# Unit 4: Network Layer

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links) i.e., it ensures that each packet gets from its point of origin to its final destination. The network layer is considered the backbone of the OSI Model. It selects and manages the best logical path for data transfer between nodes. The routing information contained within a packet includes the source of the sending host and the eventual destination of the remote host. This information is contained within the network layer header that encapsulates network frames at the data link layer. The primary function of the network layer is to permit different networks to be interconnected. It does this by forwarding packets to network routers, which rely on algorithms to determine the best paths for the data to travel. The network layer can support either connection-oriented or connectionless networks, but such a network can only be of one type and not both.

**Internet Protocol:**

The Internet Protocol (IP) is the method or protocol by which data is sent from one computer to another on the Internet. Each computer (known as a host) on the Internet has at least one IP address that uniquely identifies it from all other computers on the Internet. When you send or receive data (for example, an e-mail note or a Web page), the message gets divided into little chunks called packets. Each of these packets contains both the sender's Internet address and the receiver's address. Any packet is sent first to a gateway computer that understands a small part of the Internet. The gateway computer reads the destination address and forwards the packet to an adjacent gateway that in turn reads the destination address and so forth across the Internet until one gateway recognizes the packet as belonging to a computer within its immediate neighborhood or domain. That gateway then forwards the packet directly to the computer whose address is specified.

**IP Address:**

An Internet Protocol address (IP address) is a logical numeric address that is assigned to every single computer, printer, switch, router or any other device that is part of a TCP/IP-based network. The IP address is the core component on which the networking architecture is built; no network exists without it. An IP address is a logical address that is used to uniquely identify every node in the network. Because IP addresses are logical, they can change. They are similar to addresses in a town or city because the IP address gives the network node an address so that it can communicate with other nodes or networks.

The numerals in an IP address are divided into 2 parts:

- The network part specifies which networks this address belongs to and
- The host part further pinpoints the exact location.

**IPv4:**

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device to the internet. If a device operating in the network layer has m connections, then it needs to have m addresses. Router is such type of device.

An IPV4 address consists of 4 bytes in the form a.b.c.d (E.g. 173.14.2.225, 11.12.13.3). It can be logically divided into a network and a host portion. While the network portion identifies the network to which the end node belongs to, the host portion uniquely identifies the end node, from the other end nodes, inside the network.

| **Binary Format** | **Dotted Decimal Notation** |
|---|---|
| 11000000 10101000 00000011 00011000 | 192.168.3.24 |

IPv4 uses 32-bit addresses, which means that the address space is $2^{32}$ (more than 4 billion). This means that, theoretically, if there were no restrictions, more than 4 billion devices could be connected to the internet.

**IPv4 ADDRESSING SCHEME**

IP addresses falls into two types:

- Classful IP addressing is a legacy scheme which divides the whole IP address pools into 5 distinct classes—A, B, C, D and E.
- Classless IP addressing has an arbitrary length of the prefixes.

Classful Addressing

Class A

The first octet denotes the network address, and the last three octets are the host portion. Any IP address whose first octet is between 1 and 126 is a Class A address. Note that 0 is reserved as a part of the default address and 127 is reserved for internal loopback testing.

Format: network.host.host.host

Default subnet mask = 255.0.0.0 or (slash notation) /8

Class B

The first two octets denote the network address, and the last two octets are the host portion. Any address whose first octet is in the range 128 to 191 is a Class B address.

Format: network.network.host.host

Default subnet mask =255.255.0.0 or /16

Class C

The first three octets denote the network address, and the last octet is the host portion. The first octet range of 192 to 223 is a Class C address.

Format: network.network.network.host

Default subnet mask = 255.255.255.0 or /24

Class D

Used for multicast. Multicast IP addresses have their first octets in the range 224 to 239.

Class E

Reserved for future use or research purpose and includes the range of addresses with a first octet from 240 to 255.

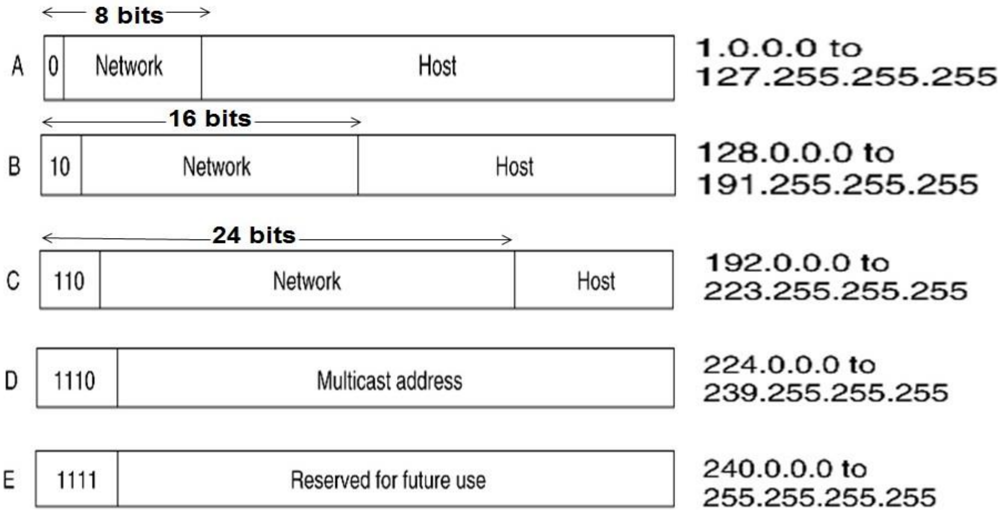
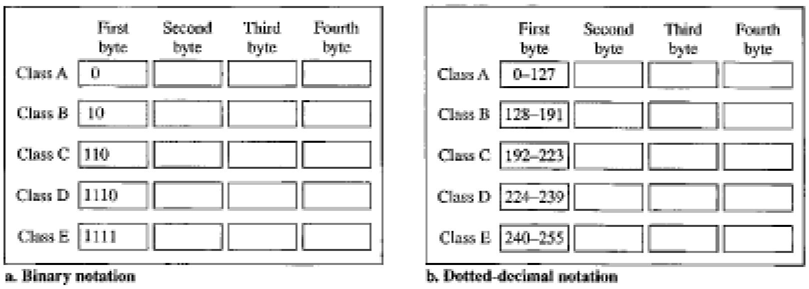


**Figure 19.2** *Finding the classes in binary and dotted-decimal notation*



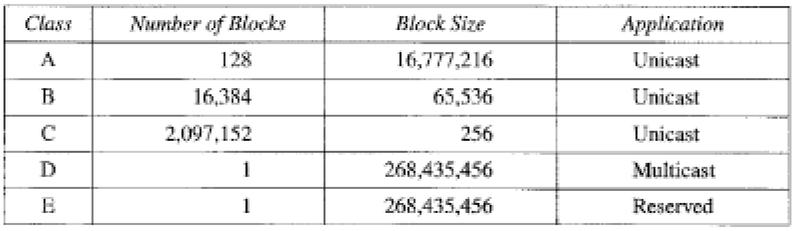
| | First byte | Second byte | Third byte | Fourth byte |
|---|---|---|---|---|
| Class A | 0 | | | |
| Class B | 10 | | | |
| Class C | 110 | | | |
| Class D | 1110 | | | |
| Class E | 1111 | | | |

a. Binary notation

| | First byte | Second byte | Third byte | Fourth byte |
|---|---|---|---|---|
| Class A | 0–127 | | | |
| Class B | 128–191 | | | |
| Class C | 192–223 | | | |
| Class D | 224–239 | | | |
| Class E | 240–255 | | | |

b. Dotted-decimal notation

**Table 19.1** *Number of blocks and block size in classful IPv4 addressing*



| *Class* | *Number of Blocks* | *Block Size* | *Application* |
|---|---|---|---|
| A | 128 | 16,777,216 | Unicast |
| B | 16,384 | 65,536 | Unicast |
| C | 2,097,152 | 256 | Unicast |
| D | 1 | 268,435,456 | Multicast |
| E | 1 | 268,435,456 | Reserved |

### *Netid and Hostid*

In classful addressing, an IP address in class A, B, or C is divided into **netid** and **hostid.** These parts are of varying lengths, depending on the class of the address. Figure 19.2 shows some netid and hostid bytes. The netid is in color, the hostid is in white. Note that the concept does not apply to classes D and E.

In class A, one byte defines the netid and three bytes define the hostid. In class B, two bytes define the netid and two bytes define the hostid. In class C, three bytes define the netid and one byte defines the hostid.

### *Mask*

Although the length of the netid and hostid (in bits) is predetermined in classful addressing, we can also use a **mask** (also called the **default mask**), a 32-bit number made of

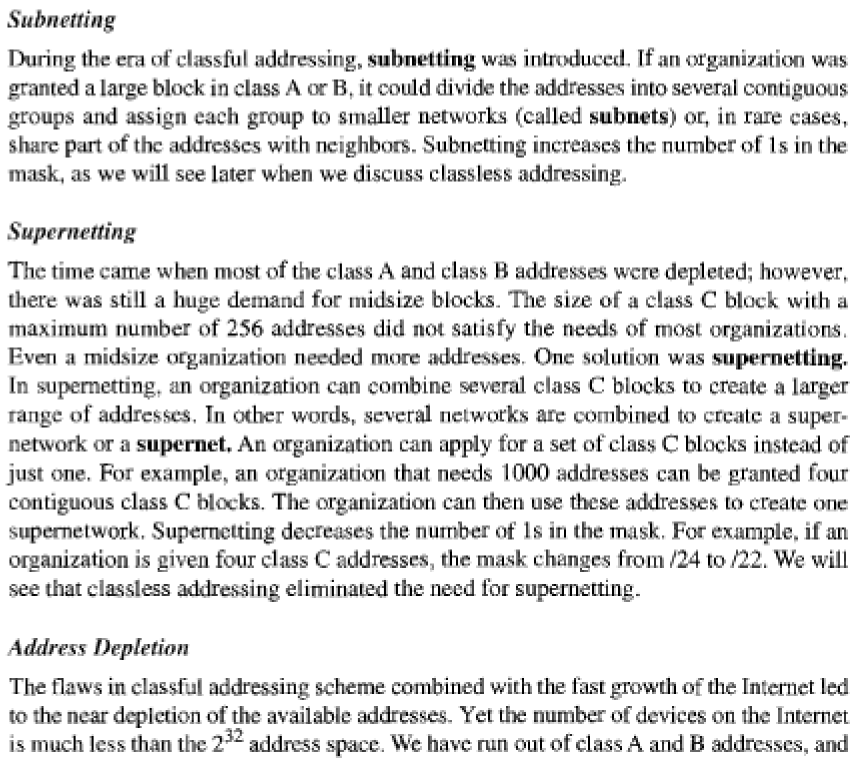
contiguous 1s followed by contiguous 0s. The masks for classes A, B, and C are shown in Table 19.2. The concept does not apply to classes D and E.

**Table 19.2** *Default masks for classful addressing*

| *Class* | *Binary* | *Dotted-Decimal* | *CIDR* |
|---|---|---|---|
| A | **11111111** 00000000 00000000 00000000 | **255**.0.0.0 | /8 |
| B | **11111111 11111111** 00000000 00000000 | **255.255**.0.0 | /16 |
| C | **11111111 11111111 11111111** 00000000 | **255.255.255**.0 | /24 |

The mask can help us to find the netid and the hostid. For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the netid; the next 24 bits define the hostid.

The last column of Table 19.2 shows the mask in the form /*n* where *n* can be 8, 16, or 24 in classful addressing. This notation is also called slash notation or **Classless Interdomain Routing (CIDR)** notation. The notation is used in classless addressing, which we will discuss later. We introduce it here because it can also be applied to classful addressing. We will show later that classful addressing is a special case of classless addressing.



### *Subnetting*

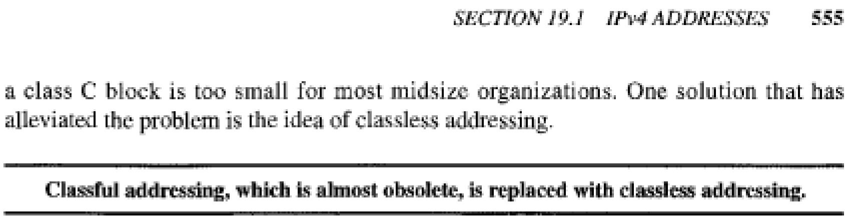
During the era of classful addressing, **subnetting** was introduced. If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks (called **subnets**) or, in rare cases, share part of the addresses with neighbors. Subnetting increases the number of 1s in the mask, as we will see later when we discuss classless addressing.

### *Supernetting*

The time came when most of the class A and class B addresses were depleted; however, there was still a huge demand for midsize blocks. The size of a class C block with a maximum number of 256 addresses did not satisfy the needs of most organizations. Even a midsize organization needed more addresses. One solution was **supernetting.** In supernetting, an organization can combine several class C blocks to create a larger range of addresses. In other words, several networks are combined to create a supernetwork or a **supernet.** An organization can apply for a set of class C blocks instead of just one. For example, an organization that needs 1000 addresses can be granted four contiguous class C blocks. The organization can then use these addresses to create one supernetwork. Supernetting decreases the number of 1s in the mask. For example, if an organization is given four class C addresses, the mask changes from /24 to /22. We will see that classless addressing eliminated the need for supernetting.

### *Address Depletion*

The flaws in classful addressing scheme combined with the fast growth of the Internet led to the near depletion of the available addresses. Yet the number of devices on the Internet is much less than the $2^{32}$ address space. We have run out of class A and B addresses, and



a class C block is too small for most midsize organizations. One solution that has alleviated the problem is the idea of classless addressing.

**Classful addressing, which is almost obsolete, is replaced with classless addressing.**

Classless IP addresses

Classful IP addresses is no longer popular and instead has been replaced with the concept of classless IP address, where there is no concept of IP address classes and no strict network and host boundaries. In classless IP addressing, there is no concept of Classful addressing like Classes A, B, C, D and E. IPv4 address range 0.0.0.0 to 223.255.255.255 treated as a single class. No strict 8-byte boundaries for the network and host portions. A Subnet masks defines network & host boundaries. This approach is very useful for optimizing address usage.

Examples of Classless Addressing

Network address – 22.10.0.0 /16

Network address – 173.2.224.0 / 21

In the above address, 16 and 21 denote the subnet masks respectively. This means that in the first address 22.10.0.0, the first 16 bits are reserved for the network portion and the rest of the 16 bits are reserved for the host portion. Similarly, in the second address 173.2.224.0/21, the first 21 bits are reserved for the network portion and the remaining 13 bits are reserved for the host portion. Thus, it can be seen that classless addressing gives a flexible boundary between the network and host portions, thereby allowing lot of flexibility in partitioning the networks.

**Subnetting:**

Subnetting is the practice of dividing a network into two or more smaller networks. The major advantage of subnetting is to reduce the address wastage. It increases routing efficiency, enhances the security of the network and reduces the size of the broadcast domain.

The reasons to use subnetting are:

- Conservation of IP addresses
- Reduced network traffic
- Simplified troubleshooting

**FLSM vs VLSM:**

FLSM stands for Full Length Subnet Mask. It means all the subnets are of the same size. In FLSM, the subnet mask remains the same for all the subnets.

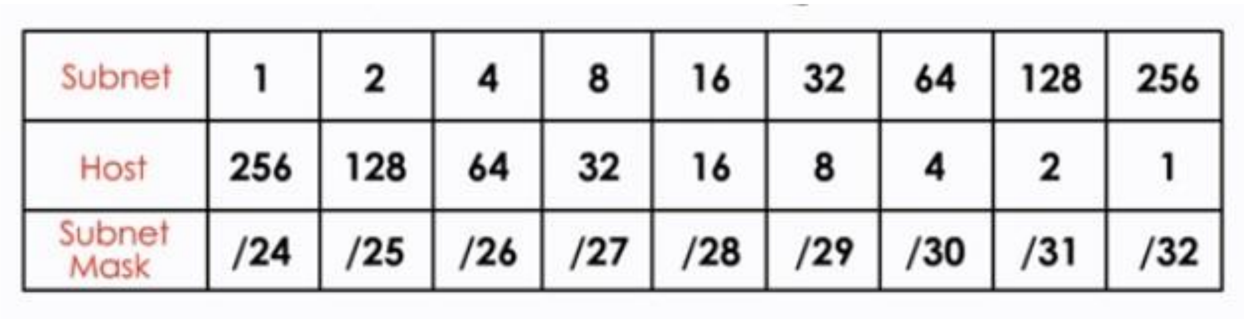
VLSM stands for Variable Length Subnet Mask. It means the size of the subnet varies according to the needs. In VLSM, the subnet mask is different normally but it can be same for any two or more subnets depending upon the situation.

Internet Service Providers may face a situation where they need to allocate IP subnets of different sizes as per the requirement of customer. One customer may ask Class C subnet of 3 IP addresses and another may ask for 100 IPs. For an ISP, it is not feasible to divide the IP addresses into fixed size subnets, rather he may want to subnet the subnets in such a way which results in minimum wastage of IP addresses.

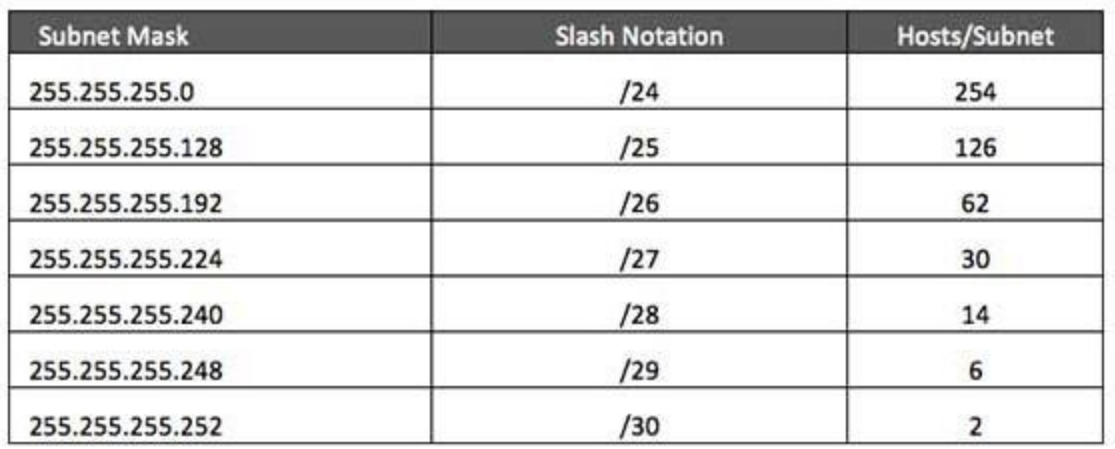
For example, an administrator has 192.168.1.0/24 network. The suffix /24 (pronounced as "slash 24") tells the number of bits used for network address. In this example, the administrator has three different departments with different number of hosts. Sales department has 100 computers, Purchase department has 50 computers, Accounts has 25 computers and Management has 5 computers. In CIDR, the subnets are of fixed size. Using the same methodology, the administrator cannot fulfill all the requirements of the network.

The following procedure shows how VLSM can be used in order to allocate department-wise IP addresses as mentioned in the example.

**Step 1:** Make a list of Subnets possible.



| Subnet | 1 | 2 | 4 | 8 | 16 | 32 | 64 | 128 | 256 |
|---|---|---|---|---|---|---|---|---|---|
| Host | 256 | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
| Subnet Mask | /24 | /25 | /26 | /27 | /28 | /29 | /30 | /31 | /32 |



| Subnet Mask | Slash Notation | Hosts/Subnet |
|---|---|---|
| 255.255.255.0 | /24 | 254 |
| 255.255.255.128 | /25 | 126 |
| 255.255.255.192 | /26 | 62 |
| 255.255.255.224 | /27 | 30 |
| 255.255.255.240 | /28 | 14 |
| 255.255.255.248 | /29 | 6 |
| 255.255.255.252 | /30 | 2 |

**Step 2:** Sort the requirements of IPs in descending order (Highest to Lowest).

Sales 100

Purchase 50

Accounts 25

Management 5

**Step 3**: Allocate the highest range of IPs to the highest requirement, so let's assign 192.168.1.0 /25 (255.255.255.128) to the Sales department. This IP subnet with Network number 192.168.1.0 has 126 valid Host IP addresses which satisfy the requirement of the Sales department. The subnet mask used for this subnet has 10000000 as the last octet.

**Step 4:** Allocate the next highest range, so let's assign 192.168.1.128 /26 (255.255.255.192) to the Purchase department. This IP subnet with Network number 192.168.1.128 has 62 valid Host IP Addresses which can be easily assigned to all the PCs of the Purchase department. The subnet mask used has 11000000 in the last octet.

**Step 5:** Allocate the next highest range, i.e. Accounts. The requirement of 25 IPs can be fulfilled with 192.168.1.192 /27 (255.255.255.224) IP subnet, which contains 30 valid host IPs. The network number of Accounts department will be 192.168.1.192. The last octet of subnet mask is 11100000.

**Step 6:** Allocate the next highest range to Management. The Management department contains only 5 computers. The subnet 192.168.1.224 /29 with the Mask 255.255.255.248 has exactly 6 valid host IP addresses. So, this can be assigned to Management. The last octet of the subnet mask will contain 11111000.

By using VLSM, the administrator can subnet the IP subnet in such a way that least number of IP addresses are wasted. Even after assigning IPs to every department, the administrator, in this example, is still left with plenty of IP addresses which was not possible if he has used FLSM.

**Numerically, we can show the VLSM subnetting process as:**

The given network address is: 192.168.1.0/24

Given requirement in descending order is:

| | |
|---|---|
| Sales | 100 |
| Purchase | 50 |
| Accounts | 25 |
| Management | 5 |

The complete range of the address in the above provided network is:

192.168.1.0 to 192.168.1.255

Divide the given network consisting 256 hosts into 2 networks with 128 hosts each:

192.168.1.0-192.168.1.127 (192.168.1.0/25)

192.168.1.128-192.168.1.255 (192.168.1.128/25)

The largest network requirement is of 100 hosts for Sales department. For this, we need to assign subnetwork with 128 hosts.

Let us assign the first divided subnetwork 192.168.1.0/25 to Sales Department.

We now have remaining subnetwork 192.168.1.128/25.

Dividing this subnetwork, two subnetworks with 64 hosts each are formed.

192.168.1.128 to 192.168.1.191 (192.168.1.128/26)

192.168.1.192 to 192.168.1.255 (192.168.1.192/26)

Our second network requirement is of 50 hosts for Purchase department. We need to assign subnetwork consisting of 64 hosts.

Assigning 192.168.1.128/26 to Purchase department.

The remaining subnetwork available is 192.168.192/26.

Dividing this subnetwork, two subnetworks with 32 hosts each are formed.

192.168.1.192 to 192.168.1.223 (192.168.1.192/27)

192.168.1.224 to 192.168.1.255 (192.168.1.224/27)

The third largest requirement is of 25 hosts for Account department.

Assigning 192.168.1.192/27 to Account Department.

Remaining subnetwork is 192.168.1.224/27

Dividing this subnetwork, two subnetworks with 16 hosts each are formed.

192.168.1.224 to 192.168.1.239 (192.168.1.224/28)

192.168.1.240 to 192.168.1.255 (192.168.1.240/28)

Our fourth network requirement is of 5 hosts for Management department. We need to assign subnetwork consisting of 8 hosts, which is sufficient.

So, again dividing the subnetwork 192.168.1.240/28, two subnetworks with 8 hosts each are formed.

192.168.1.240 to 192.168.1.247 (192.168.1.240/29)

192.168.1.248 to 192.168.1.255 (192.168.1.248/29)

Our fourth network requirement is of 5 hosts for Management department. We need to assign subnetwork consisting of 8 hosts.

We can Assign either of the subnetwork to Management department.

Summarizing the subnetting results,

| **Network Name** | **Network ID** | **Subnet mask** | **No. of usable hosts** | **Usable Host ID Range** | **Broadcast address** |
|---|---|---|---|---|---|
| Sales | 192.168.1.0 | /25 | 126 | 192.168.1.1 to 192.168.1.126 | 192.168.1.127 |
| Purchase | 192.168.1.128 | /26 | 62 | 192.168.129 to 192.168.1.190 | 192.168.1.191 |
| Account | 192.168.1.192 | /27 | 30 | 192.168.1.193 to 192.168.1.222 | 192.168.1.223 |
| Management | 192.168.1.240 | /29 | 6 | 192.168.1.241 to 192.168.1.246 | 192.168.1.247 |
| Unused | 192.168.1.224/28 (192.168.1.224 to 192.168.1.239) | | | | |
| Unused | 192.168.1.247/29 (192.168.1.247 to 192.168.1.255) | | | | |

**FLSM Numerical Example:**

***Q1. If you are given a network 210.25.23.0 with the subnet mask 255.255.255.0, assign the networks to four different departments with 50 hosts each*.**

Ans: The complete range of the address in the above provided network is:

210.25.23.0 to 210.25.23.255

Total no of hosts available: 256 hosts

Each subnetwork requires 50 usable hosts. So, we need to assign n/w with 64 hosts each to the four departments.

Since we are using FLSM, the divided networks will be of same size. The given network consists of 256 hosts which needs to be divided into four subnetworks with 64 hosts each.

The process is as follows:

First of all, divide the given network range into four equal parts.

| | |
|---|---|
| 210.25.23.0 to 210.25.23.63 | (210.25.23.0/26) |
| 210.25.23.64 to 210.25.23.127 | (210.25.23.64/26) |
| 210.25.23.128 to 210.25.23.191 | (210.25.23.128/26) |
| 210.25.23.192 to 210.25.23.255 | (210.25.23.192/26) |

Now, as per the requirement, there are four networks required and we can assign the above networks to each of the four departments.

| **Network Name** | **Network ID** | **Subnet mask** | **No. of usable hosts** | **Usable Host ID Range** | **Broadcast address** |
|---|---|---|---|---|---|
| Dept 1 | 210.25.23.0 | /26 | 62 | 210.25.23.1 to 210.25.23.62 | 210.25.23.63 |
| Dept 2 | 210.25.23.64 | /26 | 62 | 210.25.23.65 to 210.25.23.126 | 210.25.23.127 |
| Dept 3 | 210.25.23.128 | /26 | 62 | 210.25.23.129 to 210.25.23.190 | 210.25.23.191 |
| Dept 4 | 210.25.23.192 | /26 | 62 | 210.25.23.193 to 210.25.23.254 | 210.25.23.255 |

***Q2. Suppose you are network administrator with provided network 172.16.0.0/24. You need to manage the entire n/w by dividing into subnetworks so that each of the Development, Sales, Reception, HR and Production. How would you do so?***

Ans: Provided network: 172.16.0.0/24. Here, /24 indicates 256 hosts are contained in the given network.

There are five departments to address the networks with. So, we divide the given network into 8 networks. 256/8 = 32

Each of the 8 subnetworks will contain 32 hosts each. The divided networks will be:

| | |
|---|---|
| 172.16.0.0 to 172.16.0.31 | (172.16.0.0/27) |
| 172.16.0.32 to 172.16.0.63 | (172.16.0.32/27) |
| 172.16.0.64 to 172.16.0.95 | (172.16.0.64/27) |
| 172.16.0.96 to 172.168.0.127 | (172.16.0.96/27) |
| 172.16.0.128 to 172.16.0.159 | (172.16.0.128/27) |
| 172.16.0.160 to 172.16.0.191 | (172.16.0.160/27) |
| 172.16.0.192 to 172.16.0.223 | (172.16.0.192/27) |
| 172.16.0.224 to 172.16.0.255) | (172.16.0.224/27) |

Now, we can assign 5 of the above 8 subnetworks to the departments of our requirement.

The result will be as follows:

| **Network Name** | **Network ID** | **Subnet mask** | **No. of usable hosts** | **Usable Host ID Range** | **Broadcast address** |
|---|---|---|---|---|---|
| Development | 172.16.0.0 | /27 | 30 | 172.16.0.1 to 172.16.0.30 | 172.16.0.31 |
| Sales | 172.16.0.32 | /27 | 30 | 172.16.0.33 to 172.16.0.62 | 172.16.0.63 |
| Reception | 172.16.0.64 | /27 | 30 | 172.16.0.65 to 172.16.0.94 | 172.16.0.95 |
| HR | 172.16.0.96 | /27 | 30 | 172.16.0.97 to 172.168.0.126 | 172.168.0.127 |
| Production | 172.16.0.128 | /27 | 30 | 172.16.0.129 to 172.16.0.158 | 172.16.0.159 |
| Unused | 172.16.0.160 to 172.16.0.191 (172.16.0.160/27) | | | | |
| Unused | 172.16.0.192 to 172.16.0.223 (172.16.0.192/27) | | | | |
| Unused | 172.16.0.224 to 172.16.0.255) (172.16.0.224/27) | | | | |

**VLSM Numerical Example:**

***Q. If you are assigned an IP address 92.16.1.0/24 and plans to deploy CIDR. Here are some requirements which you have to fulfill for Subnet A= 120 hosts, Subnet B=60 hosts, Subnet C=30 hosts, Subnet D= 10 hosts, Subnet E= 5. You are also required to calculate subnet mask, range, netid, broadcast id for each subnet.***

Ans: The given network address is: 92.16.1.0/24

Given requirement in descending order is:

Subnet A: 120

Subnet B: 60

Subnet C: 30

Subnet D: 10

Subnet E: 5

The complete range of the address in the above provided network is:

92.16.1.0 to 92.16.1.255

The largest network requirement is of 120 hosts for Subnet A. For this, we need to assign subnetwork with 128 hosts.

Divide the given network consisting 256 hosts into 2 networks with 128 hosts each:

| | |
|---|---|
| 92.16.1.0-92.16.1.127 | (92.16.1.0/25) |
| 92.16.1.128-92.16.1.255 | (92.16.1.128/25) |

Let us assign the first divided subnetwork 92.16.1.0/25 to Subnet A.

We now have remaining subnetwork 92.16.1.128/25.

Our second network requirement is of 60 hosts for Subnet B. We need to assign subnetwork consisting of 64 hosts.

Dividing this subnetwork, two subnetworks with 64 hosts each are formed.

| | |
|---|---|
| 92.16.1.128 to 92.16.1.191 | (92.16.1.128/26) |
| 92.16.1.192 to 92.16.1.255 | (92.16.1.192/26) |

Assigning 92.16.1.128/26 to Subnet B.

The remaining subnetwork available is 92.16.1.192/26.

The third largest requirement is of 30 hosts for Subnet C.

Dividing this subnetwork, two subnetworks with 32 hosts each are formed.

92.16.1.192 to 92.16.1.223 (92.16.1.192/27)

92.16.1.224 to 92.16.1.255 (92.16.1.224/27)

Assigning 92.16.1.192/27 to Subnet C.

Remaining subnetwork is 92.16.1.224/27

Our fourth network requirement is of 10 hosts for Subnet D. We need to assign subnetwork consisting of 16 hosts.

Dividing this subnetwork, two subnetworks with 16 hosts each are formed.

92.16.1.224 to 92.16.1.239 (92.16.1.224/28)

92.16.1.240 to 92.16.1.255 (92.16.1.240/28)

Assigning 92.16.1.224/28 to Subnet D.

Remaining subnetwork is 92.16.1.240/28

Our fifth network requirement is of 5 hosts for Subnet E. We need to assign subnetwork consisting of 8 hosts.

So, again dividing the subnetwork 92.16.1.240/28, two subnetworks with 8 hosts each are formed.

92.16.1.240 to 92.16.1.247 (92.16.1.240/29)

92.16.1.248 to 92.16.1.255 (92.16.1.248/29)

We can Assign either of the subnetwork to Subnet E. Let us assign 92.16.1.240/29 to Subnet E.

Summarizing the subnetting results,

| **Network Name** | **Network ID** | **Subnet mask** | **No. of usable hosts** | **Usable Host ID Range** | **Broadcast address** |
|---|---|---|---|---|---|
| Subnet A | 92.16.1.0 | /25 | 126 | 92.16.1.1 to 92.16.1.126 | 92.16.1.127 |
| Subnet B | 92.16.1.128 | /26 | 62 | 92.16.129 to 92.16.1.190 | 92.16.1.191 |
| Subnet C | 92.16.1.192 | /27 | 30 | 92.16.1.193 to 92.16.1.222 | 92.16.1.223 |
| Subnet D | 92.16.1.224 | /28 | 14 | 92.16.1.225 to 92.16.1.238 | 92.16.1.239 |
| Subnet E | 92.16.1.240 | /29 | 6 | 92.16.1.241 to 92.16.1.246 | 92.16.1.247 |
| Unused | 92.16.1.248/29 (92.16.1.248 to 92.16.1.255) | | | | |

Note:

1. Network: 192.168.0.0/24, 2^8, 256 hosts

   Total Range: 192.168.0.0 to 192.168.0.255

2. Network: 192.168.1.0/25, 2^7, 128 hosts
   Total Range: 192.168.1.0 to 192.168.1.127

3. Network: 192.168.3.0/26, 2^6, 64 hosts
   Total Range: 192.168.3.0 to 192.168.3.63

4. Network: 192.168.0.0/23, 2^9, 512 hosts
   Total Range: 192.168.0.0 to 192.168.0.255, 192.168.1.0 to 192.168.1.255

5. Network: 192.168.1.0/23, 2^9, 512 hosts
   Total Range: 192.168.1.0 to 192.168.1.255, 192.168.2.0 to 192.168.2.255

6. Network: 172.16.10.0/23, 2^9, 512 hosts
   Total Range: 172.16.10.0 to 172.16.10.255, 172.16.11.0 to 172.168.11.255

7. Network: 172.16.10.0/22, 2^10, 1024 hosts
   Total Range: 172.16.10.0 to 172.16.10.255
   172.16.11.0 to 172.16.11.255
   172.16.12.0 to 172.16.12.255
   172.16.13.0 to 172.16.13.255

8. Network: 172.16.10.0/21, 2^11, 2048 hosts
   Total Range: 172.16.10.0 to 172.16.10.255
   172.16.11.0 to 172.16.11.255
   172.16.12.0 to 172.16.12.255
   172.16.13.0 to 172.16.13.255
   172.16.14.0 to 172.16.14.255
   172.16.15.0 to 172.16.15.255
   172.16.16.0 to 172.16.16.255
   172.16.17.0 to 172.16.17.255

**Q2. Given Network: 192.168.0.0/23**

**Requirement:**

**A: 128 hosts, B: 64 hosts, C: 31 hosts, D: 15 hosts**

Solution: Total Range= 192.168.0.0 to 192.168.0.255 (192.168.0.0/24)

192.168.1.0 to 192.168.1.255 (192.168.1.0/24)

A-> 128 hosts, need to assign n/w of 256 hosts

Let us assign: 192.168.0.0/24

B-> 64 hosts, need to assign n/w of 128 hosts

Divide 192.168.1.0/24,

    192.168.1.0 to 192.168.1.127      (192.168.1.0/25)
    192.168.1.128 to 192.168.1.255 (192.168.1.128/25)

Assign 192.168.1.0/25 to B.

Remaining: 192.168.1.128/25

C->31 hosts, need to assign n/w of 64 hosts

Divide 192.168.1.128/25,

    192.168.1.128 to 192.168.1.191          (192.168.1.128/26)
    192.168.1.192 to 192.168.1.255          (192.168.1.192/26)

Assign 192.168.1.128/26 to C.

Remaining: 192.168.1.192/26

D-> 15 hosts, need to assign n/w of 32 hosts

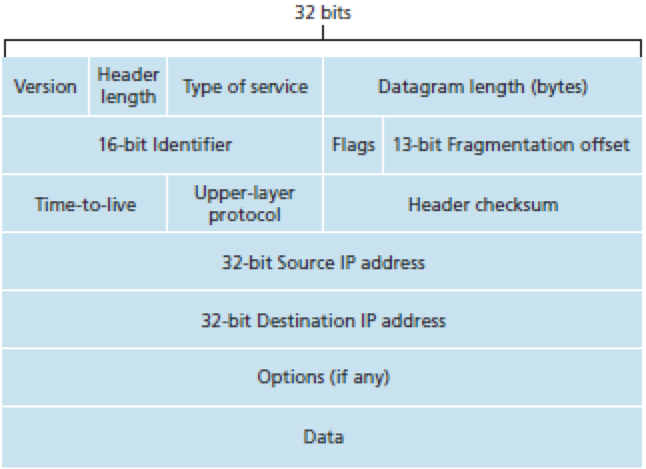
Divide 192.168.1.192/26,

    192.168.1.192 to 192.168.1.223          (192.168.1.192/27)
    192.168.1.224 to 192.168.1.255          (192.168.1.224/27)

Assign 192.168.1.192/27 to D

Unused: 192.168.1.224/27

**<u>IPv4 Header Format:</u>**



The key fields in the IPv4 datagram are the following:

- Version number. These 4 bits specify the IP protocol version of the datagram. By looking at the version number, the router can determine how to interpret the remainder of the IP datagram. Different versions of IP use different datagram formats. The datagram format for the current version of IP, IPv4, is shown in Figure.
- Header length. Because an IPv4 datagram can contain a variable number of options (which are included in the IPv4 datagram header), these 4 bits are needed to determine where in the IP datagram the data actually begins. Most IP datagrams do not contain options, so the typical IP datagram has a 20-byte header.
- Type of service. The type of service (TOS) bits were included in the IPv4 header to allow different types of IP datagrams (for example, datagrams particularly requiring low delay, high throughput, or reliability) to be distinguished from each other. For example, it might be useful to distinguish real-time datagrams (such as those used by an IP telephony application) from non-real-time traffic (for example, FTP). The specific level of service to be provided is a policy issue determined by the router's administrator.
- Datagram length. This is the total length of the IP datagram (header plus data), measured in bytes. Since this field is 16 bits long, the theoretical maximum size of the IP datagram is 65,535 bytes. However, datagrams are rarely larger than 1,500 bytes.
- Identifier, flags, fragmentation offset. These three fields have to do with so-called IP fragmentation. Interestingly, the new version of IP, IPv6, does not allow for fragmentation at routers.
- Time-to-live. The time-to-live (TTL) field is included to ensure that datagrams do not circulate forever (due to, for example, a long-lived routing loop) in the network. This field is decremented by one each time the datagram is processed by a router. If the TTL field reaches 0, the datagram must be dropped.
- Protocol. This field is used only when an IP datagram reaches its final destination. The value of this field indicates the specific transport-layer protocol to which the data portion of this IP datagram should be passed. For example, a value of 6 indicates that the data portion is passed to

TCP, while a value of 17 indicates that the data is passed to UDP. Note that the protocol number in the IP datagram has a role that is analogous to the role of the port number field in the transport layer segment. The protocol number is the glue that binds the network and transport layers together, whereas the port number is the glue that binds the transport and application layers together. The link-layer frame also has a special field that binds the link layer to the network layer.

- Header checksum. The header checksum aids a router in detecting bit errors in a received IP datagram. The header checksum is computed by treating each 2 bytes in the header as a number and summing these numbers using 1s complement arithmetic. The 1s complement of this sum, known as the Internet checksum, is stored in the checksum field. A router computes the header checksum for each received IP datagram and detects an error condition if the checksum carried in the datagram header does not equal the computed checksum. Routers typically discard datagrams for which an error has been detected. Note that the checksum must be recomputed and stored again at each router, as the TTL field, and possibly the options field as well, may change.
- Source and destination IP addresses. When a source creates a datagram, it inserts its IP address into the source IP address field and inserts the address of the ultimate destination into the destination IP address field. Often the source host determines the destination address via a DNS lookup.
- Options. The options fields allow an IP header to be extended. Header options were meant to be used rarely—hence the decision to save overhead by not including the information in options fields in every datagram header. However, the mere existence of options does complicate matters—since datagram headers can be of variable length, one cannot determine a priori where the data field will start. Also, since some datagrams may require options processing and others may not, the amount of time needed to process an IP datagram at a router can vary greatly. These considerations become particularly important for IP processing in high-performance routers and hosts. For these reasons and others, IP options were dropped in the IPv6 header.
- Data (payload). Finally, we come to the last and most important field—the raison d'être for the datagram in the first place! In most circumstances, the data field of the IP datagram contains the transport-layer segment (TCP or UDP) to be delivered to the destination. However, the data field can carry other types of data, such as ICMP messages.

**<u>Issues with IPv4:</u>**

Changes since IPv4 was developed (mid 70's)

- Provider market has changed dramatically
- Immense increase in user and traffic on the Internet
- Rapid technology advancement
- Bandwidth increase from kb/s to Tb/s

IPv4 issues: The major issues in IPv4 are

- Deficiency of address space - The devices connected to the Internet grows exponentially. The size of address space 2^32 is quickly exhausted;
- Too large routing tables

Some more issues are:

- Weak expansibility of the protocol - the insufficient size of heading IPv4 doesn't allow to place demanded quantity of additional parameters in it;
- Problem of safety of communications - it is not stipulated any means for differentiation of access to the information placed in a network;
- Absence of support of quality of service (QoS) - accommodation of the information about throughput, the delays and demanded for normal work of some network appendices is not supported;
- The problems connected with the mechanism of a fragmentation - the size of the maximal block of data transmission on each concrete way is not defined;
- Absence of the auto-configuration IP addresses mechanism.

**Overview of IPv6:**

To respond to the need for a large IP address space, a new IP protocol, IPv6, was developed. Also, major issues of IPv4 are addressed in this version.

The most important changes introduced in IPv6 are evident in the datagram format:

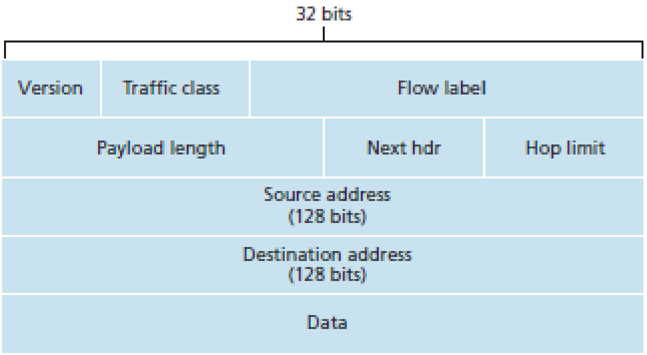
- *Expanded addressing capabilities.* IPv6 increases the size of the IP address from 32 to 128 bits. This ensures that the world won’t run out of IP addresses. Now, every grain of sand on the planet can be IP-addressable. In addition to unicast and multicast addresses, IPv6 has introduced a new type of address, called an **any-cast address**, which allows a datagram to be delivered to any one of a group of hosts.
- *A streamlined 40-byte header.* As discussed below, a number of IPv4 fields have been dropped or made optional. The resulting 40-byte fixed-length header allows for faster processing of the IP datagram. A new encoding of options allows for more flexible options processing.
- *Flow labeling and priority.* IPv6 has an elusive definition of a flow. This allows “labeling of packets belonging to particular flows for which the sender requests special handling, such as a non-default quality of service or real-time service.” For example, audio and video transmission might likely be treated as a flow.

**IPv6 Simplifications:**

- Remove header checksum: Because the transport-layer (for example, TCP and UDP) and link-layer (for example, Ethernet) protocols in the Internet layers perform check summing, the designers of IP probably felt that this functionality was sufficiently redundant in the network layer that it could be removed.
- Remove hop-by-hop segmentation: IPv6 does not allow for fragmentation and reassembly at intermediate routers; these operations can be performed only by the source and destination. If an IPv6 datagram received by a router is too large to be forwarded over the outgoing link, the router simply drops the datagram and sends a “Packet Too Big” ICMP error message back to the sender.

- Options. An options field is no longer a part of the standard IP header. However, it has not gone away. Instead, the options field is one of the possible next headers pointed to from within the IPv6 header. The removal of the options field results in a fixed-length, 40-byte IP header.

**<u>IPv6 Header:</u>**



| **S.N.** | **Field & Description** |
|---|---|
| 1 | Version (4-bits): It represents the version of Internet Protocol, i.e. 0110 |
| 2 | Traffic Class (8-bits): These 8 bits are divided into two parts. The most significant 6 bits are used for Type of Service to let the Router know what services should be provided to this packet. The least significant 2 bits are used for Explicit Congestion Notification (ECN). |
| 3 | Flow Label (20-bits): This label is used to maintain the sequential flow of the packets belonging to a communication. The source labels the sequence to help the router identify that a particular packet belongs to a specific flow of information. |
| 4 | Payload Length (16-bits): This field is used to tell the routers how much information a particular packet contains in its payload. Payload is composed of Extension Headers and Upper Layer data. |
| 5 | Next Header (8-bits): This field is used to indicate either the type of Extension Header, or if the Extension Header is not present, then it indicates the Upper Layer PDU. |

| | |
|---|---|
| 6 | Hop Limit (8-bits): This field is used to stop packet to loop in the network infinitely. This is same as TTL in IPv4. The value of Hop Limit field is decremented by 1 as it passes a link (router/hop). When the field reaches 0, the packet is discarded. |
| 7 | Source Address (128-bits): This field indicates the address of originator of the packet. |
| 8 | Destination Address (128-bits): This field provides the address of intended recipient of the packet. |

**<u>IPv6 Addresses :( IPv6 Format)</u>**

IPv6 address is 128 bits long and is arranged in eight groups, each of which is 16 bits. Each group is expressed as four hexadecimal digits and the groups are separated by colons.

An example of a full IPv6 address: FE80:CD00: 0000:0CDE: 1257:0000:211E:729C

IPv6 has three address categories:

- Unicast - identifies exactly one interface
- Multicast - identifies a group; packets get delivered to all members of the group
- Anycast - identifies a group; packets normally get delivered to nearest member of the group

**<u>IPv6 Address Abbreviations and CIDR:</u>**

Even after converting into Hexadecimal format, IPv6 address remains long. An IPv6 address may be abbreviated to shorter notations by application of the following rules:

**Rule 1**: Discard leading zero (es)

That address can be shortened because the addressing scheme allows the omission of any leading zero, as well as any sequences consisting only of zeroes.

E.g.: FE80:CD00:0000:0CDE:1257:0000:211E:729C

Here's the short version:

FE80:CD00:0:CDE:1257:0:211E:729C

**Rule 2:** If two of more blocks contain consecutive zeroes, omit them all and replace with double colon sign **::**

2001:0000:3238:DFE1:63:0000:0000:FEFB

can be written as

2001:0000:3238:DFE1:63::FEFB

The IPv6 addressing architecture allows you use the two-colon (::) notation to represent contiguous 16-bit fields of zeros.

CIDR Notation is similar to IPv4 addresses, IPv6 addresses consist of NetworkID + HostID, and use classless notation to identify (distinguish between) the two. Network ID is also referred to as prefix, and the number of bits allocated to Network ID as prefix length. Information on the prefix is provided together with each IPv6 address as a slash (/) at the end of the address followed by the prefix length.

For example, the site prefix of the IPv6 address 2001:db8:3c4d:0015:0000:0000:1a2f:1a2b/48 is contained in the leftmost 48 bits, 2001:db8:3c4d. You use the following representation, with zeros compressed, to represent this prefix: 2001:db8:3c4d::/48

**IPv6 vs IPv4:**

| **IPv4** | **IPv6** |
|---|---|
| IPv4 addresses are 32 bit length. | IPv6 addresses are 128 bit length. |
| IPv4 addresses are binary numbers represented in decimals. | IPv6 addresses are binary numbers represented in hexadecimals. |
| IPSec support is only optional. | Inbuilt IPSec support. |
| Fragmentation is done by sender and forwarding routers. | Fragmentation is done only by sender. |
| No packet flow identification | Packet flow identification is available within the IPv6 header using the Flow Label field. |
| Checksum field is available in IPv4 header | No checksum field in IPv6 header |
| Options fields are available in IPv4 header | No option fields, but IPv6 Extension headers are available. |

**Transition from IPv4 to IPv6:**

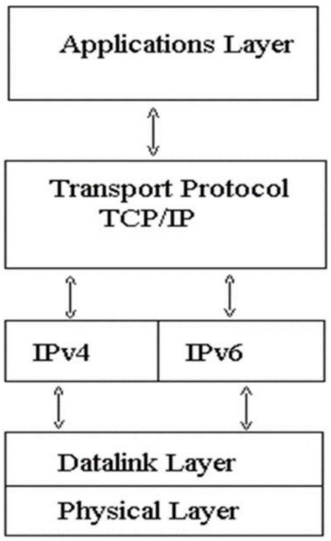
Because of the huge number of systems on the internet, the transition from IPv4 to IPv6 cannot happen suddenly. The transition must be smooth to prevent any problems between IPv4 and IPv6 systems.

Three strategies have been devised to help the transition:

- Dual stack
- Tunneling
- Header translation

Dual Stack:

Dual-stack transition mechanism enables to run both IP stacks (IPv4 and IPv6) in a single node. Maintains both IP protocol stacks that operates parallel and thus allow the end node to use either protocols. Node is capable of handling both kinds of IP (IPv4&IPv6) routing. Flow or routing decisions in the node are based on IP header version's field. Both IPv4 and IPv6 shares common transport layer protocols such as TCP/IP. Many of client and server operating systems provide dual IP protocol stacks. For example: Windows 7, 8, Linux



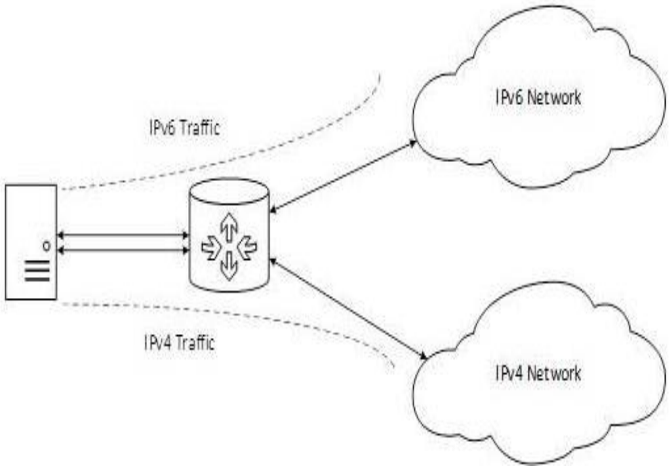


Fig: Dual Stack Router

Fig: Dual stack TCP/IP model

The Dual Stack Router, can communicate with both the networks. It provides a medium for the hosts to access a server without changing their respective IP versions.

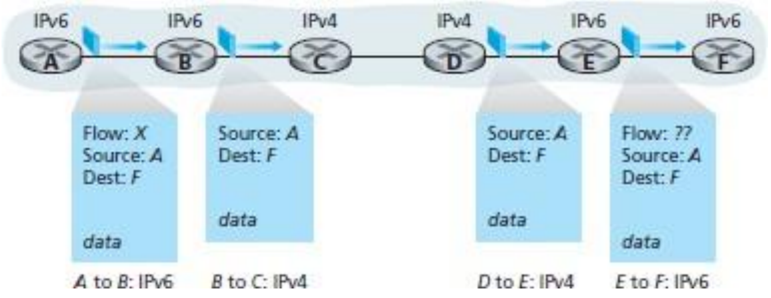
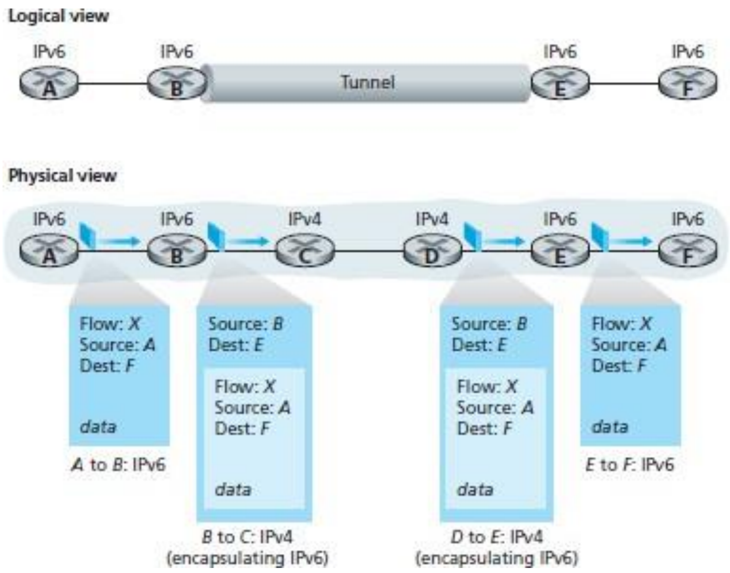


Fig: A dual-stack approach

In the dual-stack approach, if either the sender or the receiver is only IPv4-capable, an IPv4 datagram must be used. As a result, it is possible that two IPv6-capable nodes can end up, in essence, sending IPv4 datagrams to each other. Suppose Node A is IPv6-capable and wants to send an IP datagram to Node F, which is also IPv6-capable. Nodes A and B can exchange an IPv6 datagram. However, Node B must create an IPv4 datagram to send to C. Certainly, the data field of the IPv6 datagram can be copied into the data field of the IPv4 datagram and appropriate address mapping can be done. However, in performing the conversion from IPv6 to IPv4, there will be IPv6-specific fields in the IPv6 datagram (for example, the flow identifier field) that have no counterpart in IPv4. The information in these fields will be lost. Thus, even though E and F can exchange IPv6 datagrams, the arriving IPv4 datagrams at E from D do not contain all of the fields that were in the original IPv6 datagram sent from A.

Tunneling:

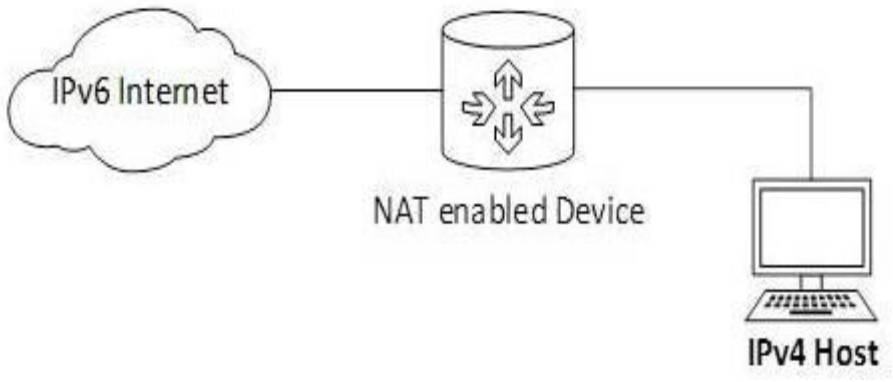
Tunneling is a strategy used when two computers using IPv4 want to communicate with each other and the packet must pass through a region that uses IPv6. To pass through this region, the packet must have an IPv6 address. So the IPv4 packet is encapsulated in an IPv6 packet when it enters the region, and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



An alternative to the dual-stack approach is known as tunneling. Tunneling can solve the problem noted above, allowing, for example, E to receive the IPv6 datagram originated by A. The basic idea behind tunneling is the following. Suppose two IPv6 nodes (for example, B and E in Figure) want to interoperate using IPv6 datagrams but are connected to each other by intervening IPv4 routers. We refer to the intervening set of IPv4 routers between two IPv6 routers as a tunnel, as illustrated in Figure. With tunneling, the IPv6 node on the sending side of the tunnel (for example, B) takes the entire IPv6 datagram and puts it in the data (payload) field of an IPv4 datagram. This IPv4 datagram is then addressed to the IPv6 node on the receiving side of the tunnel (for example, E) and sent to the first node in the tunnel (for example, C). The intervening IPv4 routers in the tunnel route this IPv4 datagram among themselves, just as they would any other datagram, blissfully unaware that the IPv4 datagram itself contains a complete IPv6 datagram. The IPv6 node on the receiving side of the tunnel eventually receives the IPv4 datagram (it is the destination of the IPv4 datagram!), determines that the IPv4 datagram contains an IPv6 datagram, extracts the IPv6 datagram, and then routes the IPv6 datagram exactly as it would if it had received the IPv6 datagram from a directly connected IPv6 neighbor.

Header Translation:

Translation mechanism refers the direct conversion of IP protocols. May include transformation of both IPv4 and IPv6 protocol's header and payload according to their IP specifications. Translation mechanisms always need translators that can translate particular IPv4 address to particular IPv6 address and vice versa. A host with IPv4 address sends a request to an IPv6 enabled server on Internet that does not understand IPv4 address. In this scenario, the NAT-PT device can help them communicate. When the IPv4 host sends a request packet to the IPv6 server, the NAT-PT device/router strips down the IPv4 packet, removes IPv4 header, and adds IPv6 header and passes it through the Internet. When a response from the IPv6 server comes for the IPv4 host, the router does vice versa.



**Routing:**

Routing is the process of selecting a path for traffic in a network, or between or across multiple networks. It refers to establishing the routes that data packets take on their way to a particular destination. In general, routing involves the network topology, or the setup of hardware, that can effectively relay data. Standard protocols help to identify the best routes for data and to ensure quality transmission. Individual pieces of hardware such as routers are referred to as "nodes" in the network. Different algorithms and protocols can be used to figure out how to best route data packets, and which nodes should be used. There are 3 types of routing:
**Static routing** – Static routing is a process in which we have to manually add routes in routing table.

Advantages –

- No routing overhead for router CPU which means a cheaper router can be used to do routing.
- It adds security because only administrator can allow routing to particular networks only.
- No bandwidth usage between routers.

Disadvantage –

- For a large network, it is a hectic task for administrator to manually add each route for the network in the routing table on each router.

- The administrator should have good knowledge of the topology. If a new administrator comes, then he has to manually add each route so he should have very good knowledge of the routes of the topology.

**Default Routing** –This is the method where the router is configured to send all packets towards a single router (next hop). It doesn't matter to which network the packet belongs, it is forwarded out to router which is configured for default routing. It is generally used with stub routers. A stub router is a router which has only one route to reach all other networks.

**Fixed Path Routing:**

A route is selected for each source and destination pair of nodes in the network. The routes are fixed and changes only if topology of the network changes. It is sometimes also referred to as static routing since the routes are fixed as in static routing.

**Flooding:**

Flooding adapts the technique in which every incoming packet is sent on every outgoing line expect from which it arrived. One problem with this is that packets may go in a loop and as a result of which a node may receive duplicate packets. These problems can overcome with the help of sequence numbers and hop count. No routing table is required for flooding and no network information like topology, load condition, cost of different paths is required. All possible routes between source and destination is tried, and there will be at least one route which is the shortest.

**Unicast vs Multicast Routing:**

A Unicast transmission/routing sends IP packets to a single recipient on a network. If the streaming data is to be distributed to a single destination, then we should start a Unicast stream by setting the destination IP address. For example, a device having IP address 10.1.2.0 in a network wants to send the traffic stream (data packets) to the device with IP address 20.12.4.2 in the other network, then unicast comes into the picture. This is the most common form of data transfer over the networks.

A Multicast transmission sends IP packets to a group of hosts on a network. IP multicast requires support of some other protocols like IGMP (Internet Group Management Protocol), Multicast routing for its working. Also, in Classful IP addressing Class D is reserved for multicast groups. If we want to distribute the data at multiple concurrent locations/destinations, then we should set the destination IP address to a valid Multicast IP address (224.0.0.0 – 239.255.255.255). Since Multicasting is a relatively new technology, some legacy devices that are part of the network might not support Multicasting.

**Dynamic Routing** –Dynamic routing makes automatic adjustment of the routes according to the current state of the route in the routing table. Dynamic routing uses protocols to discover network destinations and the routes to reach it. RIP and OSPF are the best examples of dynamic routing protocol. Automatic adjustment will be made to reach the network destination if one route goes down.

A dynamic protocol has following features:

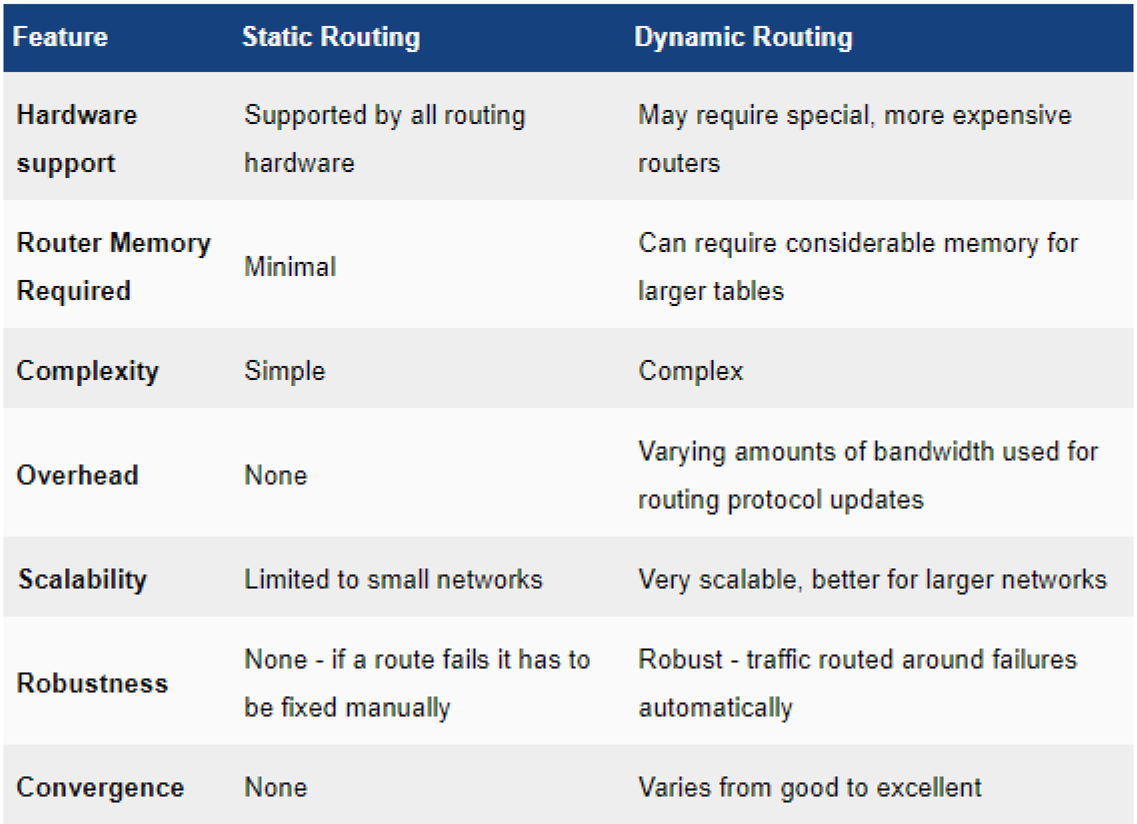
- The routers should have the same dynamic protocol running in order to exchange routes.
- When a router finds a change in the topology then router advertises it to all other routers.

Advantages –

- Easy to configure.
- More effective at selecting the best route to a destination remote network and also for discovering remote network.

Disadvantage –

- Consumes more bandwidth for communicating with other neighbors.
- Less secure than static routing.



| Feature | Static Routing | Dynamic Routing |
|---|---|---|
| **Hardware support** | Supported by all routing hardware | May require special, more expensive routers |
| **Router Memory Required** | Minimal | Can require considerable memory for larger tables |
| **Complexity** | Simple | Complex |
| **Overhead** | None | Varying amounts of bandwidth used for routing protocol updates |
| **Scalability** | Limited to small networks | Very scalable, better for larger networks |
| **Robustness** | None - if a route fails it has to be fixed manually | Robust - traffic routed around failures automatically |
| **Convergence** | None | Varies from good to excellent |

Dynamic routing is a networking technique that provides optimal data routing. Unlike static routing, dynamic routing enables routers to select paths according to real-time logical network layout changes. In dynamic routing, the routing protocol operating on the router is responsible for the creation, maintenance and updating of the dynamic routing table. In static routing, all these jobs are manually done by the system administrator. The cost of routing is a critical factor for all organizations. The least-expensive routing technology is provided by dynamic routing, which automates table changes and provides the best paths for data transmission.

Typically, dynamic routing protocol operations can be explained as follows:

- The router delivers and receives the routing messages on the router interfaces.
- The routing messages and information are shared with other routers, which use exactly the same routing protocol.
- Routers swap the routing information to discover data about remote networks.

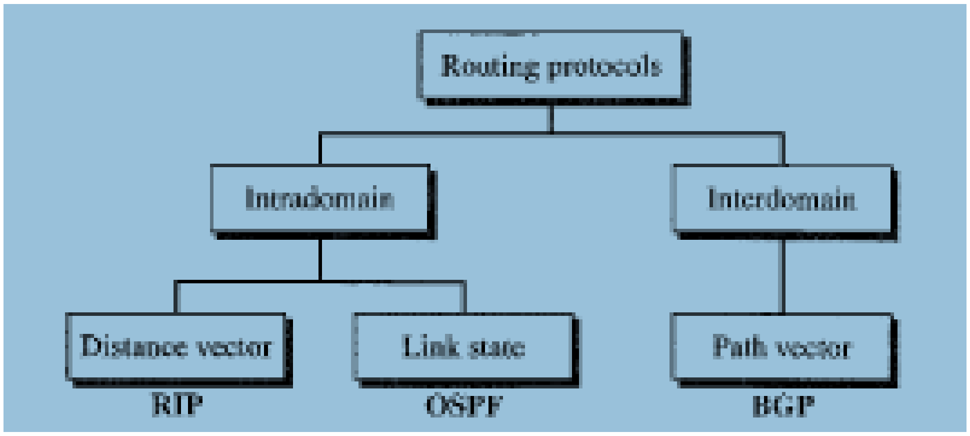
- Whenever a router finds a change in topology, the routing protocol advertises this topology change to other routers.

Dynamic routing is easy to configure on large networks and is more intuitive at selecting the best route, detecting route changes and discovering remote networks. However, because routers share updates, they consume more bandwidth than in static routing; the routers' CPUs and RAM may also face additional loads as a result of routing protocols. Also, dynamic routing is less secure than static routing. Dynamic routing uses multiple algorithms and protocols. The most popular are Routing Information Protocol (RIP) and Open Shortest Path First (OSPF).

**<u>Popular Routing Algorithms:</u>**

A routing algorithm is a set of step-by-step operations used to direct Internet traffic efficiently. When a packet of data leaves its source, there are many different paths it can take to its destination. The routing algorithm is used to determine mathematically the best path to take.

Dynamic routing algorithms are basically categorized as follows:



**Interior vs Exterior Routing:**

Interior routing is a Routing mechanism which is used to find network path information within an Autonomous System. Known Interior Routing Protocols are Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS).

Exterior routing is a Routing mechanism which is used to find network path information between different Autonomous Systems. Exterior Routing Protocols are commonly used in the Internet to exchange routing table information. There is only one Exterior routing protocol exists now and it is Border Gateway Protocol (BGP).

**Shortest Path Routing:**

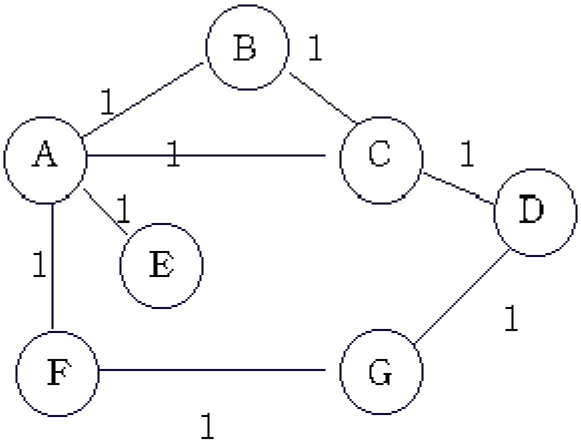
Shortest path routing refers to the process of finding paths through a network that have a minimum of distance or other cost metric. Shortest-path routing algorithms have existed since two independent research works by Bellman and Ford, and Dijkstra in 1950's. The difference between these two algorithms is the way information needed for computing the shortest-path is used. In the context of packet-switched networks and Internet routing, in particular, Bellman-Ford's algorithm has enabled the development of distance-vector routing protocols while Dijkstra's algorithm has paved the way to the introduction of link-state routing protocols.

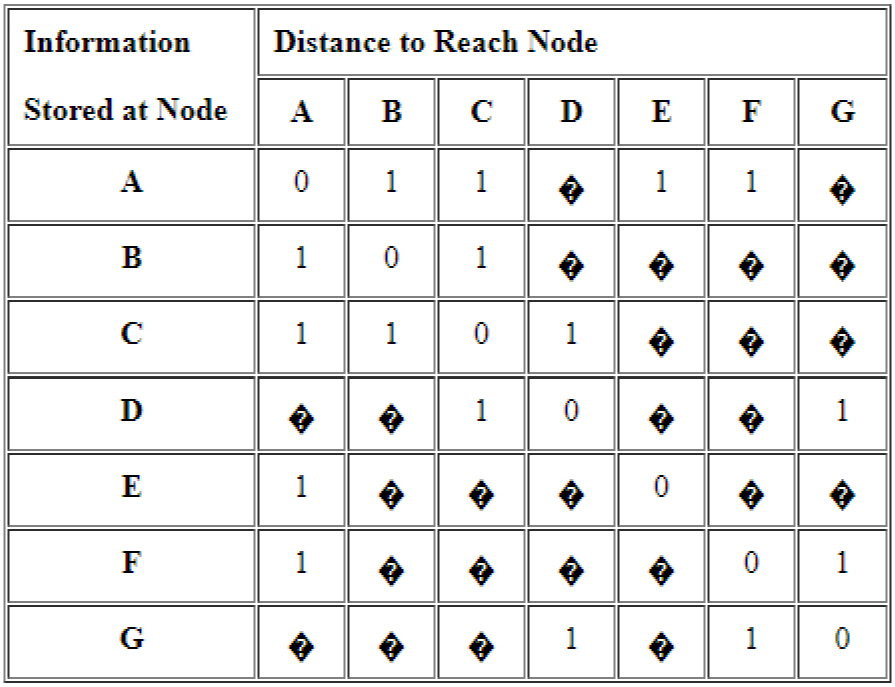
**Distance Vector Routing Algorithm:**

A distance-vector routing protocol in data networks determines the best route for data packets based on distance. Distance-vector routing protocols measure the distance by the number of routers a packet has to pass, one router counts as one hop. The vector describes the route of the message over a given set of network nodes. To determine the best route across a network router on which a distance-vector protocol is implemented exchange information with one another, usually routing tables plus hop counts for destination networks and possibly other traffic information. The basic idea here is that each node receives some information from one or more of its directly attached neighbors, performs a calculation, and then distributes the results of its calculation back to its neighbors.

Distance vector routing algorithm is also called **Bellman Ford algorithm**. Each router maintains a Distance Vector table containing the distance between itself and all possible destination nodes. Distances, based on a chosen metric, are computed using information from the neighbors' distance vectors. The starting assumption for distance-vector routing is that each node knows the cost of the link to each of its directly connected neighbors. A link that is down is assigned an infinite cost.

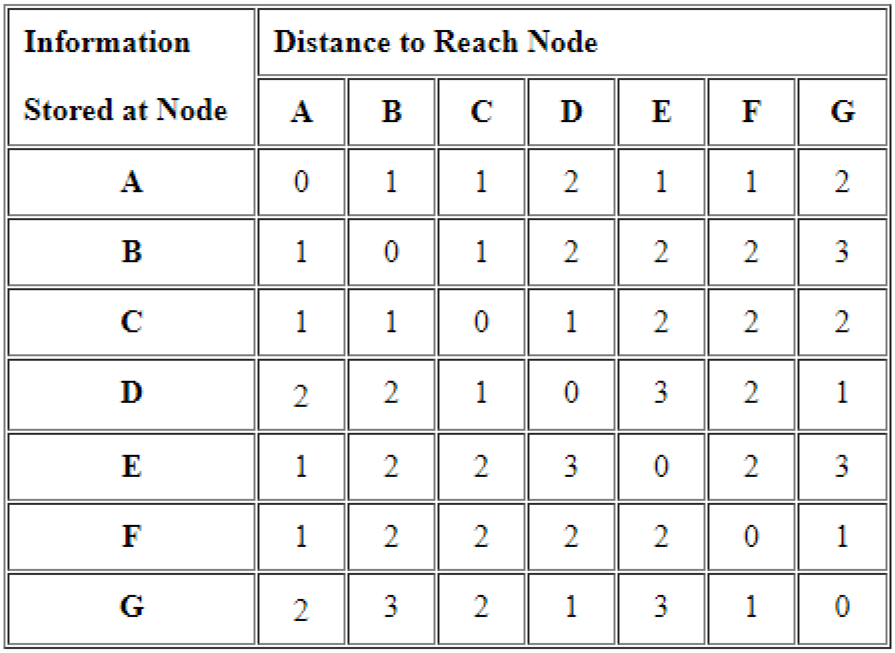
E.g.:





| Information Stored at Node | Distance to Reach Node | | | | | | |
|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G |
| **A** | 0 | 1 | 1 | � | 1 | 1 | � |
| **B** | 1 | 0 | 1 | � | � | � | � |
| **C** | 1 | 1 | 0 | 1 | � | � | � |
| **D** | � | � | 1 | 0 | � | � | 1 |
| **E** | 1 | � | � | � | 0 | � | � |
| **F** | 1 | � | � | � | � | 0 | 1 |
| **G** | � | � | � | 1 | � | 1 | 0 |

**Table 1. Initial distances stored at each node(global view).**

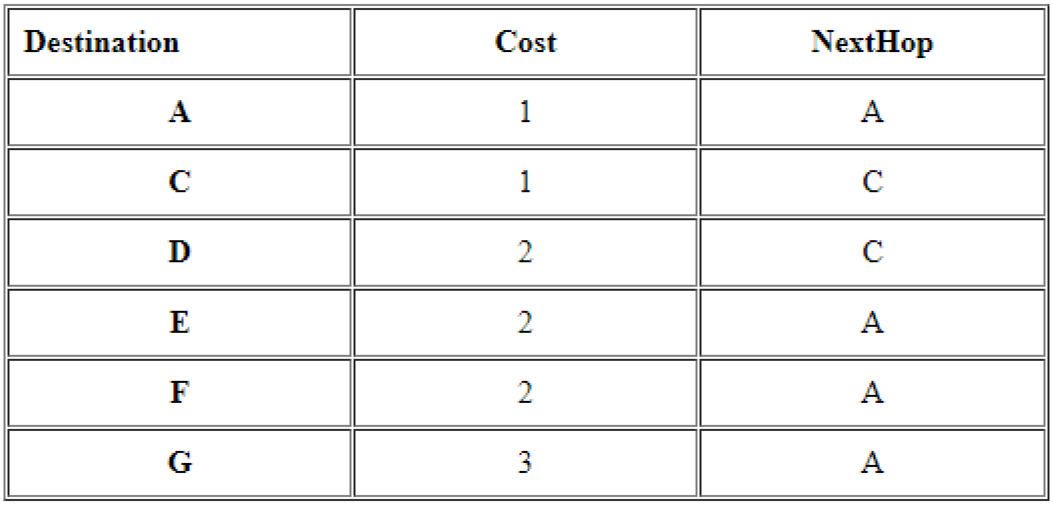


| Information Stored at Node | Distance to Reach Node | | | | | | |
|---|---|---|---|---|---|---|---|
| | A | B | C | D | E | F | G |
| **A** | 0 | 1 | 1 | 2 | 1 | 1 | 2 |
| **B** | 1 | 0 | 1 | 2 | 2 | 2 | 3 |
| **C** | 1 | 1 | 0 | 1 | 2 | 2 | 2 |
| **D** | 2 | 2 | 1 | 0 | 3 | 2 | 1 |
| **E** | 1 | 2 | 2 | 3 | 0 | 2 | 3 |
| **F** | 1 | 2 | 2 | 2 | 2 | 0 | 1 |
| **G** | 2 | 3 | 2 | 1 | 3 | 1 | 0 |

**Table 2. final distances stored at each node ( global view).**

In practice, each node's forwarding table consists of a set of triples of the form: (Destination, Cost, Next Hop).

For example, Table below shows the complete routing table maintained at node B for the network in figure above.



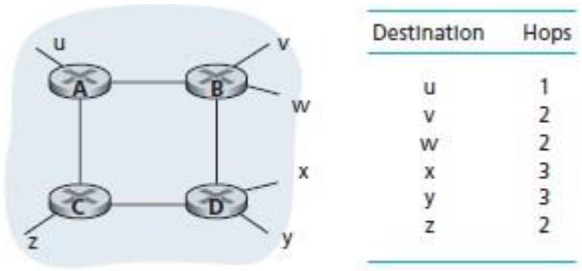
| Destination | Cost | NextHop |
|---|---|---|
| A | 1 | A |
| C | 1 | C |
| D | 2 | C |
| E | 2 | A |
| F | 2 | A |
| G | 3 | A |

**Table 3. Routing table maintained at node B.**

**RIP (Routing Information Protocol):**

Routing Information Protocol (RIP) is a dynamic protocol used to find the best route or path from end-to-end (source to destination) over a network by using a routing metric/hop count algorithm. This algorithm is used to determine the shortest path from the source to destination, which allows the data to be delivered at high speed in the shortest time. RIP plays an important role providing the shortest and best path for data to take from node to node. The hop is the step towards the next existing device, which could be a router, computer or other device. Once the length of the hop is determined, the information is stored in a routing table for future use. RIP is being used in both local and wide area networks and is generally considered to be easily configured and implemented.

Figure below illustrates an AS with six leaf subnets. The table in the figure indicates the number of hops from the source A to each of the leaf subnets.



| Destination | Hops |
|---|---|
| u | 1 |
| v | 2 |
| w | 2 |
| x | 3 |
| y | 3 |
| z | 2 |

Figure 4.34 ♦ Number of hops from source router A to various subnets

The maximum cost of a path is limited to 15, thus limiting the use of RIP to autonomous systems that are fewer than 15 hops in diameter. Recall that in DV protocols, neighboring routers exchange distance vectors with each other. The distance vector for any one router is the current estimate of the shortest path distances from that router to the subnets in the AS. In RIP, routing updates are exchanged between neighbors approximately every 30 seconds using a RIP response message. The response message sent by a router or host contains a list of up to 25 destination subnets within the AS, as well as the sender's distance to each of those subnets. Response messages are also known as RIP advertisements.

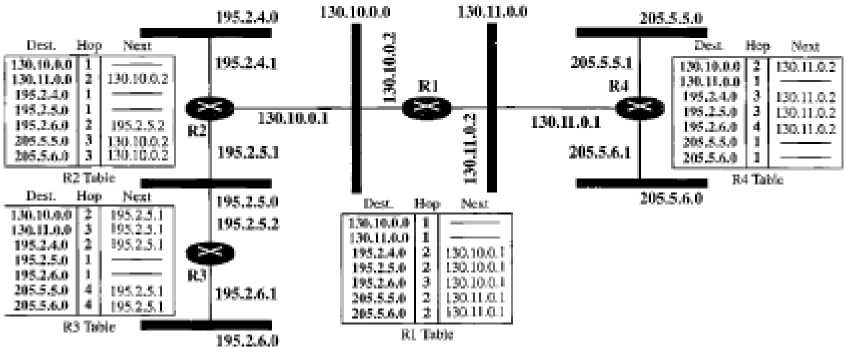
In brief the RIP protocol works as follows:

- Each router initializes its routing table with a list of locally connected networks.
- Periodically, each router advertises the entire contents of its routing table over all of its RIP-enabled interfaces.
    - ➢ Whenever a RIP router receives such an advertisement, it puts all of the appropriate routes into its routing table and begins using it to forward packets. This process ensures that every network connected to every router eventually becomes known to all routers.
    - ➢ If a router does not continue to receive advertisements for a remote route, it eventually times out that route and stops forwarding packets over it.
- Every route has a property called a metric, which indicates the "distance" to the route's destination.
    - ➢ Every time a router receives a route advertisement, it increments the metric.
    - ➢ Routers prefer shorter routes to longer routes when deciding which of two versions of a route to program in the routing table.
    - ➢ The maximum metric permitted by RIP is 16, which means that a route is unreachable. This means that the protocol cannot scale to networks where there may be more than 15 hops to a given destination.

RIP also includes some optimizations of this basic algorithm to improve stabilization of the routing database and to eliminate routing loops.

- When a router detects a change to its routing table, it sends an immediate "triggered" update. This speeds up stabilization of the routing table and elimination of routing loops.
- When a route is determined to be unreachable, RIP routers do not delete it straightaway. Instead they continue to advertise the route with a metric of 16 (unreachable). This ensures that neighbors are rapidly notified of unreachable routes, rather than having to wait for a soft state timeout.
- When router A has learnt a route from router B, it advertises the route back to B with a metric of 16 (unreachable). This ensures that B is never under the impression that A has a different way of getting to the same destination. This technique is known as "split horizon with poison reverse."
- A "Request" message allows a newly-started router to rapidly query all of its neighbors' routing tables.

**Figure 22.19** *Example of a domain using RIP*



| Dest. | Hop | Next |
|---|---|---|
| 130.10.0.0 | 1 | ——— |
| 130.11.0.0 | 2 | 130.10.0.2 |
| 195.2.4.0 | 1 | ——— |
| 195.2.5.0 | 1 | ——— |
| 195.2.6.0 | 2 | 195.2.5.2 |
| 205.5.5.0 | 3 | 130.10.0.2 |
| 205.5.6.0 | 3 | 130.10.0.2 |

R2 Table

| Dest. | Hop | Next |
|---|---|---|
| 130.10.0.0 | 2 | 195.2.5.1 |
| 130.11.0.0 | 3 | 195.2.5.1 |
| 195.2.4.0 | 2 | 195.2.5.1 |
| 195.2.5.0 | 1 | ——— |
| 195.2.6.0 | 1 | ——— |
| 205.5.5.0 | 4 | 195.2.5.1 |
| 205.5.6.0 | 4 | 195.2.5.1 |

R3 Table

| Dest. | Hop | Next |
|---|---|---|
| 130.10.0.0 | 1 | ——— |
| 130.11.0.0 | 1 | ——— |
| 195.2.4.0 | 2 | 130.10.0.1 |
| 195.2.5.0 | 2 | 130.10.0.1 |
| 195.2.6.0 | 3 | 130.10.0.1 |
| 205.5.5.0 | 2 | 130.11.0.1 |
| 205.5.6.0 | 2 | 130.11.0.1 |

R1 Table

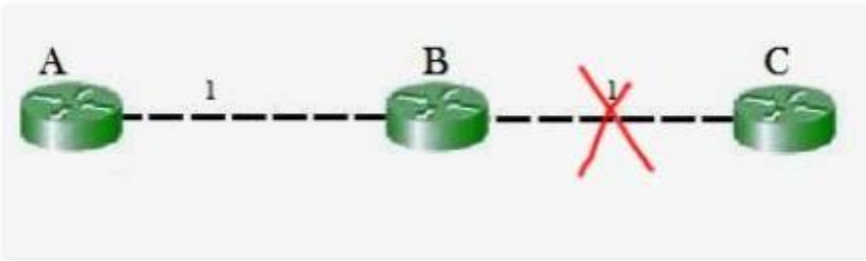
| Dest. | Hop | Next |
|---|---|---|
| 130.10.0.0 | 2 | 130.11.0.2 |
| 130.11.0.0 | 1 | ——— |
| 195.2.4.0 | 3 | 130.11.0.2 |
| 195.2.5.0 | 3 | 130.11.0.2 |
| 195.2.6.0 | 4 | 130.11.0.2 |
| 205.5.5.0 | 1 | ——— |
| 205.5.6.0 | 1 | ——— |

R4 Table

The figure above shows an autonomous system with seven networks and four routers. Table for each router is also shown. Looking at routing table for R1, it has seven entries to show how to reach each network in the autonomous system. Router R1 is directly connected to networks 130.10.0.0 and 130.11.0.0, which means that there are no next hop entries for these two networks. To send a packet to one of the three networks at the far left, router R1 needs to deliver the packet to R2. The next-node entry for these three networks is the interface of router R2 with IP address 130.10.0.1. To send a packet to the two networks at the far right, router R1 needs to send the packet to the interface of router R4 with IP address 130.11.0.1. The other tables can be explained similarly.

The main issue with Distance Vector Routing (DVR) protocols is Routing Loops, since Bellman-Ford Algorithm cannot prevent loops. This routing loop in DVR network causes Count to Infinity Problem. Routing loops usually occur when any interface goes down or two-routers send updates at the same time.

**Counting to infinity problem:**



So in this example, the Bellman-Ford algorithm will converge for each router, they will have entries for each other. B will know that it can get to C at a cost of 1, and A will know that it can get to C via B at a cost of 2. If the link between B and C is disconnected, then B will know that it can no longer get to C via that link and will remove it from its table. Before it can send any updates it's possible that it will receive an update from A which will be advertising that it can get to C at a cost of 2. B can get to A at a cost of 1, so it will update a route to C via A at a cost of 3. A will then receive updates from B later and update its cost to 4. They will then go on feeding each other bad information toward infinity which is called as Count to Infinity problem.

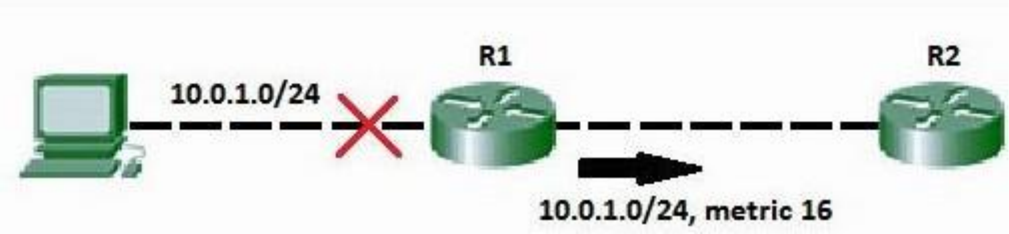
**Solution for Count to infinity:**

**Triggered Update:**

A type of Routing Information Protocol (RIP) announcement that occurs when network topology changes is called triggered update. With triggered updates, the update announcing network topology changes is sent almost immediately rather than waiting for the next periodic announcement. Triggered updates deal with count to infinity issues by forcing an update as soon as the link changes. Triggered updates improve the convergence time (the time it takes for a router to update its routing tables) of RIP internetworks, but at the cost of additional broadcast traffic while the triggered updates are propagated.

**Route Poisoning:**

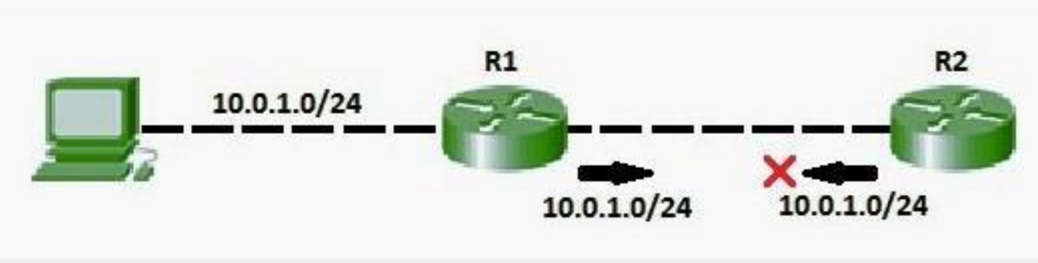
When a route fails, distance vector protocols spread the bad news about a route failure by poisoning the route. Route poisoning refers to the practice of advertising a route, but with a special metric value called Infinity. Routers consider routes advertised with an infinite metric to have failed. Each distance vector routing protocol uses the concept of an actual metric value that represents infinity. RIP defines infinity as 16. The main disadvantage of poison reverse is that it can significantly increase the size of routing announcements in certain fairly common network topologies.



**Split horizon:**

If the link between B and C goes down, and B had received a route from A , B could end up using that route via A. A would send the packet right back to B, creating a loop. But according to Split horizon Rule, Node A does not advertise its route for C (namely A to B to C) back to B. On the surface, this seems redundant since B will never route via node A because the route costs more than the direct route from B to C.

Consider the following network topology showing Split horizon:



In addition to these, we can also use split horizon with route poisoning where above both technique will be used combinely to achieve efficiency and less increase the size of routing announcements.

Split horizon with Poison reverse technique is used by Routing Information Protocol (RIP) to reduce routing loops. Additionally, **Holddown** timers can be used to avoid the formation of loops. Holddown timer immediately starts when the router is informed that attached link is down. Till this time, router ignores all updates of down route unless it receives an update from the router of that downed link. During the timer, if the down link is reachable again, routing table can be updated.

**Disadvantage:**

- The primary drawback of this algorithm is its vulnerability to the 'Count-to-Infinity' problem. Many partial solutions have been proposed but none works under all circumstances.
- Another drawback of this scheme is that it does not take into account link bandwidth.
- Yet another problem with this algorithm is that it takes longer time for convergence as network size grows.
- Increased network traffic: RIP checks with its neighboring routers every 30 seconds, which increases network traffic.
- Maximum hop count: RIP has a maximum hop count of 15, which means that on large networks, other remote routers may not be able to be reached.
- Closest may not be shortest: Choosing the closest path by hop count does not necessarily mean that the fastest route was selected. RIP does not consider other factors when calculating best path.
- RIP only updates neighbors so the updates for non-neighboring routers are not first-hand information

## <u>Link State Protocols:</u>

The basic concept of link-state routing is that every node constructs a map of the connectivity to the network, in the form of a graph, showing which nodes are connected to which other nodes. Each node then independently calculates the next best logical path from it to every possible destination in the network. Each collection of best paths will then form each node's routing table. While distance vector routers use a distributed algorithm to compute their routing tables, link-state routing uses link-state routers to exchange messages that allow each router to learn the entire network topology. Based on this learned topology, each router is then able to compute its routing table by using a shortest path computation.

Features of link state routing protocols:

- Link state packet – A small packet that contains routing information.
- Link state database – A collection information gathered from link state packet.
- Shortest path first algorithm (Dijkstra algorithm) – A calculation performed on the database results into shortest path
- Routing table – A list of known paths and interfaces.

Calculation of shortest path –

To find shortest path, each node need to run the famous **Dijkstra algorithm**. Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph. This famous algorithm uses the following steps:

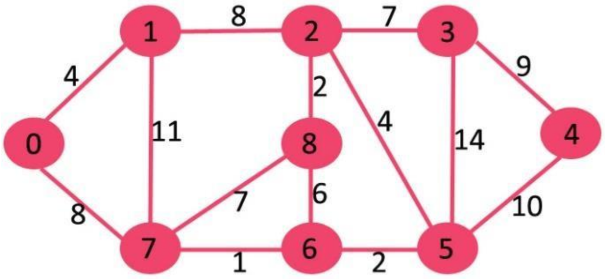
**Step-1**: The node is taken and chosen as a root node of the tree, this creates the tree with a single node, and now set the total cost of each node to some value based on the information in Link State Database

**Step-2**: Now the node selects one node, among all the nodes not in the tree like structure, which is nearest to the root, and adds this to the tree. The shape of the tree gets changed.

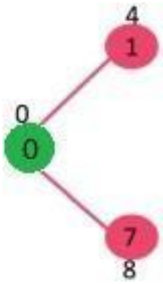
**Step-3**: After this node is added to the tree, the cost of all the nodes not in the tree needs to be updated because the paths may have been changed.

**Step-4**: The node repeats the Step 2 and Step 3 until all the nodes are added in the tree.

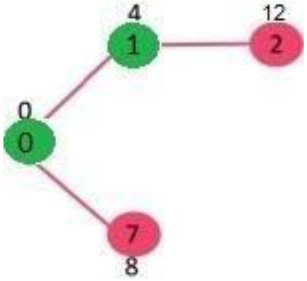
Let us understand with the following example:



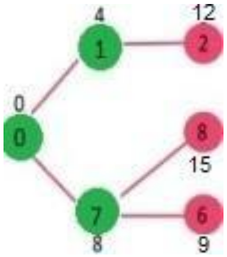
The set sptSet is initially empty and distances assigned to vertices are {0, INF, INF, INF, INF, INF, INF, INF} where INF indicates infinite. Now pick the vertex with minimum distance value. The vertex 0 is picked, include it in sptSet. So sptSet becomes {0}. After including 0 to sptSet, update distance values of its adjacent vertices. Adjacent vertices of 0 are 1 and 7. The distance values of 1 and 7 are updated as 4 and 8. Following subgraph shows vertices and their distance values, only the vertices with finite distance values are shown. The vertices included in SPT are shown in green colour.



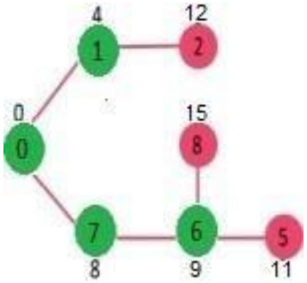
Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). The vertex 1 is picked and added to sptSet. So sptSet now becomes {0, 1}. Update the distance values of adjacent vertices of 1. The distance value of vertex 2 becomes 12.



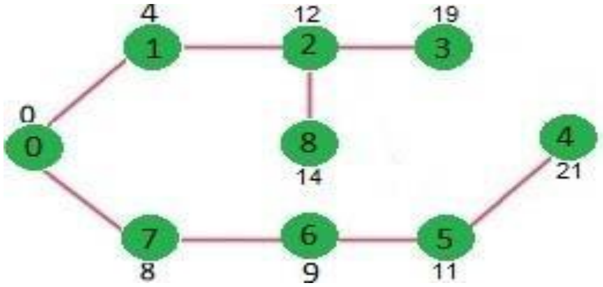
Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). Vertex 7 is picked. So sptSet now becomes {0, 1, 7}. Update the distance values of adjacent vertices of 7. The distance value of vertex 6 and 8 becomes finite (15 and 9 respectively).



Pick the vertex with minimum distance value and not already included in SPT (not in sptSET). Vertex 6 is picked. So sptSet now becomes {0, 1, 7, 6}. Update the distance values of adjacent vertices of 6. The distance value of vertex 5 and 8 are updated.

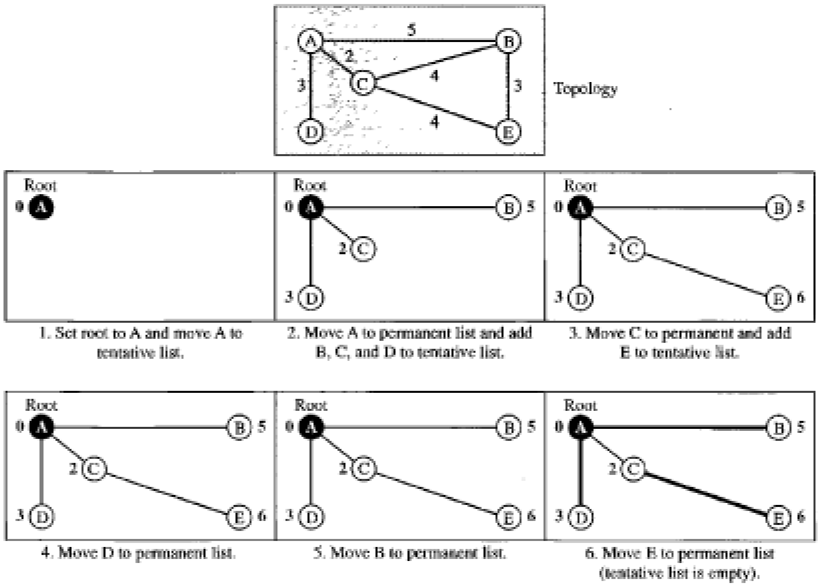


We repeat the above steps until sptSet doesn't include all vertices of given graph. Finally, we get the following Shortest Path Tree (SPT).



Another example for Dijkstra's algorithm is as follows:

Figure 22.23 *Example of formation of shortest path tree*



1. Set root to A and move A to tentative list.

2. Move A to permanent list and add B, C, and D to tentative list.

3. Move C to permanent and add E to tentative list.

4. Move D to permanent list.

5. Move B to permanent list.

6. Move E to permanent list (tentative list is empty).

**Overview of OSPF (Open Path Shortest First):**

Open Shortest Path First (OSPF) is a link state routing protocol (LSRP) that uses the Shortest Path First (SPF) network communication algorithm (Dijkstra's algorithm) to calculate the shortest connection path between known devices.

The OSPF routing protocol has largely replaced the older Routing Information Protocol (RIP) in corporate networks. Using OSPF, a router that learns of a change to a routing table (when it is reconfigured by network staff, for example) or detects a change in the network immediately multicasts the information to all other OSPF hosts in the network so they will all have the same routing table information. Unlike RIP, which requires routers to send the entire routing table to neighbors every 30 seconds, OSPF sends only the part that has changed and only when a change has taken place. When routes change -- sometimes due to equipment failure -- the time it takes OSPF routers to find a new path between endpoints with no loops (which is called "open") and that minimizes the length of the path is called the convergence time. Rather than simply counting the number of router hops between hosts on a network, as RIP does, OSPF bases its path choices on "link states" that take into account additional network information, including IT-assigned cost metrics that give some paths higher assigned costs. For example, a satellite link may be assigned higher cost than a wireless WAN link, which in turn may be assigned higher cost than a metro Ethernet link.

For example, a person in city A wants to travel to city M and is given two options:

Travel via cities B and C. The route would be ABCM. And the distance (or bandwidth cost in the networking case) for A-B is 10 miles, B-C is 5 miles and C-M is 10 miles.

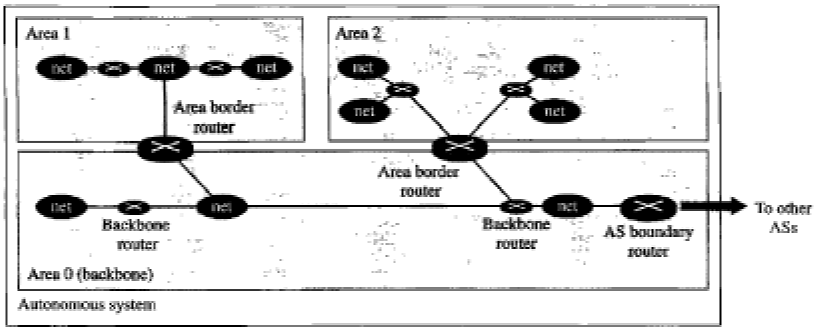
Travel via city F. The route would be AFM. And the distance for A-F is 20 miles and F-M is 10 miles.

The shortest route is always the one with least amount of distance covered in total. Thus, the ABCM route is the better option (10+5+10=25), even though the person has to travel to two cities as the associated total cost to travel to the destination is less than the second option with a single city (20+10=30). OSPF performs a similar algorithm by first calculating the shortest path between the source and destination based on link bandwidth cost and then allows the network to send and receive IP packets via the shortest route.

**OSPF Network Topology:**

Two routers communicating OSPF to each other exchange information about the routes they know about and the cost for them to get there. When many OSPF routers are part of the same network, information about all of the routes in a network are learned by all of the OSPF routers within that network—technically called an area. Each OSPF router passes along information about the routes and costs they've heard about to all of their adjacent OSPF routers, called neighbors.

**Figure 22.24** *Areas in an autonomous system*



An area is a collection of networks, hosts, and routers all contained within an autonomous system. At the border of an area, special routers called area border routers summarize the information about the area and send it to other areas. Among the areas inside an autonomous system is a special area called the backbone; all the areas inside an autonomous system must be connected to the backbone. In other words, the backbone serves as a primary area and the other areas as secondary areas. The routers inside the backbone are called the backbone routers.

OSPF works best in a hierarchical routing environment. When designing an OSPF network, the first and most important task is to determine which routers and links are to be included in the backbone (area 0) and which are to be included in each area. The following are three important characteristics to OSPF to ensure that your OSPF network has a hierarchical routing structure:

- The hierarchical routing structure must exist or be created to effectively use OSPF. The benefits of having a single area include simplicity, ease of troubleshooting, and so on.
- A contiguous backbone area must be present, and all areas must have a connection to the backbone.
- Explicit topology (shortest path) has precedence over any IP addressing schemes that might have been applied; that is, your physical topology takes precedence over a summarized route.

When designing the topology for an OSPF network, consider the following important items:

- Number of routers in an area
- Number of areas connected to an ABR (Area border router)
- Number of neighbors for a router
- Number of areas supported by a router
- Selection of the designated router (DR)
- Size and development of the OSPF LSDB (link state database)

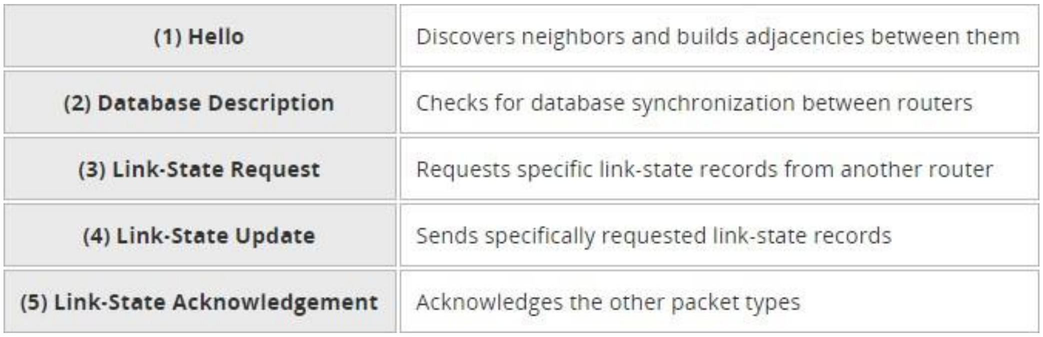
**OSPF Protocols (hello, exchange, flooding):**

Routers periodically send hello packets on all interfaces to establish and maintain neighbor relationships. Hello packets are multicast on physical networks that have a multicast or broadcast capability, which enables dynamic discovery of neighboring routers. Hello packets are sent out every 10 seconds which helps to detect failed neighbors. RouterDeadInterval (default 40 seconds) is specified for detecting such neighbors. Also, hello message ensures that link between neighbors is bidirectional. Neighboring routers agree on intervals where hello interval is set so that a link is not accidentally brought down.

OSPF uses hello packets and two timers to check if a neighbor is still alive or not:

Hello interval: this defines how often we send the hello packet.

Dead interval: this defines how long we should wait for hello packets before we declare the neighbor dead.



| | |
|---|---|
| **(1) Hello** | Discovers neighbors and builds adjacencies between them |
| **(2) Database Description** | Checks for database synchronization between routers |
| **(3) Link-State Request** | Requests specific link-state records from another router |
| **(4) Link-State Update** | Sends specifically requested link-state records |
| **(5) Link-State Acknowledgement** | Acknowledges the other packet types |

The Hello message contains a list of information needed to form an OSPF neighbor relation between two neighboring routers, the following a list of information contained the Hello messages:

- OSPF Router ID. The router's ID which is configured or automatically selected by OSPF (analyzed below)
- Hello Interval Timer. Frequency upon which Hello packets are sent.
- Dead Interval Timer. Defines how long we should wait for hello packets before we declare the neighbor dead.
- Subnet Mask
- Router Priority. Used to help determine the Designated Router (DR). Higher priority takes precedence. A configured Priority of 0 means the router will not become a DR or BDR.

- List of reachable OSPF neighbors in the network.
- Area ID
- DR & BDR’s IP addresses (if exists)
- Authentication Password (if configured)

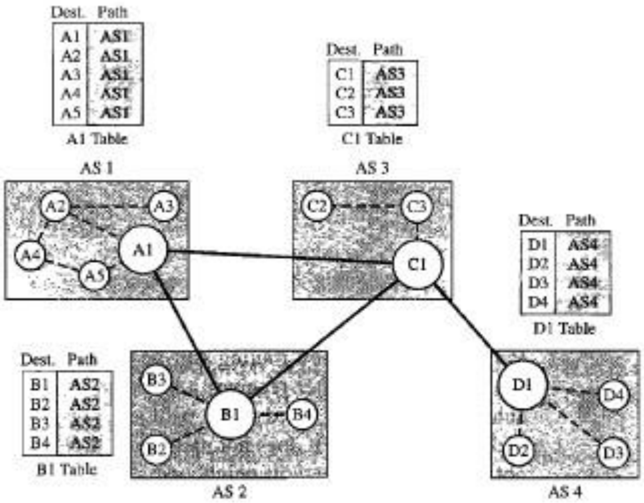
**Path Vector:**

Distance vector and Link State routing are both intradomain routing protocols. They can be used inside an autonomous system, but not between the autonomous systems. These two protocols are not suitable for interdomain routing mostly because of scalability. Both of these routing protocols become intractable when the domain of operation becomes large. Distance vector routing is subject to instability if there are more than a few hops in the domain of operation. Link state routing needs a huge amount of resources to calculate routing tables. It also creates heavy traffic because of flooding. There is a need for a third routing protocol which we call path vector routing.

Path Vector Routing is a routing algorithm in unicast routing protocol of network layer, and it is useful for interdomain routing. The principle of path vector routing is similar to that of distance vector routing. In path vector routing, we assume that there is one node (there can be more, but one is enough) in each autonomous system that acts on behalf of the entire autonomous system, referred to as speaker node. The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring autonomous systems. A speaker node advertises the path, not the metrics of the nodes, in its autonomous system or other autonomous systems.

Initialization: At the beginning, each speaker node can only know only the reachability of the nodes inside its autonomous system.

**Figure 22.30** *Initial routing tables in path vector routing*



| Dest. | Path |
|---|---|
| A1 | AS1 |
| A2 | AS1 |
| A3 | AS1 |
| A4 | AS1 |
| A5 | AS1 |

A1 Table

| Dest. | Path |
|---|---|
| C1 | AS3 |
| C2 | AS3 |
| C3 | AS3 |

C1 Table

| Dest. | Path |
|---|---|
| D1 | AS4 |
| D2 | AS4 |
| D3 | AS4 |
| D4 | AS4 |

D1 Table

| Dest. | Path |
|---|---|
| B1 | AS2 |
| B2 | AS2 |
| B3 | AS2 |
| B4 | AS2 |

B1 Table

Sharing: Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors. In figure, node A1 shares its table with nodes B1 and C1.

Node C1 shares its table with nodes D1, B1 and A1. Node B1 shares its table with C1 and A1. Node D1 shares its table with C1.

Updating: When a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table. After a while, each speaker has a table and knows how to reach each node in other autonomous systems. Figure below shows the tables for each speaker node after the system is stabilized.

Loop prevention: The instability of distance vector routing and the creation of loops can be avoided in path vector routing. When a router receives a message, it checks to see if its autonomous system is in the path list to the destination. If it is, looping is involved and the message is ignored.

Policy routing: Policy routing can be easily implemented through path vector routing. When a router receives a message, it can check the path. If one of the autonomous systems listed in the path is against its policy, it can ignore that path and that destination. It does not update its routing table with this path, and it does not send this message to its neighbors.

Optimum path: The optimum path in path vector routing is a path to a destination that is the best for the organization that runs the autonomous system. We cannot use metrics in this route because each autonomous system that is included in the path may use a different criterion for the metric. One system may use RIP which defines hop count as the metric, another may use OSPF with the minimum delay (higher link bandwidth) as the metric. The optimum path is the path that fits the organization. In previous figure, each autonomous system may have more than one path to a destination. For eg: a path from AS4 to AS1 can be AS4-AS3-AS2-AS1 or it can be AS4-AS3-AS1. For the tables, we choose the one that had the smaller number of autonomous systems, but this is not always the case. Other criteria, such as security, safety, and reliability can also be applied.

**Border Gateway Protocol:**

Border gateway protocol (BGP) is an interdomain routing protocol using path vector routing. BGP is protocol that manages how packets are routed across the internet through the exchange of routing and reachability information between edge routers. BGP directs packets between autonomous systems (AS) -- networks managed by a single enterprise or service provider. Traffic that is routed within a single network AS is referred to as internal BGP, or iBGP. More often, BGP is used to connect one AS to other autonomous systems, and it is then referred to as an external BGP, or eBGP.

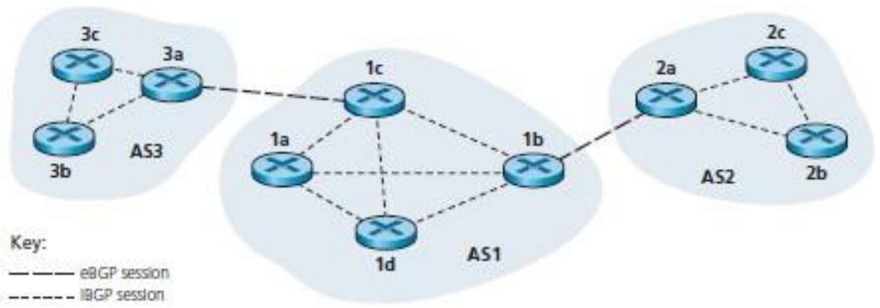


**Figure 4.40** ♦ eBGP and iBGP sessions

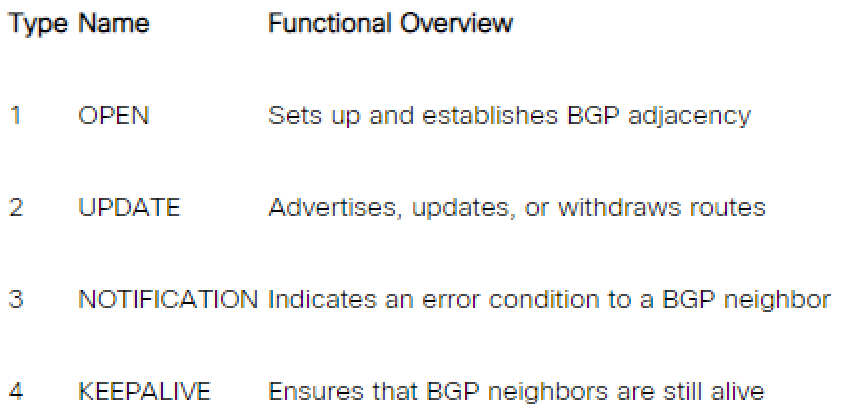
All other routing protocols are concerned solely with finding the optimal path towards all known destinations. BGP cannot take this simplistic approach because the peering agreements between ISPs almost always result in complex routing policies. To help network operators implement these policies, BGP carries a large number of attributes with each IP prefix: **(BGP Path Attributes)**

- Weight – The BGP weight attribute is Cisco-specific and is used to influence how traffic is routed for a specific BGP device. This value does not pass between internal or external BGP neighbors (peers).
- Local Preference – The local preference attribute is used to dictate how traffic prefers to leave a specific BGP ASN. This attribute is passed between neighbors within the same ASN. The highest local preference gets priority.
- Local Routes – Routes which have been sourced from the local router will be preferred over those sourced from other routers.
- Shortest AS_PATH – With BGP, the path is notated by the ASN of the external BGP networks that must be traversed to reach the destination network; e.g., 10 20 30 means that the traffic must pass through ASNs 10, 20, and 30 to reach the destination. If multiple options exist to a specific network, the one with the shortest AS path will be preferred.
- Origin – With origin, BGP is looking for the source of the initial network advertisement, for example if it was redistributed from an IGP, an EGP or through an unknown source. When analyzing this attribute, routes that have originated from an IGP are preferred to those from an EGP, and routes that have originated by an EGP will be preferred over those originated from an unknown source. I < E <?
- Multi-Exit Discriminator (MED) – The MED is a value that can be injected into a neighboring BGP ASN. This is used when multiple paths exist between two different BGP ASNs. The MED is used to suggest to the neighboring ASN the preferred way to route traffic into their network. The lowest MED value gets priority.
- BGP Neighbor Type – There are two different types of BGP neighborship: internal and external. A BGP neighborship that exists within the same ASN between two devices is considered internal, and a BGP neighborship that exists between devices from different ASNs is considered external. External (or eBGP) routes are preferred to Internal (iBGP) routes.
- IGP metric/next hop – The next attribute uses the IGP metric to the BGP next hop address.
- Oldest External Route – If the contending BGP routes are external then the one which has existed the longest will be preferred
- Lowest Router-ID – The route with the lowest BGP router ID will be preferred
- Lowest Neighbor Address – The route coming through a neighbor with the lowest address will be preferred.

**BGP Message (Packet) Types:**

BGP communication uses four message types: Open, Update, Keep Alive, Notification.

There is a 5th message type defined in BGP called **Route-Refresh** to support the route refresh capability. 'Route Refresh Capability', which would allow the dynamic exchange of route refresh request between BGP speakers and subsequent re-advertisement. One possible application of this capability is to facilitate non-disruptive routing policy changes.



| Type | Name | Functional Overview |
|---|---|---|
| 1 | OPEN | Sets up and establishes BGP adjacency |
| 2 | UPDATE | Advertises, updates, or withdraws routes |
| 3 | NOTIFICATION | Indicates an error condition to a BGP neighbor |
| 4 | KEEPALIVE | Ensures that BGP neighbors are still alive |

Open Message:

Once two BGP routers have completed a TCP 3-way handshake they will attempt to establish a BGP session, this is done using open messages. In the open message we will find some information about the BGP router, these have to be negotiated and accepted by both routers before we can exchange any routing information.

Update Message:

Once two routers have become BGP neighbors, they can start exchanging routing information. This is done with the update message. In the update message you will find information about the prefixes that are advertised.

Notification Message:

A Notification message is sent when an error is detected with the BGP session, such as a hold timer expiring, neighbor capabilities change, or a BGP session reset is requested. This causes the BGP connection to close.

Keep Alive Message:

BGP does not rely on the TCP connection state to ensure that the neighbors are still alive. Keepalive messages are exchanged every one-third of the Hold Timer agreed upon between the two BGP routers. Cisco devices have a default Hold Time of 180 seconds, so the default Keepalive interval is 60 seconds. If the Hold Time is set for zero, no Keepalive messages are sent between the BGP neighbors.

**Characteristics of Border Gateway Protocol (BGP):**

- Inter-Autonomous System Configuration: The main role of BGP is to provide communication between two autonomous systems.
- BGP supports Next-Hop Paradigm.
- Coordination among multiple BGP speakers within the AS (Autonomous System).
- Path Information: BGP advertisement also include path information, along with the reachable destination and next destination pair.

- Policy Support: BGP can implement policies that can be configured by the administrator. For ex:- a router running BGP can be configured to distinguish between the routes that are known within the AS and that which are known from outside the AS.
- Runs Over TCP.
- BGP conserve network Bandwidth.
- BGP supports CIDR.
- BGP also supports Security.

**Functionality of Border Gateway Protocol (BGP):**

BGP peers performs 3 functions, which are given below.

- The first function consists of initial peer acquisition and authentication. both the peers established a TCP connection and perform message exchange that guarantees both sides have agreed to communicate.
- The second function mainly focus on sending of negative or positive reach-ability information.
- The third function verifies that the peers and the network connection between them are functioning correctly.

**BGP Route Information Management Functions:**

- Route Storage: Each BGP stores information about how to reach other networks.
- Route Update: In this task, Special techniques are used to determine when and how to use the information received from peers to properly update the routes.
- Route Selection: Each BGP uses the information in its route databases to select good routes to each network on the internet network.
- Route advertisement: Each BGP speaker regularly tells its peer what is knows about various networks and methods to reach them.

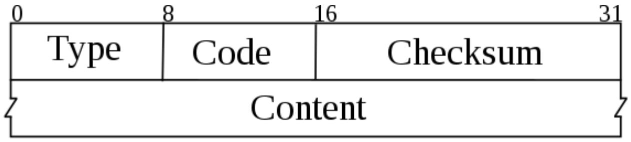
**Internet Control Message Protocol (ICMP):**

- ICMP is a TCP/IP network layer protocol that provides troubleshooting, control and error message services.
- Internet Control Message Protocol is also known as RFC 792.
- While ICMP is not used regularly in end-user applications, it is used by network administrators to troubleshoot Internet connections in diagnostic utilities.
- An ICMP message is created as a result of errors in an IP datagram. These errors are reported to the originating datagram's source IP address.
- An ICMP message is encapsulated directly within a single IP datagram and reports errors in the processing of datagrams.
- ICMP messages are transmitted as datagrams and consist of an IP header that encapsulates the ICMP data.
- ICMP packets are IP packets with ICMP in the IP data portion. ICMP messages also contain the entire IP header from the original message, so the end system knows which packet failed.

There can be several reasons behind reporting the error like:

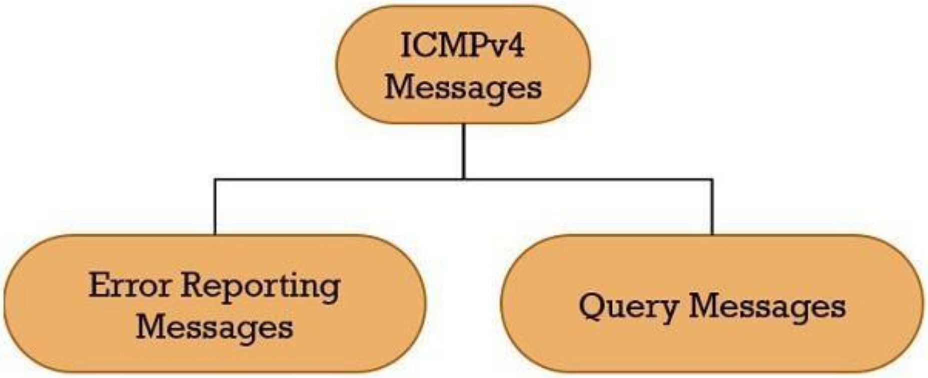
- A router with a datagram for a host in another network, may not find the next hop (router) to the final destination host.
- Datagram's time-to-live field has become zero.
- There may be ambiguity in the header of IP datagram.
- It may happen that all the fragments of datagram if do not arrive within a time limit to the destination host.

ICMP Header Format:



ICMP is available for both IPv4 and IPv6. The header format is similar for both versions of ICMP. ICMPv4 is the messaging protocol for IPv4. ICMPv6 provides these same services for IPv6 but includes additional functionality.

ICMPv4: ICMP for IPv4



Error Reporting Messages: (Report the error)

- Destination unreachable
- Source Quench
- Time Exceeded
- Parameter Problem (header field parameters corrupted)
- Redirection (when packet being routed wrongly, informed by intermediate router)

Query Messages: (identify network problems)

- Echo Request and Reply
- Timestamp Request and Reply

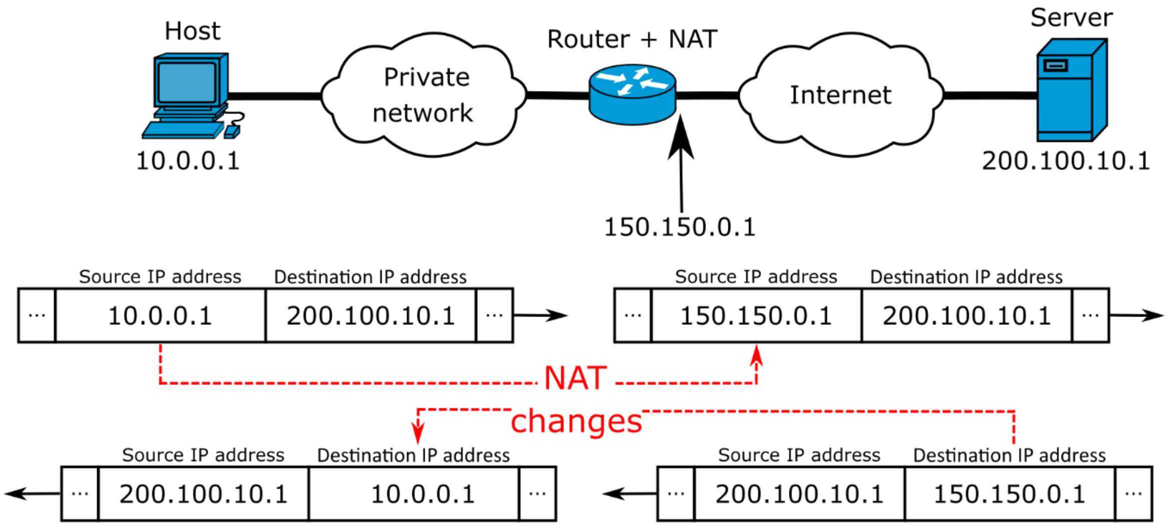
ICMPv6:

- Internet Control Message Protocol version 6 (ICMPv6) is the implementation of the Internet Control Message Protocol (ICMP) for Internet Protocol version 6 (IPv6).
- ICMPv6 is defined in RFC 4443.
- ICMPv6 plays a far more important role in the operation of IPv6 than ICMPv4 does for IPv4.
- ICMPv6 is an integral part of IPv6 and performs error reporting and diagnostic functions (e.g., ping).
- ICMPv6 has a framework for extensions to implement future changes. Several extensions have been published, defining new ICMPv6 message types as well as new options for existing ICMPv6 message types.
- For example, Neighbor Discovery Protocol (NDP) is a node discovery protocol based on ICMPv6 which replaces and enhances functions of ARP.
- Secure Neighbor Discovery (SEND) is an extension of NDP with extra security.
- Multicast Listener Discovery (MLD) is used by IPv6 routers for discovering multicast listeners on a directly attached link, much like Internet Group Management Protocol (IGMP) is used in IPv4.
- Multicast Router Discovery (MRD) allows the discovery of multicast routers.

**Network Address Translation (NAT):**

NAT is a technique to map multiple local private addresses to a public one before transferring the information. Organizations that want multiple devices to employ a single IP address use NAT, as do most home routers.

The most common form of network translation involves a large private network using addresses in a private range (10.0.0.0 to 10.255.255.255, 172.16.0.0 to 172.31.255.255, or 192.168.0 0 to 192.168.255.255).



NAT Types:

There are three different types of NATs.

1. Static NAT

When the local address is converted to a public one, this NAT chooses the same one. This means there will be a consistent public IP address associated with that router or NAT device.

2. Dynamic NAT

Instead of choosing the same IP address every time, this NAT goes through a pool of public IP addresses. This results in the router or NAT device getting a different address each time the router translates the local address to a public address.

3. PAT

PAT stands for port address translation. It's a type of dynamic NAT, but it bands several local IP addresses to a singular public one. Organizations that want all their employees' activity to use a singular IP address use a PAT, often under the supervision of a network administrator.

Why use NAT?

*IP Conservation:* IP addresses identify each device connected to the internet. The existing IP version 4 (IPv4) uses 32-bit numbered IP addresses, which allows for 4 billion possible IP addresses, which seemed like more than enough when it launched in the 1970s.

However, the internet has exploded, and while not all 7 billion people on the planet access the internet regularly, those that do often have multiple connected devices: phones, personal desktop, work laptop, tablet, TV, even refrigerators.

Therefore, the number of devices accessing the internet far surpasses the number of IP addresses available. Routing all of these devices via one connection using NAT helps to consolidate multiple private IP addresses into one public IP address. This helps to keep more public IP addresses available even while private IP addresses proliferate.

**Network Traffic Analysis:**

Network traffic analysis (NTA) is a method of monitoring network availability and activity to identify anomalies, including security and operational issues.

In other words, Network traffic analysis (NTA) is the process of intercepting, recording and analyzing network traffic communication patterns in order to detect and respond to security threats.

Network traffic analysis is primarily done to get in-depth insight into what type of traffic/network packets or data is flowing through a network.

Typically, network traffic analysis is done through a network monitoring or network bandwidth monitoring software/application. The traffic statistics from network traffic analysis helps in:

- Understanding and evaluating the network utilization
- Download/upload speeds
- Type, size, origin and destination and content/data of packets
- Collecting a real-time and historical record of what's happening on your network
- Detecting malware activity
- Detecting the use of vulnerable protocols and ciphers
- Troubleshooting a slow network
- Improving internal visibility and eliminating blind spots

Network security staff uses network traffic analysis to identify any malicious or suspicious packets within the traffic. Similarly, network administrations seek to monitor download/upload speeds, throughput, content, etc. to understand network operations.

Network traffic analysis is also used by attackers/intruders to analyze network traffic patterns and identify any vulnerabilities or means to break in or retrieve sensitive data.

## Security Concepts: Firewall & Router Access Control

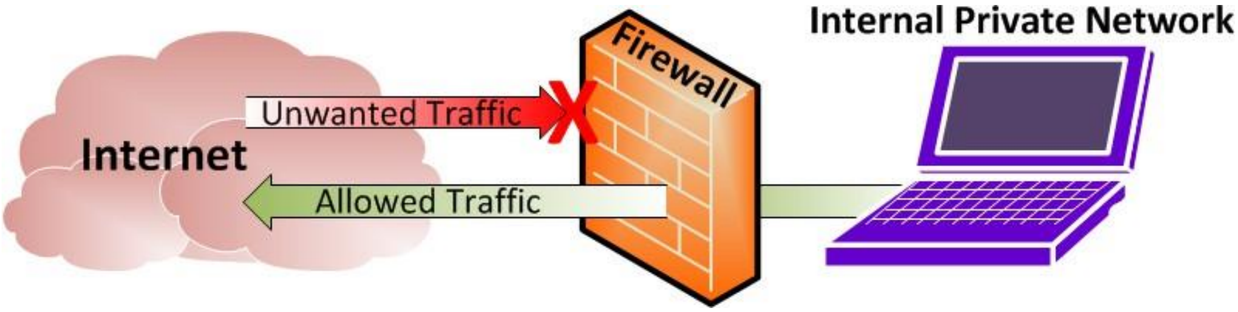
Network security consists of the policies, processes and practices adopted to prevent, detect and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources.

**Firewall:**

A firewall is a software or a hardware device that examines the data from several networks and then either permits it or blocks it to communicate with your network and this process is governed by a set of predefined security guidelines.

In other words, a firewall is a hardware device or software application installed on the borderline of secured networks to examine and control incoming and outgoing network communications.

A firewall typically establishes a barrier between a trusted internal network and untrusted external network, such as the internet.



Hardware firewalls can be purchased as a stand-alone product but are also typically found in routers, and should be considered as an important part of system and network set-up.

Software firewalls are installed on your computer (like any software) and we can customize it; allowing us some control over its function and protection features.

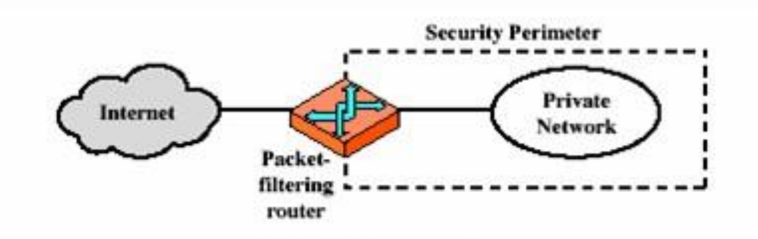
Types of Firewall:

There are four basic types of firewalls:

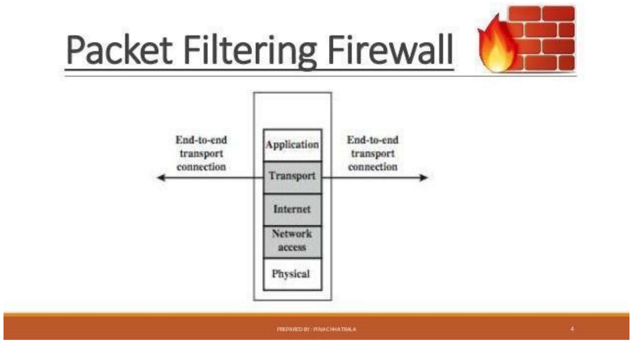
- Packet Filtering Firewall
- Stateful Inspection Firewall
- Circuit-Level Gateway
- Application-Level Gateway

Packet Filtering Firewall:

- A packet filtering firewall applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet.
- The firewall is typically configured to filter packets going in both directions (from and to the internal network).
- Filtering rules are based on information contained in a network packet.
- Two default policies are possible:
  - Default = discard: That which is not expressly permitted is prohibited.
  - Default = forward: That which is not expressly prohibited is permitted.
- Advantage of a packet filtering firewall is its simplicity. Also, packet filters typically are transparent to users and are very fast.



(a) Packet-filtering router



Packet Filtering Firewall

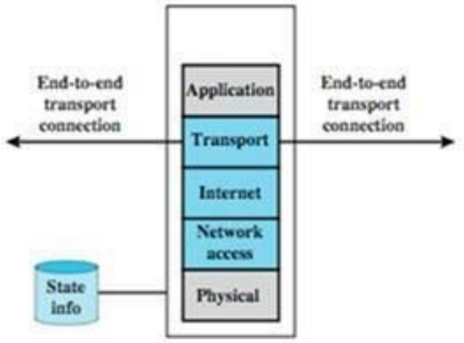
End-to-end transport connection
Application
Transport
Internet
Network access
Physical
End-to-end transport connection

PREPARED BY : PINA CHHATRALA

4

State-full Inspection Firewall:

- State-full packet filtering is a firewall technology that monitors the state of active connections and uses this information to determine which network packets to allow through the firewall.
- It is also known as dynamic packet filtering and Stateful inspection filtering.
- Only packets matching a known active connection are allowed to pass the firewall.
- It is a security feature often included in business networks.

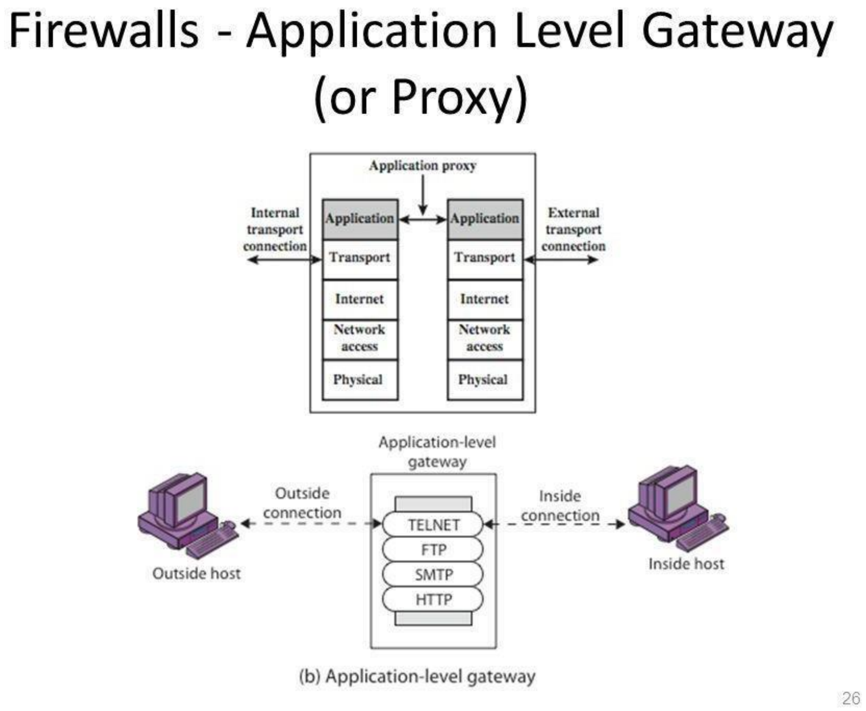

End-to-end transport connection
Application
End-to-end transport connection
Transport
Internet
Network access
State info
Physical

(c) Stateful inspection firewall

- A stateful inspection packet firewall tightens up the rules for TCP traffic by creating a directory of outbound TCP connections.
- There is an entry for each currently established connection.
- The packet filter will now allow incoming traffic to high-numbered ports only for those packets that fit the profile of one of the entries in this directory.
- By recording session information such as IP addresses and port numbers, a dynamic packet filter can implement a much tighter security posture than a static packet filter can.

Application-Level Gateway:

- An application-level gateway, also called an application proxy, acts as a relay of application-level traffic.
- The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed.
- When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints.
- If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall.
- Application proxy filters incoming node traffic to certain specifications which mean that only transmitted network application data is filtered.
- Application-level gateways tend to be more secure than packet filters.
- In addition, it is easy to log and audit all incoming traffic at the application level.
- Rather than trying to deal with the numerous possible combinations that are to be allowed and forbidden at the TCP and IP level, the application-level gateway need only examine a few allowable applications.

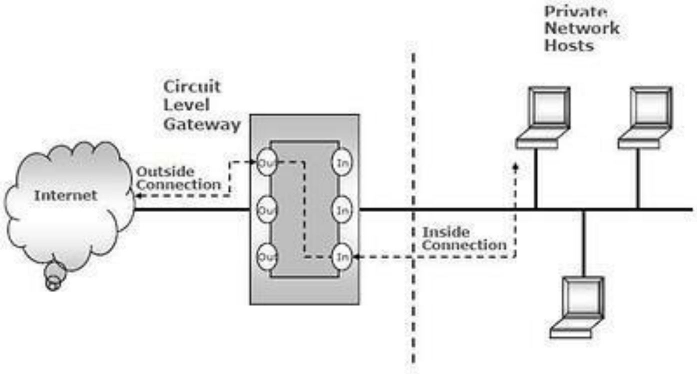


# Firewalls - Application Level Gateway (or Proxy)

(b) Application-level gateway

Circuit Level Gateway:

- A fourth type of firewall is the circuit-level gateway or circuit-level proxy.
- The circuit level gateway firewalls work at the transport and session layer of the OSI model. They monitor TCP handshaking between the packets to determine if a requested session is legitimate.
- As with an application gateway, a circuit-level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host.
- Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents.
- The security function consists of determining which connections will be allowed.
- A typical use of circuit-level gateways is a situation in which the system administrator trusts the internal users.



**Router Access Control (ACL):**

- ACLs are a network filter utilized by routers and some switches to permit and restrict data flows into and out of network interfaces.
- When an ACL is configured on an interface, the network device analyzes data passing through the interface, compares it to the criteria described in the ACL, and either permits the data to flow or prohibits it.
- There are a variety of reasons we use ACLs. The primary reason is to provide a basic level of security for the network.
- ACLs are not as complex and in depth of protection as firewalls, but they do provide protection on higher speed interfaces where line rate speed is important and firewalls may be restrictive. Also, ACLs do offer a significant amount of firewall capability.
- ACLs are also used to restrict updates for routing from network peers and can be instrumental in defining flow control for network traffic.
- ACLs should be placed on external routers to filter traffic against less desirable networks and known vulnerable protocols.
- One of the most common methods in this case is to setup a DMZ, or de-militarized buffer zone in your network.
- This architecture is normally implemented with two separate network devices:

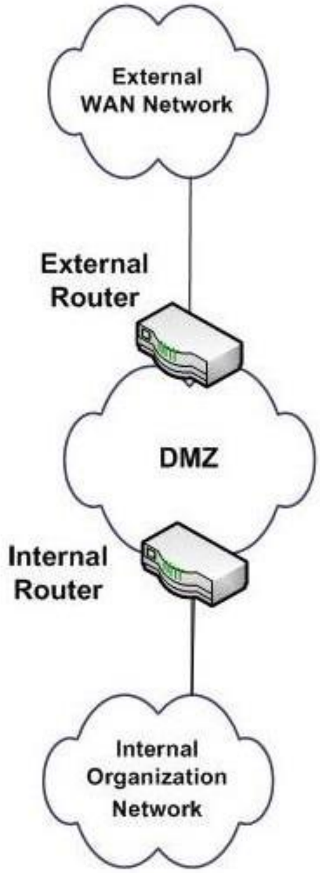


Fig: External Network Configuration with DMZ