Chapter 4

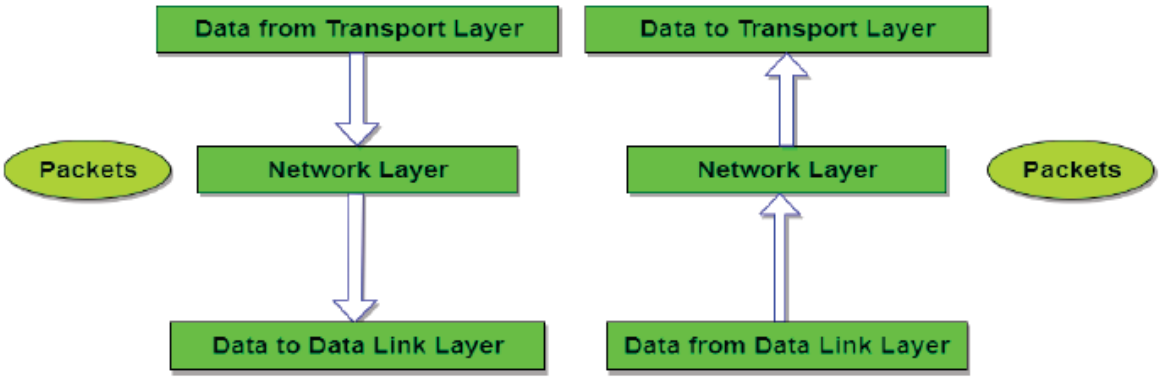
Network Layer

1. Introduction

The network layer is the third layer of the OSI model. This layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links).

It provides services to the transport layer and receives services from the data- link layer.

The network layer translates the logical addresses into physical addresses. It determines the route from the source to the destination and also manages the traffic problems such as switching, routing and controls the congestion of data packets.



2. Important functions of the Network Layer

a. Logical Addressing

- Assigns a unique IP address to each device on a network.
- IP addresses are logical and can change based on the network configuration.

b. Routing

- Determines the optimal path for data to travel from the sender to the receiver.
- Utilizes algorithms like **distance vector**, **link state**, and **path vector**.

c. Packet Forwarding

- Transfers packets from one router to another until it reaches its destination.

d. Fragmentation and Reassembly

- If a packet is too large for a network's maximum transmission unit (MTU), it is fragmented and reassembled at the destination.

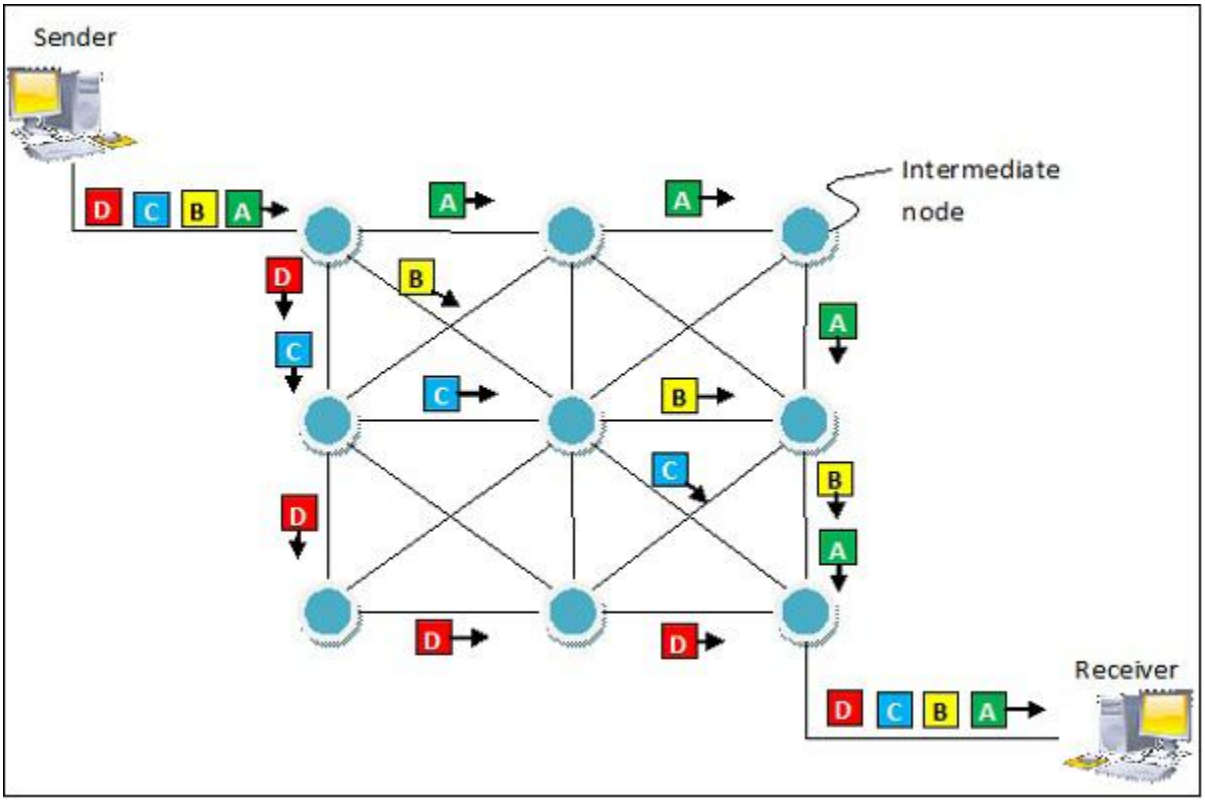
3. Packet Switching

Packet switching is a digital network transmission process in which data is broken into suitably-sized pieces or blocks for fast and efficient transfer via different network devices.

When a computer attempts to send a file to another computer, the file is broken into packets so that it can be sent across the network in the most efficient way. These packets are then routed by network devices to the destination.

3.1. Process

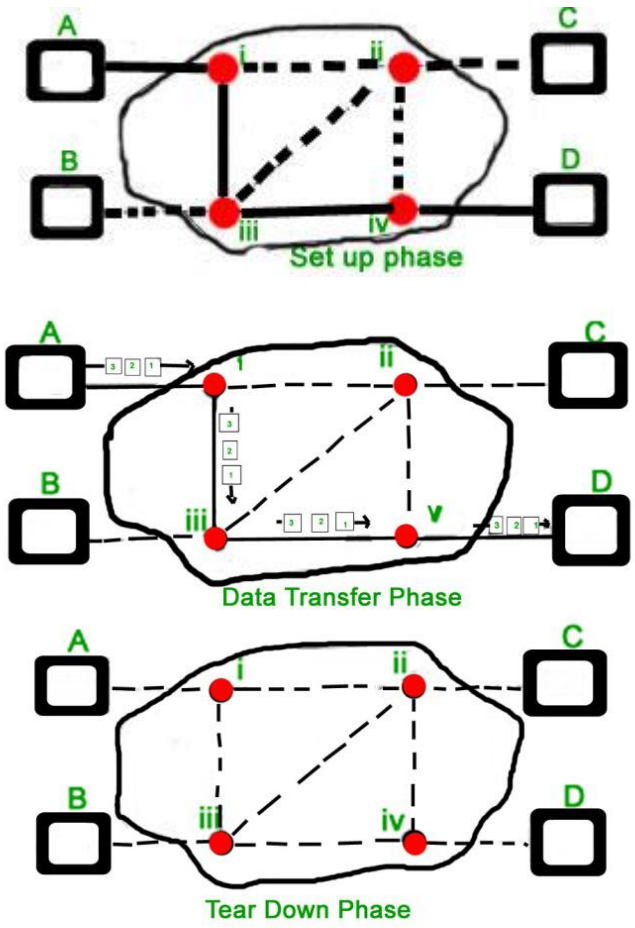
- Each packet in a packet switching technique has two parts: a header and a payload. The header contains the addressing information of the packet and is used by the intermediate routers to direct it towards its destination. The payload carries the actual data.
- A packet is transmitted as soon as it is available in a node, based upon its header information. The packets of a message are not routed via the same path. So, the packets in the message arrive in the destination out of order. It is the responsibility of the destination to reorder the packets in order to retrieve the original message.
- The process is diagrammatically represented in the following figure. Here the message comprises of four packets, A, B, C and D, which may follow different routes from the sender to the receiver.



3.2. Modes of Packet Switching

3.2.1. Connection-oriented Packet Switching (Virtual Circuit)

Before starting the transmission, it establishes a logical path or virtual connection using signaling protocol, between sender and receiver and all packets belongs to this flow will follow this predefined route. Virtual Circuit ID is provided by switches/routers to uniquely identify this virtual connection. Data is divided into small units and all these small units are appended with help of sequence number. Overall, three phases takes place here- Setup, data transfer and tear down phase.



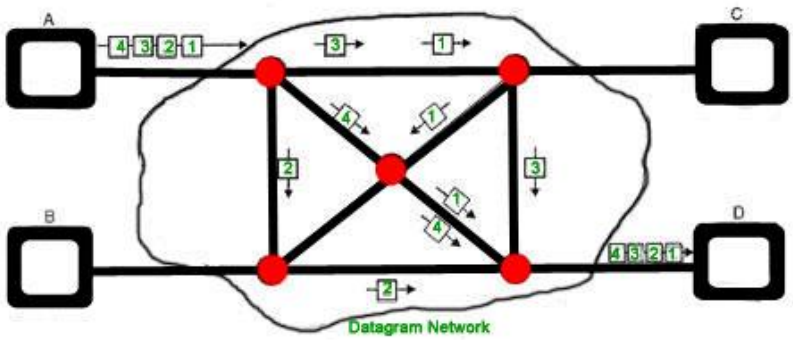
Phases in virtual circuit packet switching

All address information is only transferred during setup phase. Once the route to destination is discovered, entry is added to switching table of each intermediate node. During data transfer, packet header (local header) may contain information such as length, timestamp, sequence number etc.

Connection-oriented switching is very useful in switched WAN. Some popular protocols which use Virtual Circuit Switching approach are X.25, Frame-Relay, ATM and MPLS (Multi- Protocol Label Switching).

3.2.2. Connectionless Packet Switching (Datagram)

Unlike Connection-oriented packet switching, in Connectionless Packet Switching each packet contains all necessary addressing information such as source address, destination address and port numbers, etc. In Datagram Packet Switching, each packet is treated independently. Packets belonging to one flow may take different routes because routing decisions are made dynamically, so the packets arrived at destination might be out of order. It has no connection setup and teardown phase, like Virtual Circuits.



Datagram Packet Switching

Packet delivery is not guaranteed in connectionless packet switching, so the reliable delivery must be provided by end systems using additional protocols.

4. IPv4 addresses

4.1. Logical addressing

- Each machine has a **MAC address** which is also called the **physical address**. This is the address of the network interface card of the machine.
- But if we know the MAC address of a machine, how do we know where the machine is? It is impossible to locate a machine among millions of machines based on its MAC address.
- Hence, we need another addressing scheme which is global and depends on the network to which the machine is connected.
- This is required so that we can identify the network to which the destination machine belongs to. This global address is called **logical address (IP address).**
- The TCP/IP protocol model uses IP addresses to **uniquely** identify a router and machine in the internet.
- No two machines on the internet can have the same IP address.
- There are two versions of IP:
    1. IPv4
    2. IPv6

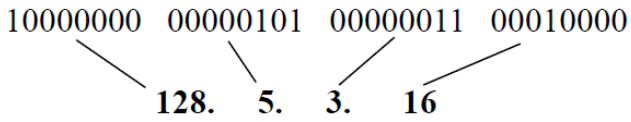
4.2. IPv4 Addresses

The Internet protocol version 4 (IPv4) is the most important protocol in the network layer of the TCP/IP model.

- **Address Space:** IPv4 uses **32 bit** addresses. Hence, the IPv4 address space is 2^{32} **i.e., 4,294,976,296** addresses in all.
- **Notations:** An IPv4 address can be represented using two notations:
  1. Binary notation
  2. Dotted-decimal notation
- **Binary notation:** In this notation, the address is represented as 32 bits. Thus it is a 4-byte address.
- **Example:** 10010101 00000010 00011101 01110101
- **Dotted-decimal notation:**
  - It is usually difficult to read and write IP addresses in binary. To make it easier, the dotted-decimal format is used.
  - In this case, each byte (8 bits) are converted to decimal and separated by a **dot (.).** The general form is A.B.C.D

    where A, B, C and D are positive integers in the **range 0 to 255.**

    Example: 128.5.3.16

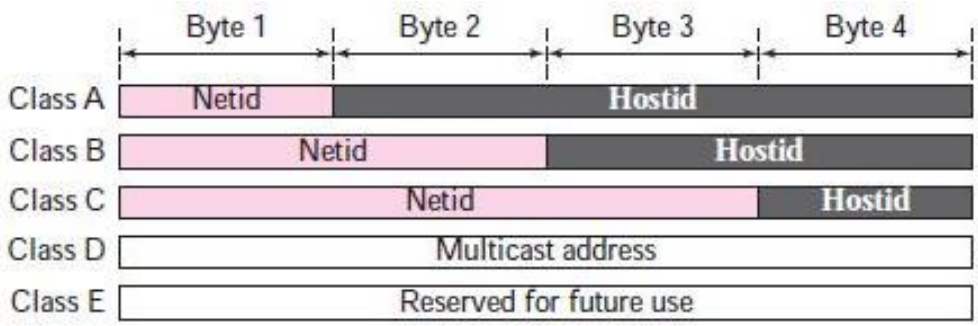


Binary and dotted-decimal format

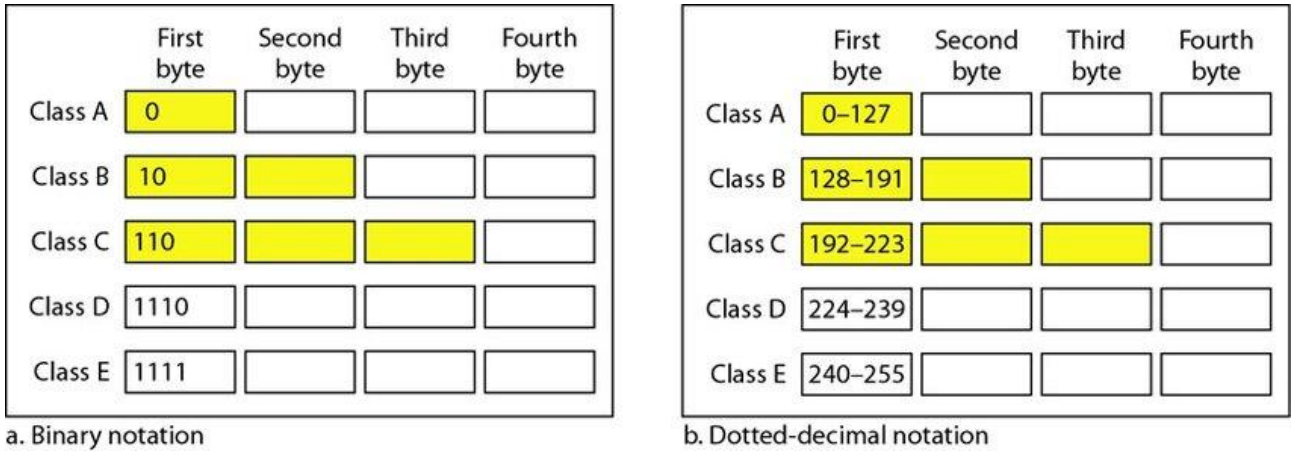
- **Categories of IPv4 addressing:** There are two broad categories of IPv4 Addressing techniques. They are:
  - Classful Addressing
  - Classless Addressing

4.3. Classful Addressing

- The entire IP address space of 2^{32} addresses is divided into **five "classes",** A, B, C, D and E.
- Each class is allotted a portion of the address space.
- The addresses in **classes A, B and C** are divided into two parts: **netid and hostid**.
- The **length** of the netid and hostid **vary** according to the class. The following diagram shows the 5 address classes along with the Netid and Hostid.



- The class of an IP address can be identified from the value of the first byte. The range of values are given below. The shaded part is the netid.



Identifying IP address classes

- Class A, B and C addresses are used in internetworks. Class D and E are reserved. Class A networks are the largest networks with maximum number of hosts while class C networks are the smallest networks with 256 hosts. Thus, if an organization has 100 machines, it should take a class C address.

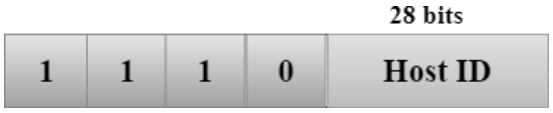
Class	Netid	Hostid	First byte range	No. of Networks	No. of Hosts
Class A	1 byte	3 bytes	0-127	$2^7 = 128$	$2^{24} = 16,777,216$
Class B	2 bytes	2 bytes	128-191	$2^{14} = 16,384$	$2^{16} = 65,536$
Class C	3 bytes	1 byte	192-223	$2^{21} = 2,097,152$	$2^8 = 256$
Class D	-	-	224-239	Not divided further. Used for multicasting	
Class E	-	-	240-255	Not divided further. Used as reserved addresses	

- **Class A:**
  - In Class A, an IP address is assigned to those networks that contain a large number of hosts.
  - The network ID is 8 bits long.
  - The host ID is 24 bits long.
  - In Class A, the first bit in higher order bits of the first octet is always set to 0 and the remaining 7 bits determine the network ID.
  - The 24 bits determine the host ID in any network.
  - The total number of networks in Class A = 2^7 = 128 network address
  - The total number of hosts in Class A = 2^{24} - 2 = 16,777,214 host address

- **Class B:**
  - In Class B, an IP address is assigned to those networks that range from small- sized to large-sized networks.
  - The Network ID is 16 bits long.
  - The Host ID is 16 bits long.
  - In Class B, the higher order bits of the first octet is always set to 10, and the remaining14 bits determine the network ID.
  - The other 16 bits determine the Host ID.
  - The total number of networks in Class B = 2^{14} = 16384 network address
  - The total number of hosts in Class B = 2^{16} - 2 = 65534 host address

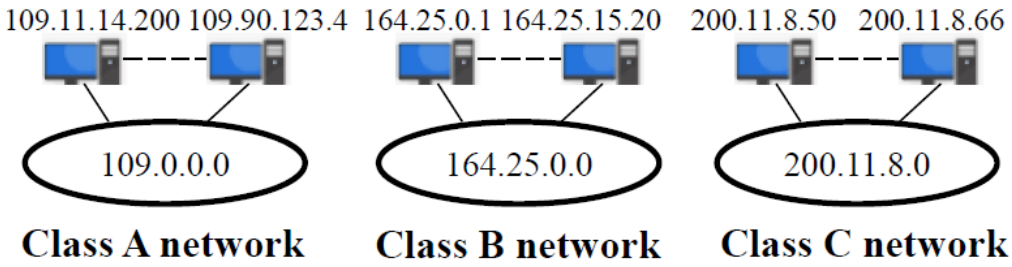
- **Class C:**
  - In Class C, an IP address is assigned to only small-sized networks.
  - The Network ID is 24 bits long.
  - The host ID is 8 bits long.
  - In Class C, the higher order bits of the first octet is always set to 110, and the remaining 21 bits determine the network ID.
  - The 8 bits of the host ID determine the host in a network.
  - The total number of networks = 2^{21} = 2097152 network address
  - The total number of hosts = 2^8 - 2 = 254 host address

- **Class D:**
  - In Class D, an IP address is reserved for multicast addresses.
  - It does not possess subnetting.
  - The higher order bits of the first octet is always set to 1110, and the remaining bits determines the host ID in any network.



- **Class E:**
  - In Class E, an IP address is used for the future use or for the research and development purposes.
  - It does not possess any subnetting.
  - The higher order bits of the first octet is always set to 1111, and the remaining bits determines the host ID in any network.

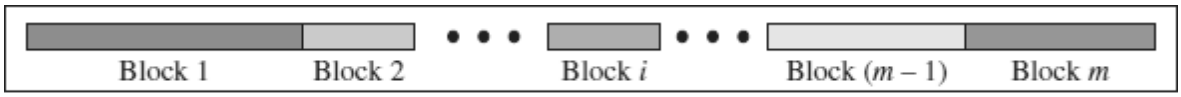
- **Network Address:** A network address is the IP address which identifies a network among multiple networks. It has both net-id and host-id, with all host-id bits zero.
  - **Example:** 109.0.0.0, 164.25.0.0 and 200.11.8.0 are network addresses which identify a specific network.



Network Addresses

4.4. Classless Addressing

- The problem with classful addressing is that it is a fixed scheme.
- Due to the fixed number of address ranges, most of the addresses are wasted. For example, an organization having 20 hosts only will have to take a class C address. This will waste 236 addresses.
- Hence, the classful scheme is now replaced with a classless addressing scheme.
- Here the addresses are assigned in blocks but there are no classes. An organization can take only the required number of addresses.
- In classless addressing, variable-length blocks are used that belong to no classes.
- We can have a block of 1 address, 2 addresses, 4 addresses, 128 addresses, and so on.
- In classless addressing, the whole address space is divided into variable length blocks.
- The prefix in an address defines the block (network); the suffix defines the node (device).
- Theoretically, we can have a block of 2^0, 2^1, 2^2, …, 2^{32} addresses.
- The number of addresses in a block needs to be a power of 2. An organization can be granted one block of addresses.



Address space

- The prefix length in classless addressing is variable.
- We can have a prefix length that ranges from 0 to 32.
- The size of the network is inversely proportional to the length of the prefix.
- A small prefix means a larger network; a large prefix means a smaller network.
- The idea of classless addressing can be easily applied to classful addressing.
- An address in class A can be thought of as a classless address in which the prefix length is 8.
- An address in class B can be thought of as a classless address in which the prefix is 16, and so on. In other words, classful addressing is a special case of classless addressing.

4.5. Address Masks

- To identify a network and route packets to the network, the routers outside the network use a default mask to get the network address.
- The default mask is a **32 bit mask** with the **net-id** portion containing **l's** and the **host-id** portion containing **0's**.

- The router **looks** at the **first byte** of the IP address to know the class and then **applies a default mask** (ANDs the masks with the IP address).
- In classless addressing, there is no class and hence, there is no default mask which can be applied.
- To solve this problem, a mask is defined using the **CIDR** (Classless Inter Domain Routing) notation.
- The notation used in classless addressing is informally referred to as *slash notation* and formally as **classless interdomain routing** or **CIDR.**
- The format of the mask is: a.b.c.d/n. Here, /n defines the mask (where n is the number of 1's in the mask).
- For example, 192.168.100.14/**24** represents the IP address 192.168.100.14 and, its subnet mask 255.255.255.0, which has 24 leading 1-bits.
- The following table shows the default masks for classes A, B and C. The last column shows the mask in the /n form.

Class	In Binary	In 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

Default Masks

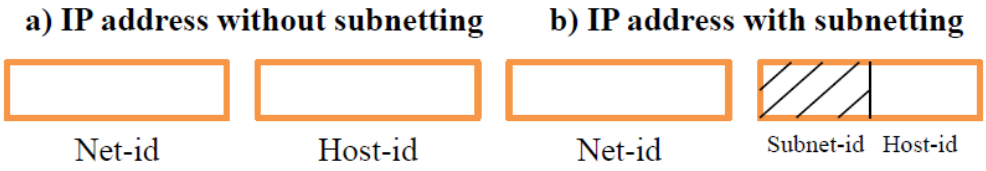
4.6. Types of IP communication or transmission methods

IPv4 basically supports three different types of addressing modes:

- **Unicast Addressing Mode**: This addressing mode is used to specify single sender and single receiver. Example: Accessing a website.
- **Broadcast Addressing Mode:** This addressing mode is used to send messages to all devices in a network. Example: sending a message in local network to all the devices.
- **Multicast Addressing Mode:** This addressing mode is typically used within a local network or across networks and sends messages to a group of devices. Example: Streaming audio to multiple devices at once.

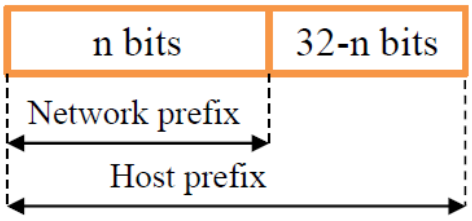
4.7. Subnetting

- An organization having a single network has one network address. The outside world recognizes the organizations network by this network address.
- However, the organization may want to divide all its hosts into different groups, i.e., the network is further divided into **subnetworks**.
- Subnetting is the process of dividing a network into smaller subnetworks, each having its own subnetwork address.
- To achieve subnetting, some part of the host-id is **borrowed** as the subnet-id.
- The number of bits borrowed will depend upon the number of subnets to be made. For example: If 8 subnetworks are to be made, 3 bits will be used as the subnet-id.
- We thus have three levels of hierarchy:
    1. Net-id (gives network address)
    2. Subnet-id (gives subnet address)
    3. Host-id (gives host address in subnet)

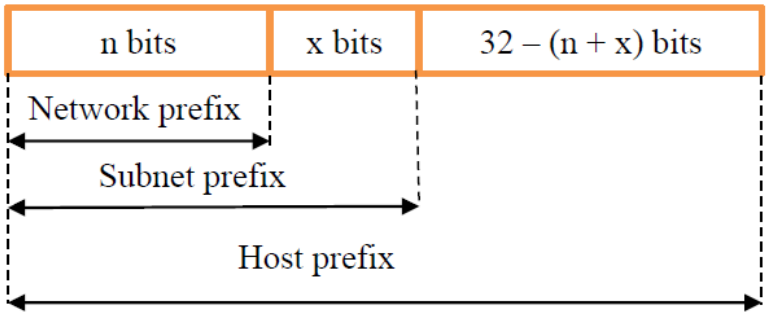


Subnetting

- With subnetting, we have a three level hierarchy. If we use the classless scheme, the 32 bits will be divided as shown.



Two level Hierarchy



Three level Hierarchy

- **How to Do Subnetting?**

1. **Understand Your Requirements:**
   - What is the original network? (e.g., 192.168.1.0/24)
   - How many subnets or hosts per subnet do you need?
2. **Determine Subnet Size:**

   Use the formula:

   $$2^n \geq \text{number of subnets or hosts needed}$$

- To create subnets: Find n such that $2^n \geq$ number of required **subnets**.
- To fit hosts per subnet: Find n such that $(2^n - 2) \geq$ number of required **hosts per subnet**.

3. **Calculate the New Subnet Mask:**
   - Original subnet: /24 (for example)
   - Add n bits for subnetting: new CIDR becomes /(24 + n)
4. **Determine Subnet Increments:**
   - The block size = $2^{(32-\text{new CIDR})}$
   - This tells you how much each subnet increases by in the fourth octet (for class C).
5. **List the Subnets:**

   Example: subnetting 192.168.1.0/24 into **4 subnets**:
   - Need 2 bits ($2^2 = 4$ subnets)
   - New mask: /24 + 2 = /26
   - Block size: $2^{(32-26)} = 64$
   - Subnets:
     - 192.168.1.0/26 → hosts: .1 to .62
     - 192.168.1.64/26 → hosts: .65 to .126
     - 192.168.1.128/26 → hosts: .129 to .190
     - 192.168.1.192/26 → hosts: .193 to .254
6. **Verify Usable Hosts:**
   - Number of host bits = 32 - CIDR
   - Use the formula again: $2^{(\text{Number of host bits})} - 2$
   - For /26, that's 64 - 2 = **62 usable hosts** per subnet.

- **Note:** The "-2" used for the calculation of the number of host bits is because two IP addresses in every subnet are reserved for:

1. Network Address (or Subnet Address):
   - The first address in the subnet

- Identifies the subnet itself
- Cannot be assigned to a host
- Example: In 192.168.1.0/24, 192.168.1.0 is the network address

2. Broadcast Address

- The last address in the subnet
- Used to send data to all devices in that subnet
- Cannot be assigned to a host
- To calculate the broadcast address, take the subnet address and set all host bits to 1.
- Example: In 192.168.1.0/24, 192.168.1.255 is the broadcast address

4.8. Supernetting

- Supernetting the opposite of subnetting. It is the process of combining subnets to create larger networks.
- Multiple subnetworks are aggregated back in a single large network so that they can be managed as a single network.
- The main purpose of supernetting is to reduce the number of entries in the routing tables and for faster routing. It is also known as route summarization and route aggregation.

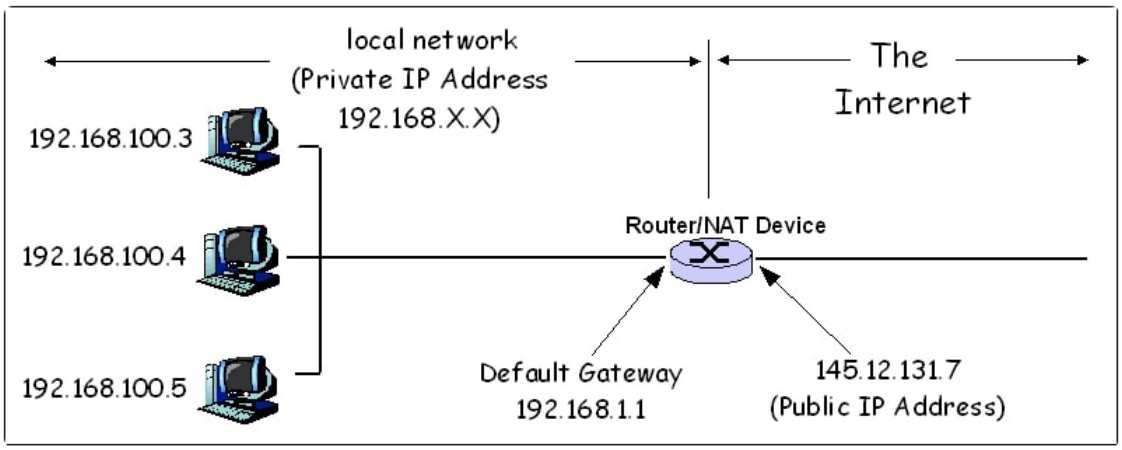
4.9. Network Address Translation

- Every machine in an internetwork must have a unique IP address.
- As the number of users keep growing, it is not possible to assign each user a separate IP address since the available addresses are not enough.
- A solution to this problem is Network Address Translation (NAT).
- To conserve IP address space, networks which are not directly connected to the Internet are often given private address space.
- The following table shows the private IP addresses:

Start	End	Network Size
10.0.0.0	10.255.255.255	2^{24}
172.16.0.0	172.31.255.255	2^{20}
192.168.0.0	192.168.255.255	2^{18}

- A network which uses private IP addresses internally is assigned a single public IP address to communicate with machines outside the network.

- All machines will use the public IP address to communicate outside. The following figure shows an example.



Public and Private IP addresses

- Here, all machines have a private IP address for communication inside their network.
- The whole network has a public IP address for communicating outside the network.
- **Address Translation:**
  - An address translation must take place between the internal private IP address and the public IP address outside. The outside world only knows the public IP of the entire network.
  - The NAT router stores a translation table. The translation table stores the mapping between the private IP address and the public IP address.
  - The translation takes place as follows:
    - All outgoing packets go through the NAT router which replaces the private source address with the public IP address.
    - All incoming packets are received by the NAT router. It replaces the destination IP address (public) of the packet with the correct private IP address.

5. Network layer protocols

Network protocols provide what are called "link services". These protocols handle addressing and routing information, error checking, and retransmission requests. Network protocols also define rules for communicating in a particular networking environment such as Ethernet or Token Ring.

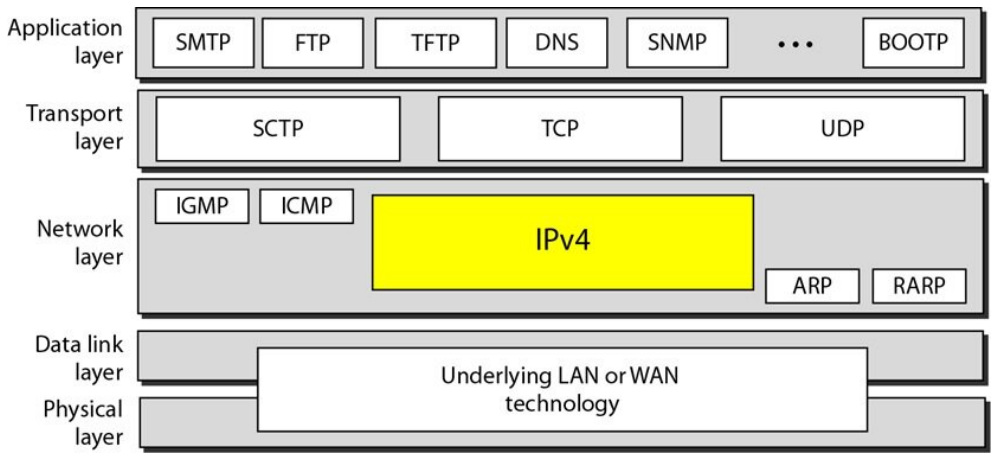
- **IP (Internet Protocol):** This is short for Internet Protocol which works at the OSI network layer and is a routed protocol for forwarding layer 3 packets.
- **ICMP:** ICMP stands for Internet Control Message Protocol. It is used to report some problem when routing a packet.

- **DHCP:** DHCP stands for Dynamic Host Configuration Protocol. It helps to get the network layer address for a host.
- **ARP:** ARP acronym for Address Resolution Protocol. It helps to finds the link layer address of a host.

6. IP (Internet Protocol)

The Internet's basic protocol called IP for Internet Protocol. The objective of starting this protocol is assigned to interconnect networks do not have the same frame-level protocols or package level. The internet acronym comes from inter-networking and corresponds to an interconnection fashion: each independent network must transport in the weft or in the data area of the packet an IP packet.

There are two generations of IP packets, called IPv4 (IP version 4) and IPv6 (IP version 6). IPv4 has been dominant so far. The transition to IPv6 could accelerate due to its adoption in many Asian countries. The transition is however difficult and will last many years. Internet Protocol (IP) of network layer contains addressing information and some control information that enables the packets to be routed.



Position of IPv4 in TCP/ IP protocol suite

- **How IP works:**
  - IP is designed to work over a dynamic network. This means that IP must work without a central directory or monitor, and that it cannot rely upon specific links or nodes existing. IP is a connectionless protocol that is datagram-oriented., so each packet must contain the source IP address, destination IP address, and other data in the header to be successfully delivered.
  - Combined, these factors make IP an unreliable, best effort delivery protocol. Error correction is handled by upper level protocols instead.

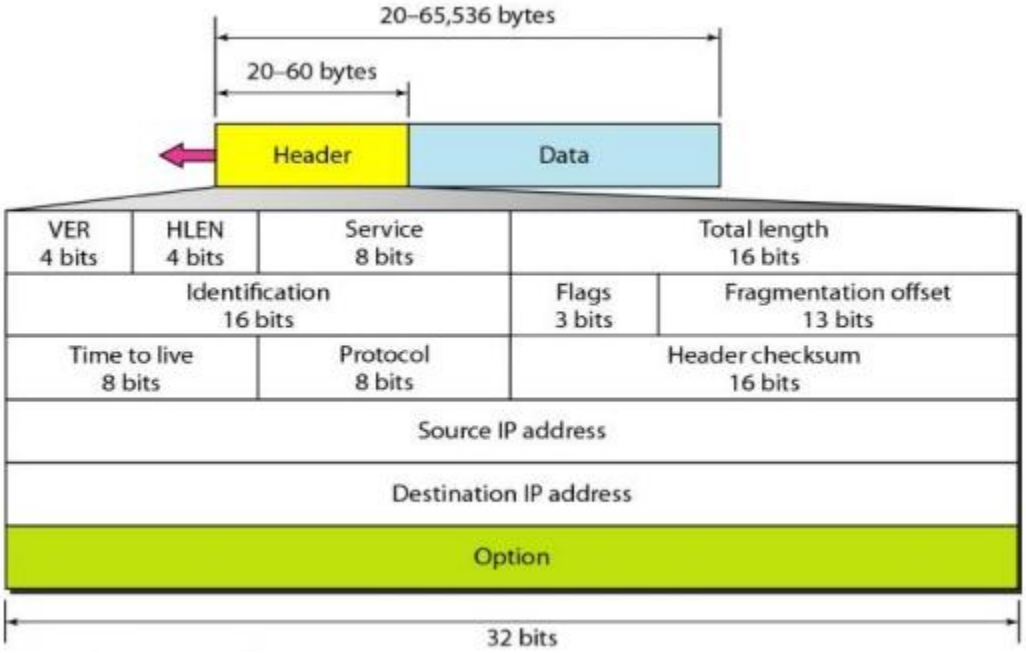
- **IP has two primary responsibilities:**

  (1) Providing connectionless, best effort delivery of datagrams through a internetwork. The term best effort delivery means that IP does not provides any error control or flow control. The term connectionless means that each datagram is handled independently, and each datagram can follow different route to the destination. This implies that datagrams sent by the same source to the same destination could arrive out of order.

  (2) Providing fragmentation and reassembly of datagrams to support data links with different maximum transmission unit (MTU) sizes.

6.1. IPv4 datagram

- The datagram has a header followed by a data area.
- The maximum datagram size is 65536 bytes, including the header.
- A header is minimum 20 bytes to maximum 60 bytes in length. The structure of an IP datagram is shown below.



IPv4 datagram format

The various fields are:

- **Version (VER):** This 4-bit field defines the version of the IP protocol. The current version is 4 (IPv4), with a binary value of 0100. IPv6 is the new version.
- **HLEN:** This 4-bit field defines length of the header in multiples of four bytes. The four bits can represent a number between 0 and 15, which then multiplied by 4, gives maximum header size, i.e., 60 bytes. The minimum header size is 20 bytes (when there are no options) and is represented by number 5 in the HLEN field.

- **Type Of Service (TOS):** This 8-bit field defines the priority of datagram.
- **Total Length:** This 16-bit field defines the total length of IP datagram (header + data). Since the maximum size of datagram is 65536 bytes, it is a 16-bit field.
- **Identification:** This 16-bit field is used in fragmentation to identify a fragment. A datagram may be divided into fragments when it passes through another network.
- **Flags:** This 3-bit field is used with fragmentation. The flags are D and M. D = Don't fragment, M = More which indicates whether more fragments follow. If it is 0, it indicates that it is the last fragment.
- **Fragment offset:** This 13-bit field is used as a pointer that shows the location of the data in the original datagram if it is fragmented.
- **Time-to-Live (TTL):** This 8-bit field is the lifetime of the datagram. It is initialized by the sender and decremented by each router that handles the datagram. When this field reaches 0, the datagram is thrown away, and the sender is notified with an ICMP (Internet Control Message Protocol) message.
- **Protocol:** This field indicates which upper layer protocol data are encapsulated in the datagram (TCP, UDP, ICMP etc.).
- **Header Checksum:** This 16-bit field is used to check for errors in the header only and not for the rest of the packet.
- **Source IP address and Destination IP address:** Every IP datagram contains 32-bit source IP address of the sender and 32-bit IP address of the receiver for the destination of the datagram
- **Options:** This field is of variable length, allows the packet to request special features such as security level, route to be taken by the packet, and timestamp at each router. They are normally used for network testing and debugging.

6.2. IPv6

- IPv6 is the recent version of IP.
- It was designed to overcome the limitations of IPv4.
- One of the main limitation of IPv4 is the limited address space (32 bit address divided into classes). As more and more devices are connected to the internet, this address space is not enough. IPv6 provides a 128 bit address and can accommodate 2^{128} devices.
- IPv6 addresses are represented as eight groups of four hexadecimal digits with the groups being separated by colons.
- Example 2001:0db8:0000:0042:0000:8a2e:0370:7334

- **IPv6 Address:** An IPv6 address consists of 128 bits. These 128 bits are logically divided into eight 16-bits blocks. Each block is then converted into 4-digit Hexadecimal numbers separated by colon symbols.

  0010000000000001 **0000000000000000** 0011001000111000 **1101111111100001**

  0000000001100011 **0000000000000000** 0000000000000000 **1111111011111011**

  Each block is then converted into Hexadecimal and separated by ":" symbol:

  0010000000000001 **0000000000000000** 0011001000111000 **1101111111100001**

  2001 0000 3238 DFE1

  0000000001100011 **0000000000000000** 0000000000000000 **1111111011111011**

  0063 0000 0000 FEFB

  Thus, address is: **2001:0000:3238:DFE1:0063:0000:0000:FEFB**

  As seen above, the address is very long. There are some ways to reduce the length of the address:

  1. Remove leading Zero(es): Any 0's appearing at the beginning of a block can be discarded. For example, 0063 can be written as 63.

  2. Replace consecutive blocks of zeroes with:: If two of more blocks contain consecutive zeroes, omit them all and replace with double colon sign (::) for example, 6th and 7th block. Consecutive blocks of zeroes can be replaced only once by ::. So if there are still blocks of zeroes in the address, they can be shrunk down to a single zero, such as the 2nd block.

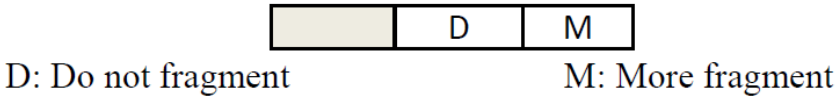
  Thus the address reduces to: **2001:0:3238:DFE1:63::FEFB**

7. Fragmentation

- From the source to the destination machine, a datagram may have to travel across several heterogeneous networks.
- Each network has its own MTU (Maximum Transfer Unit) size. Hence, it may be broken into smaller units. This is called fragmentation.
- The fragmentation takes place at the source or any router in the path. Each fragment is sent independently of other fragments and reassembly is only done at the receiver.

- **Fields in the datagram header related to fragmentation:**
  - **Identification:** This 16 bit field identifies all fragments belonging to a single datagram. When a datagram is fragmented, this value is copied into all fragments. This helps in reassembling the datagram.

- **Flags:** It is a 3 bit field. The first bit is not used. The second is D, i.e., Do not fragment. If set, the datagram cannot be fragmented. The second flag is M, i.e., More Fragments. If set, it indicates that there are more fragments of the same datagram, and if 0, this fragment is the last fragment.



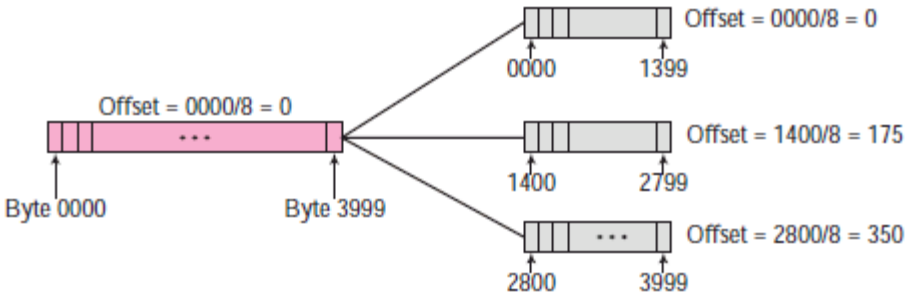
Flags

- **Offset:** This 13 bit field gives the relative position of the fragment in the datagram. If 0, it indicates that this is the first fragment. It is measured in units of 8 bytes. For example, if the second fragment carries bytes 800 to 1000, the offset field will have value 800/8 = 100.

  Example: The following diagram shows a datagram carrying 4000 bytes fragmented into 3 fragments. Fragment 1 carries bytes 0 – 1399

  Fragment 2 carries bytes 1400 – 2799

  Fragment 3 carries bytes 2800 – 3999



Fragmentation Example

- To reassemble fragments, the receiver uses the following process:
  - Find the fragment having offset 0. This is the first fragment.
  - Divide its length by 8. This gives the offset of the second fragment.
  - Locate the second fragment (having same identification and offset = calculated offset value)
  - Calculate position of third fragment using [length(fragment 1) + length fragment 2)]/ 8
  - Continue in the same manner till a fragment with M flag = 0 is reached.

8. Routing algorithms

The main function of the network layer is to route packets from source machine to the destination machine. To accomplish this a route through the network must be selected, generally more than one route is possible. The selection of route is generally based on some performance criteria.

The simplest criteria is to choose shortest route through the network. Shortest route means a route that passes through the least number of nodes. This shortest route selection results in least number of hops per packet.

A routing algorithm is a part of network layer software.

8.1. Desired Properties of a Routing Algorithm

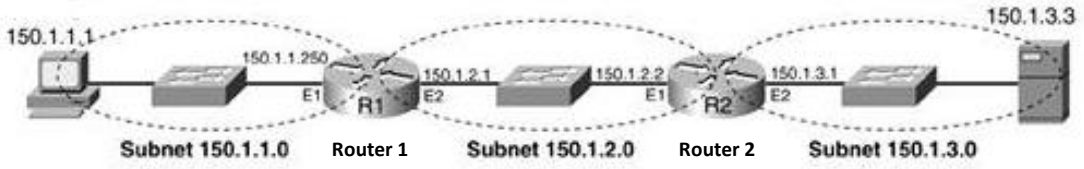
- **Correction:** The routing should be done properly and correctly so that the packets may reach their proper destination.
- **Simplicity:** The routing should be done in a simple manner so that the over head is as low as possible.
- **Robustness:** Once a major network becomes operative, it may be expected to run continuously for years without any failure.
- **Stability:** The routing algorithm should be stable under all possible circumstances.
- **Fairness:** Every node connected to the network gets a fair change of transmitting their packets. This is generally done on a first come first serve basis.
- **Optimality:** The routing algorithms should be optimal in terms of throughput and minimizing mean packet delays. Here there is a trade off and one has to choose depending on his suitability.

8.2. Routing tables

Once the routing decision is made, this information is to be stored in routing table so that the router knows how to forward a packet.

In virtual circuit packet switching the routing table contains each incoming packet number and outgoing packet number and output port to which the packet is to forwarded.

In datagram networks, routing table contains the next hop to which to forwarded the packet, based on destination address.



R1's Routing Table

Destination Subnets	Interface	Next-Hop Router
150.1.1.0	E1	N/A
150.1.2.0	E2	N/A
150.1.3.0	**E2**	**150.1.2.2**

R2 Routing Table

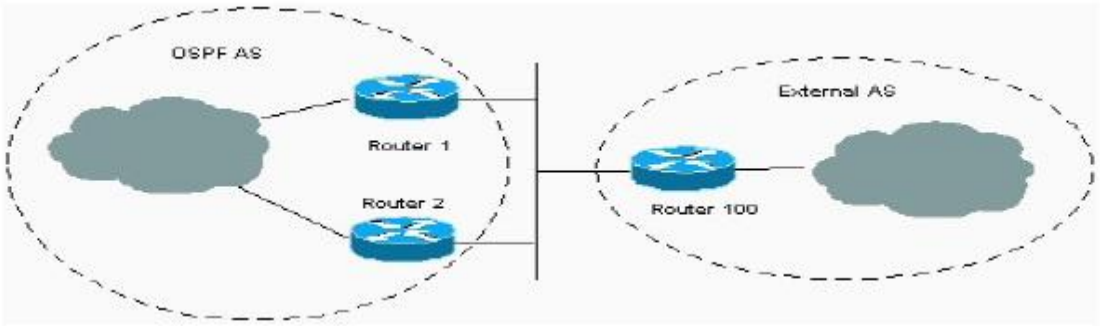
Destination Subnets	Interface	Next-Hop Router
150.1.2.0	E1	N/A
150.1.3.0	E2	N/A
150.1.1.0	**E1**	**150.1.2.1**

Example of routing tables

8.3. Forwarding

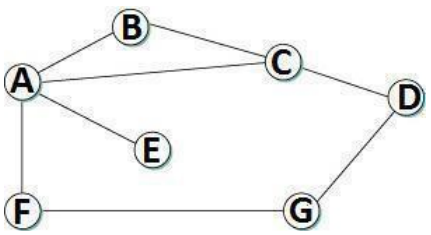
Is the sending of packets of nodes and links along a path.

One can think of a router as having two processes inside it. One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing tables. This process is forwarding.



8.4. Distance Vector Routing

- Distance vector routing is **distributed**, i.e., algorithm is run on all nodes.
- Each node knows the distance (cost) to each of its directly connected neighbors.
- Nodes construct a **vector** (Destination, Cost, NextHop) and distributes to its neighbors.
- Nodes compute routing table of minimum distance to every other node via NextHop using information obtained from its neighbors.
- **Initial State:**



- In given network, cost of each link is 1 hop.
- Each node sets a distance of 1 (hop) to its immediate neighbor and cost to itself as 0.
- Distance for non-neighbors is marked as unreachable with value ∞ (infinity).
- For node A, nodes B, C, E and F are reachable, whereas nodes D and G are unreachable.

Destination	Cost	NextHop
A	0	A
B	1	B
C	1	C
D	∞	—
E	1	E
F	1	F
G	∞	—

Node A's initial table

Destination	Cost	NextHop
A	1	A
B	1	B
C	0	C
D	1	D
E	∞	—
F	∞	—
G	∞	—

Node C's initial table

Destination	Cost	NextHop
A	1	A
B	∞	—
C	∞	—
D	∞	—
E	∞	—
F	0	F
G	1	G

Node F's initial table

- The initial table for all the nodes are given below:

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	∞	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

- Each node sends its initial table (distance vector) to neighbors and receives their estimate.
- Node A sends its table to nodes B, C, E & F and receives tables from nodes B, C, E & F.
- Each node updates its routing table by comparing with each of its neighbor's table.
- For each destination, Total Cost is computed as:

  Total Cost = *Cost (Node to Neighbor) + Cost (Neighbor to Destination)*
- If Total Cost < Cost then

  Cost = Total Cost and NextHop = Neighbor

- Node A learns from C's table to reach node D and from F's table to reach node G.
- Total Cost to reach node D via C = Cost(A to C) + Cost(C to D)

  Cost = 1 + 1 = 2.

  Since 2 < ∞, entry for destination D in A's table is changed to (D, 2, C)

  Total Cost to reach node G via F = Cost(A to F) + Cost(F to G) = 1 + 1 = 2

  Since 2 < ∞, entry for destination G in A's table is changed to (G, 2, F)

  Each node builds complete routing table after few exchanges amongst its neighbors.

Node A's final routing table

Destination	Cost	NextHop
A	0	A
B	1	B
C	1	C
D	2	C
E	1	E
F	1	F
G	2	F

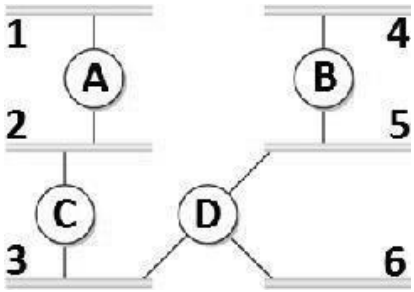
- System stabilizes when all nodes have complete routing information, i.e., **convergence.**
- Routing tables are exchanged periodically or in case of triggered update.
- The final distances stored at each node is given below:

Final 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

- **Updation of Routing Tables:** There are two different circumstances under which a given node decides to send a routing update to its neighbors:
  - ***Periodic Update:***
    - In this case, each node automatically sends an update message every so often, even if nothing has changed.
    - The frequency of these periodic updates varies from protocol to protocol, but it is typically on the order of several seconds to several minutes.
  - ***Triggered Update:***
    - In this case, whenever a node notices a link failure or receives an update from one of its neighbors that causes it to change one of the routes in its routing table.
    - Whenever a node's routing table changes, it sends an update to its neighbors, which may lead to a change in their tables, causing them to send an update to their neighbors.
- **ROUTING INFORMATION PROTOCOL (RIP):**

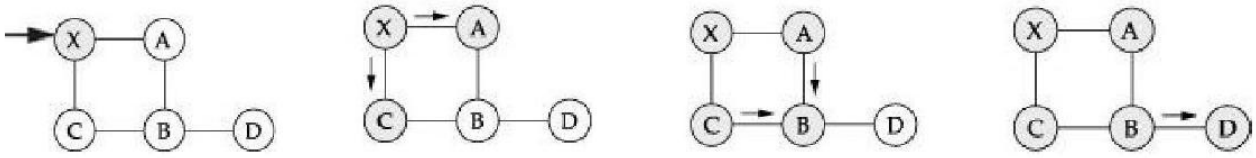
  RIP is an intra-domain routing protocol based on distance-vector algorithm.
  - **Example:**



    - Routers advertise the cost of reaching networks. Cost of reaching each link is 1 hop. For example, router C advertises to A that it can reach network 2, 3 at cost 0 (directly connected), networks 5, 6 at cost 1 and network 4 at cost 2.
    - Each router updates cost and next hop for each network number.
    - Infinity is defined as 16, i.e., any route cannot have more than 15 hops. Therefore RIP can be implemented on small-sized networks only.
    - Advertisements are sent every 30 seconds or in case of triggered update.

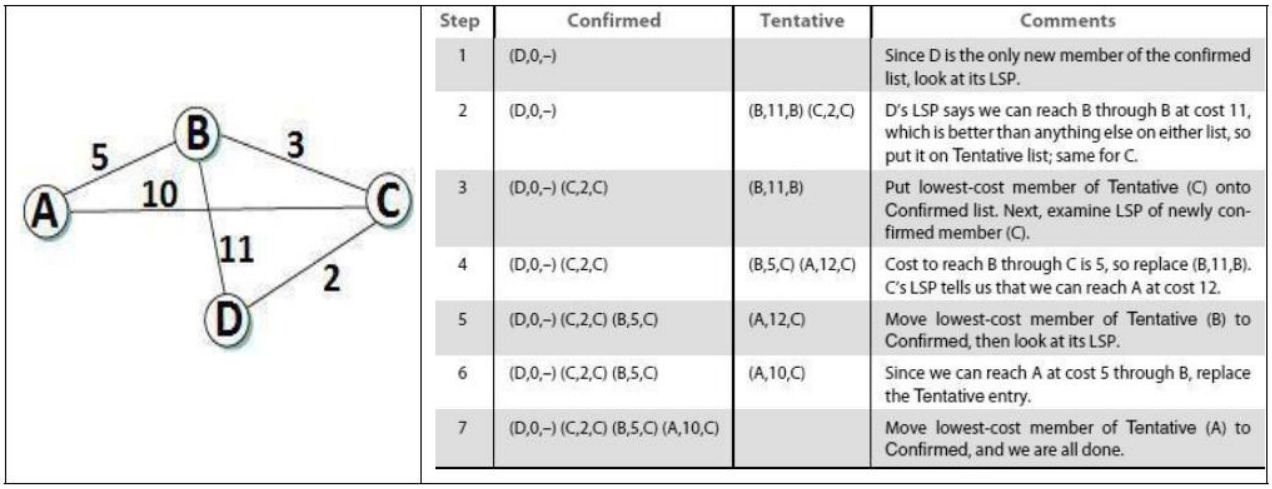
8.5. Link State Routing (LSR)

- Each node knows state of link to its neighbors and cost.
- Nodes create an update packet called link-state packet (LSP) that contains:
  - ID of the node
  - List of neighbors for that node and associated cost
  - 64-bit Sequence number
  - Time to live
- Link-State routing protocols rely on two mechanisms:
  - ***Reliable flooding*** of link-state information to all other nodes
  - ***Route calculation*** from the accumulated link-state knowledge
- **Reliable Flooding:**
  - Each node sends its LSP out on each of its directly connected links.
  - When a node receives LSP of another node, checks if it has an LSP already for that node.
  - If not, it stores and forwards the LSP on all other links except the incoming one.
  - Else if the received LSP has a bigger sequence number, then it is stored and forwarded. Older LSP for that node is discarded.
  - Otherwise discard the received LSP, since it is not latest for that node.
  - Thus recent LSP of a node eventually reaches all nodes, i.e., reliable flooding.



  - Flooding of LSP in a small network is as follows:
    - When node X receives Y's LSP (a), it floods onto its neighbors A and C (b)
    - Nodes A and C forward it to B, but does not sends it back to X (c).
    - Node B receives two copies of LSP with same sequence number.
    - Accepts one LSP and forwards it to D (d). Flooding is complete.
  - LSP is generated either periodically or when there is a change in the topology.

- **Route Calculation:**
  - Each node knows the entire topology, once it has LSP from every other node.
  - Forward search algorithm is used to compute routing table from the received LSPs.
  - Each node maintains two lists, namely Tentative and Confirmed with entries of the form (Destination, Cost, NextHop).
- **DIJKSTRA'S SHORTEST PATH ALGORITHM (FORWARD SEARCH ALGORITHM)**
  - Each host maintains two lists, known as **Tentative** and **Confirmed**
  - Initialize the Confirmed list with an entry for the Node (Cost = 0).
  - Node just added to Confirmed list is called Next. Its LSP is examined.
  - For each neighbor of Next, calculate cost to reach each neighbor as Cost (Node to Next) + Cost (Next to Neighbor).
  - If Neighbor is neither in Confirmed nor in Tentative list, then add (Neighbor, Cost, NextHop) to Tentative list.
  - If Neighbor is in Tentative list, and Cost is less than existing cost, then replace the entry with (Neighbor, Cost, NextHop).
  - If Tentative list is empty then Stop, otherwise move least cost entry from Tentative list to Confirmed list. Go to Step 2.
  - **Example:**



Step	Confirmed	Tentative	Comments
1	(D,0,–)		Since D is the only new member of the confirmed list, look at its LSP.
2	(D,0,–)	(B,11,B) (C,2,C)	D's LSP says we can reach B through B at cost 11, which is better than anything else on either list, so put it on Tentative list; same for C.
3	(D,0,–) (C,2,C)	(B,11,B)	Put lowest-cost member of Tentative (C) onto Confirmed list. Next, examine LSP of newly confirmed member (C).
4	(D,0,–) (C,2,C)	(B,5,C) (A,12,C)	Cost to reach B through C is 5, so replace (B,11,B). C's LSP tells us that we can reach A at cost 12.
5	(D,0,–) (C,2,C) (B,5,C)	(A,12,C)	Move lowest-cost member of Tentative (B) to Confirmed, then look at its LSP.
6	(D,0,–) (C,2,C) (B,5,C)	(A,10,C)	Since we can reach A at cost 5 through B, replace the Tentative entry.
7	(D,0,–) (C,2,C) (B,5,C) (A,10,C)		Move lowest-cost member of Tentative (A) to Confirmed, and we are all done.

- **Difference Between Distance-Vector And Link-State Algorithms**
  - ***Link state Routing:*** Each node talks to all other nodes, but it tells them only what it knows for sure (i.e., only the state of its directly connected links).
  - ***Distance vector Routing;*** Each node talks only to its directly connected neighbors, but it tells them everything it has learned (i.e., distance to all nodes).

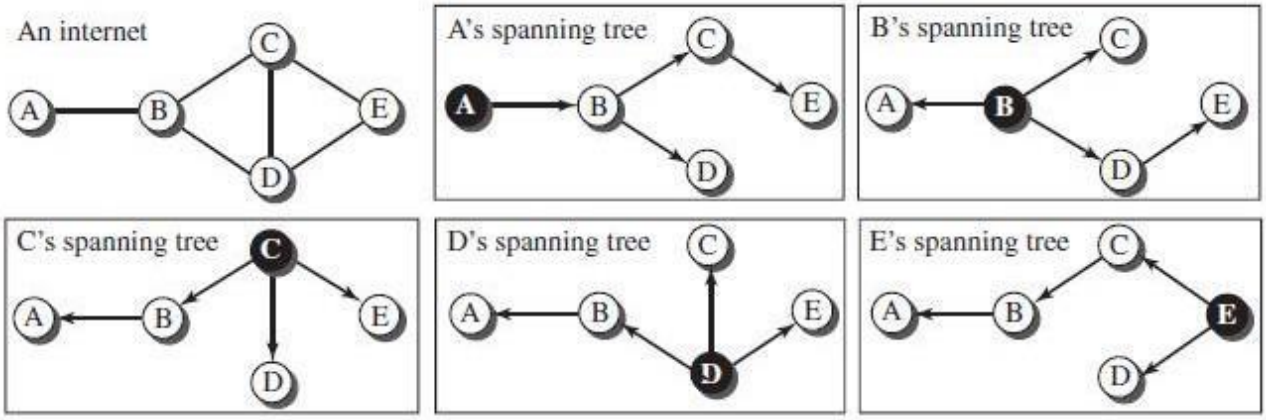
8.6. Path Vector Routing (PVR)

- Path-vector routing is an asynchronous and distributed routing algorithm.
- The Path-vector routing is not based on least-cost routing.
- The best route is determined by the source using the policy it imposes on the route.
- In other words, the source can control the path.
- Path-vector routing is not actually used in an internet, and is mostly designed to route a packet between ISPs.
- **Spanning Trees:**
  - In path-vector routing, the path from a source to all destinations is determined by the best spanning tree.
  - The best spanning tree is not the least-cost tree.
  - It is the tree determined by the source when it imposes its own policy.
  - If there is more than one route to a destination, the source can choose the route that meets its policy best.
  - A source may apply several policies at the same time.
  - One of the common policies uses the minimum number of nodes to be visited. Another common policy is to avoid some nodes as the middle node in a route.
  - The spanning trees are made, gradually and asynchronously, by each node. When a node is booted, it creates a path vector based on the information it can obtain about its immediate neighbor.
  - A node sends greeting messages to its immediate neighbors to collect these pieces of information.
  - Each node, after the creation of the initial path vector, sends it to all its immediate neighbors.
  - Each node, when it receives a path vector from a neighbor, updates its path vector using the formula:

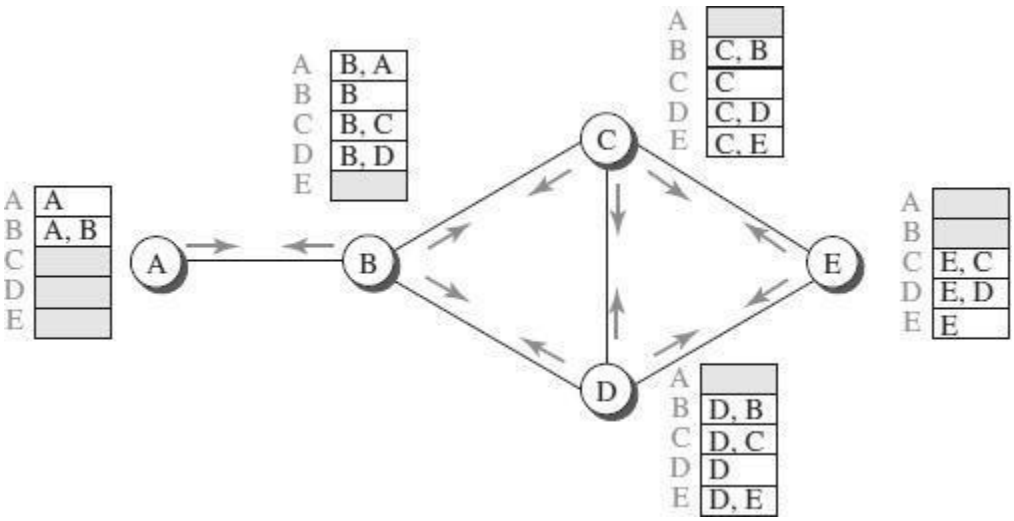
$$\text{Path}(x, y) = \text{best}\ \{\text{Path}(x, y), [(x + \text{Path}(v, y)]\} \quad \text{for all } v\text{'s in the internet.}$$

  - The policy is defined by selecting the best of multiple paths.
  - Path-vector routing also imposes one more condition on this equation.
  - If Path (v, y) includes x, that path is discarded to avoid a loop in the path.
  - In other words, x does not want to visit itself when it selects a path to y.
  - **Example:**
  - The figure below shows a small internet with only five nodes.
  - Each source has created its own spanning tree that meets its policy.

- The policy imposed by all sources is to use the minimum number of nodes to reach a destination.
- The spanning tree selected by A and E is such that the communication does not pass through D as a middle node.
- Similarly, the spanning tree selected by B is such that the communication does not pass through C as a middle node.

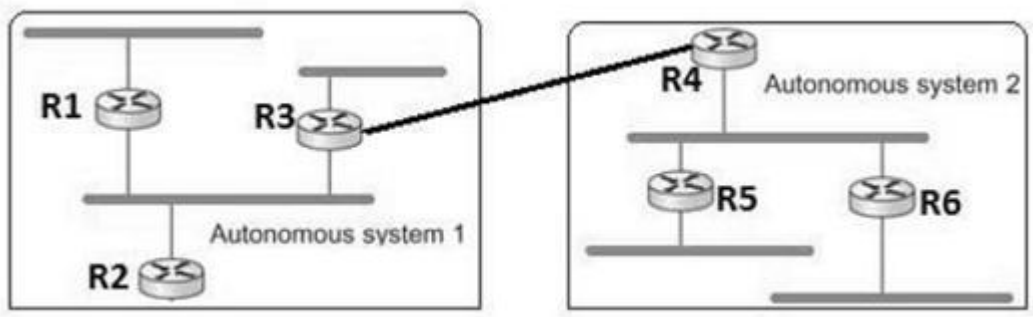


- **Path Vectors made at booting time:**
  - The figure below shows all of these path vectors for the example.
  - Not all of these tables are created simultaneously.
  - They are created when each node is booted.
  - The figure also shows how these path vectors are sent to immediate neighbors after they have been created.

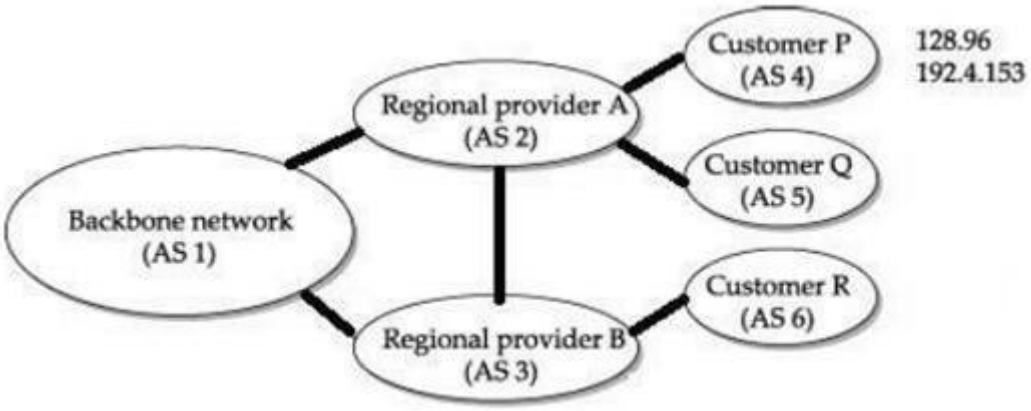


- **BORDER GATEWAY PROTOCOL (BGP):**
  - The Border Gateway Protocol version (BGP) is the only interdomain routing protocol used in the Internet today.
  - BGP4 is based on the path-vector algorithm. It provides information about the reachability of networks in the Internet.

- BGP views internet as a set of autonomous systems interconnected arbitrarily.



- Each AS (Autonomous System) have a border router (gateway), by which packets enter and leave that AS. In above figure, R3 and R4 are border routers.
- One of the router in each autonomous system is designated as BGP speaker.
- BGP Speaker exchange reachability information with other BGP speakers, known as external BGP session.
- BGP advertises complete path as enumerated list of AS (path vector) to reach a particular network.
- Paths must be without any loop, i.e., AS list is unique.
- For example, backbone network advertises that networks 128.96 and 192.4.153 can be reached along the path <AS1, AS2, AS4>.



- If there are multiple routes to a destination, BGP speaker chooses one based on policy.
- Speakers need not advertise any route to a destination, even if one exists.
- Advertised paths can be cancelled, if a link/node on the path goes down. This negative advertisement is known as withdrawn route.
- Routes are not repeatedly sent. If there is no change, keep alive messages are sent.