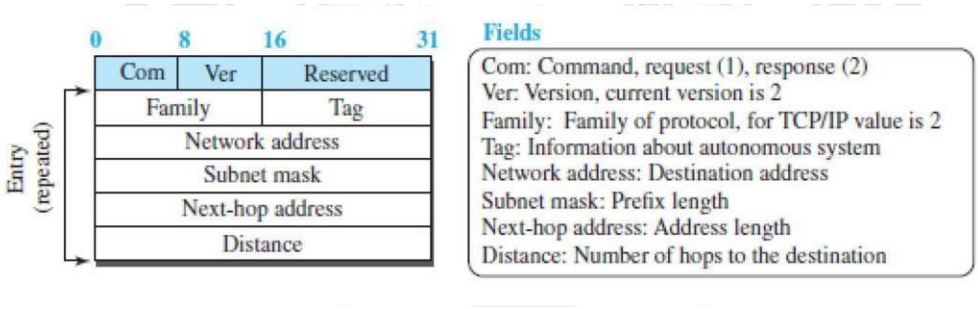


Fig3.3.3: RIP message format.

[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-615]

RIP has two types of messages:

Request and response. A request message is sent by a router that has just come up or by a router that has some time-out entries.

A request message can ask about specific entries or all entries.

A response (or update)message can be either solicited or unsolicited. A solicited response message is sent only in answer to a request message. It contains information about the destination specified in the corresponding request message.

RIP Algorithm

RIP implements the same algorithm as the distance-vector routing 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.
- 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.

Timers in RIP

RIP uses three timers to support its operation.

The periodic timer controls the advertising of regular update messages. Each router has one periodic timer that is randomly set to a number between 25 and 35 seconds (to prevent all routers sending their messages at the same time and creating excess traffic). The timer counts down; when zero is reached, the update message is sent, and the timer is randomly set once again.

The expiration timer governs the validity of a route. 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. Every route has its own expiration timer. The garbage collection timer is used to purge a route from the forwarding table.

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.

Border Gateway Protocol Version 4 (BGP4)

The Border Gateway Protocol version 4 (BGP4) is the only inter domain routing protocol used in the Internet today.

Consider an example of an internet with four autonomous systems. AS2, AS3, and AS4 are stub autonomous systems; AS1 is a transient one as shown in figure 3.3.8 .

. Here,data exchange between AS2, AS3, and AS4 should pass through AS1.

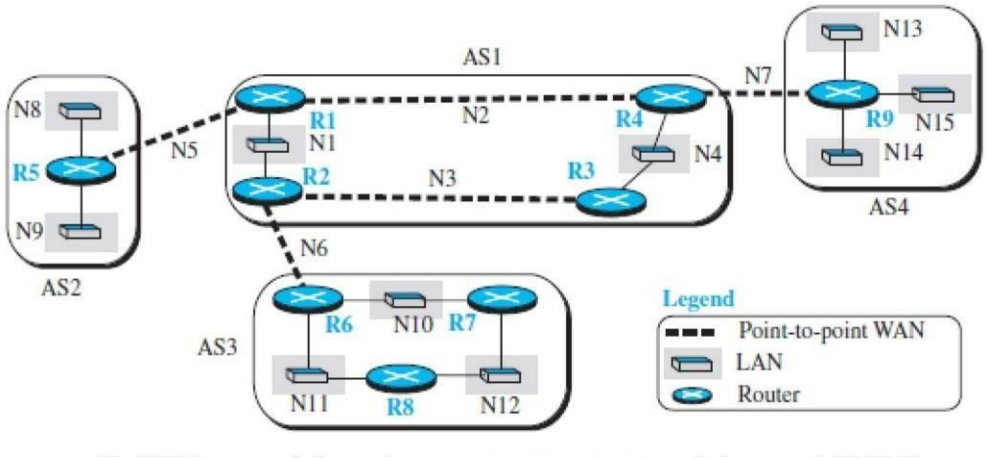


Fig3.3.8:Sample internet with four AS.

[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-623]

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 (the one at the edge of each AS which is connected to a router at another AS).We then install the second variation of BGP, called internal BGP (iBGP), on all routers.

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)

BGP is a point-to-point protocol. When the software is installed on two routers, they try to create a TCP connection using the well-known port 179.

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 we use has three pairs: R1-R5, R2-R6, and R4-R9. The connection between these pairs is established over three physical WANs (N5,N6, and N7). There is a need for a logical TCP connection to be created over the physical connection to make the exchange of information possible. Each logical connection in BGP is referred to as a session. This means that we need three sessions, as shown in Figure 3.3.9.

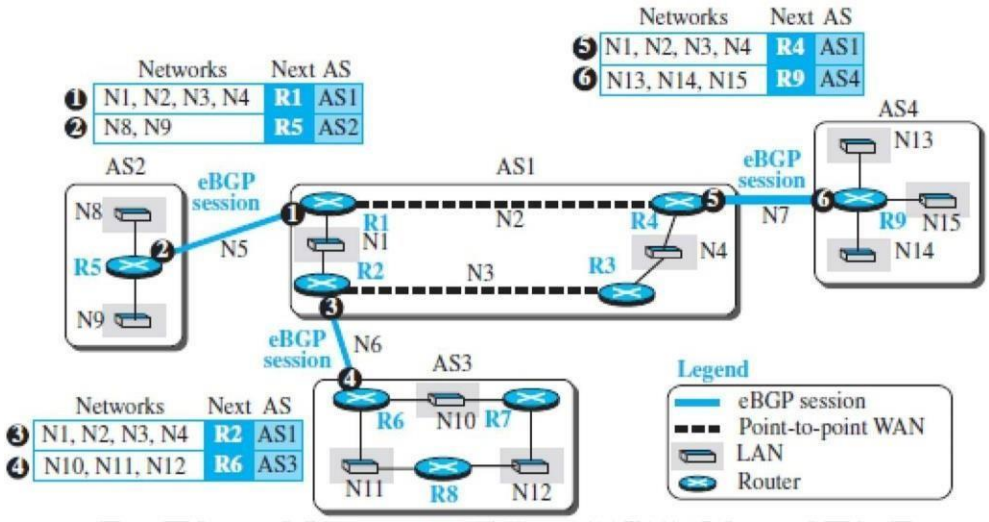


Fig3.3.9: EBGP operation.

[Source : "Data Communications and Networking" by Behrouz A. Forouzan,Page-624]

The circled number defines the sending router in each case.

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 (R1 gets this information from the corresponding intradomain forwarding table).

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.

Messages

BGP four types of messages for communication between the BGP speakers across the ASs and inside an AS:

Four messages are open, update, keep alive, 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 used by a router to withdraw destinations that have been advertised previously, to announce a route to a new destination, or both.

Note that BGP can withdraw several destinations that were advertised before, but it can only advertise one new destination in a single update message.

Keep alive Message. The BGP peers that are running exchange keep alive 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.

Performance

BGP performance can be compared with RIP. BGP speakers exchange a lot of messages to create forwarding tables, but BGP is free from loops and count-to-infinity.