**4. Network layer**

**Design Issues in Network Layer**

[Network layer](https://www.geeksforgeeks.org/layers-of-osi-model/) is majorly focused on getting packets from the source to the destination, routing error handling and congestion control.

Before learning about design issues in the network layer, let’s learn about its various functions.

* [**Addressing**](https://www.geeksforgeeks.org/introduction-of-classful-ip-addressing/)**:**  
  Maintains the address at the frame header of both source and destination and performs addressing to detect various devices in network.
* **Packetizing:**  
  This is performed by Internet Protocol. The network layer converts the packets from its upper layer.
* [**Routing**](https://www.geeksforgeeks.org/types-of-routing/)**:**  
  It is the most important functionality. The network layer chooses the most relevant and best path for the data transmission from source to destination.
* **Inter-networking:**  
  It works to deliver a logical connection across multiple devices.

**Network layer design issues:**

The network layer comes with some design issues they are described as follows:

**1. Store and Forward packet switching:**

The host sends the packet to the nearest router. This packet is stored there until it has fully arrived once the link is fully processed by verifying the checksum then it is forwarded to the next router till it reaches the destination. This mechanism is called “Store and Forward packet switching.”

**2. Services provided to**[**Transport Layer**](https://www.geeksforgeeks.org/transport-layer-responsibilities/)**:**

Through the network/transport layer interface, the network layer transfers its services to the transport layer. These services are described below.

Based on the connections there are 2 types of services provided:

* **Connectionless –** The routing and insertion of packets into subnet is done individually. No added setup is required.
* **Connection-Oriented –** Subnet must offer reliable service and all the packets must be transmitted over a single route.

**3. Implementation of**[**Connectionless Service**](https://www.geeksforgeeks.org/difference-between-connection-oriented-and-connection-less-services/)**:**

Packet are termed as “datagrams” and corresponding subnet as “datagram subnets”. When the message size that has to be transmitted is 4 times the size of the packet, then the network layer divides into 4 packets and transmits each packet to router via a few protocol. Each data packet has destination address and is routed independently irrespective of the packets.

**4. Implementation of Connection Oriented service:**

To use a connection-oriented service, first we establish a connection, use it and then release it. In connection-oriented services, the data packets are delivered to the receiver in the same order in which they have been sent by the sender.

It can be done in either two ways :

* **Circuit Switched Connection –** A dedicated physical path or a circuit is established between the communicating nodes and then data stream is transferred.
* **Virtual Circuit Switched Connection –** The data stream is transferred over a packet switched network, in such a way that it seems to the user that there is a dedicated path from the sender to the receiver. A virtual path is established here. While, other connections may also be using the same path.

**Communication primitives**

### What is Unicast?

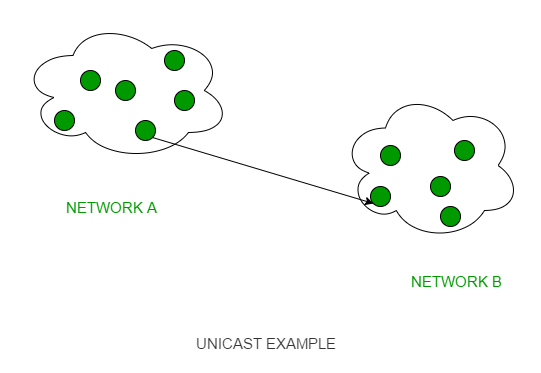
Unicast is a type of communication where data is sent from one computer to another computer.

In Unicast type of communication, there is only one sender, and one receiver.

Example:

1) Browsing a website. (Webserver is the sender and your computer is the receiver.)

2) Downloading a file from a FTP Server. (FTP Server is the sender and your computer is the receiver.)

****

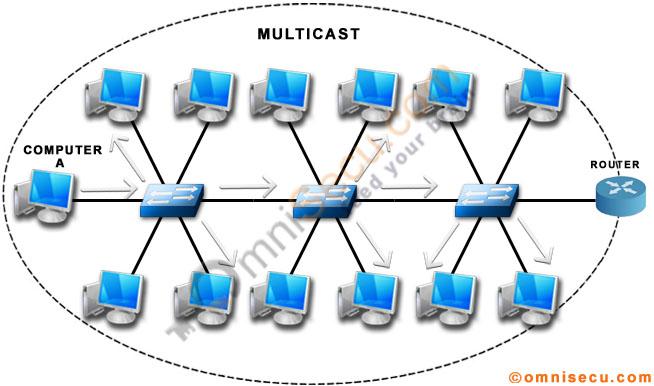
### What is Multicast?

Multicast is a type of communication where multicast traffic addressed for a group of devices on the network. IP multicast traffic are sent to a group and only members of that group receive and/or process the Multicast traffic.

Devices which are interested in a particular Multicast traffic must join to that Multicast group to receive the traffic. IP Multicast Groups are identified by Multicast IP Addresses ([IPv4 Class D Addresses](https://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php))

In Multicast, the sender transmit only one copy of data and it is delivered and/or processed to many devices (Not as delivered and processed by all devices as in Broadcast) who are interested in that traffic.

Example : Multicast Windows Deployment Services (WDS) OS deployment traffic, IP TV etc



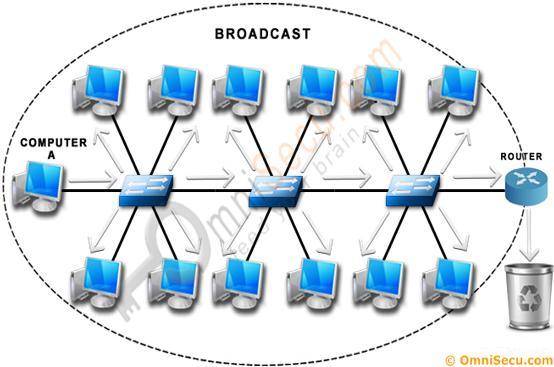
### What is Broadcast?

Broadcast is a type of communication where data is sent from one computer once and a copy of that data will be forwarded to all the devices.

In Broadcast, there is only one sender and the data is sent only once. But the Broadcast data is delivered to all connected devices.

Switches by design will forward the broadcast traffic and Routers by design will drop the broadcast traffic. In other words, Routers will not allow a broadcast from one LAN to cross the Router and reach another Network Segment. The primary function of a Router is to divide a big [Broadcast domain](https://www.omnisecu.com/cisco-certified-network-associate-ccna/what-are-collision-domain-and-broadcast-domain.php) to multiple smaller [Broadcast domain](https://www.omnisecu.com/cisco-certified-network-associate-ccna/what-are-collision-domain-and-broadcast-domain.php).

Example: [ARP Request message](https://www.omnisecu.com/tcpip/address-resolution-protocol-arp.php), [DHCP DISCOVER Message](https://www.omnisecu.com/tcpip/dhcp-dynamic-host-configuration-protocol-how-dhcp-works.php)



**IP Address**

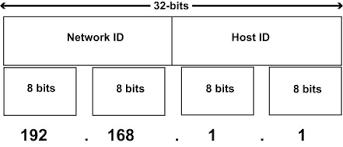
IP stands for Internet Protocol and describes a set of standards and requirements for creating and transmitting data packets, or datagrams, across networks. The Internet Protocol (IP) is part of the Internet layer of the Internet protocol suite. In the OSI model, IP would be considered part of the network layer. IP is traditionally used in conjunction with a higher-level protocol, most notably TCP. The IP standard is governed by RFC 791.

An IP address (*internet protocol address*) is a numerical representation that uniquely identifies a specific interface on the network.

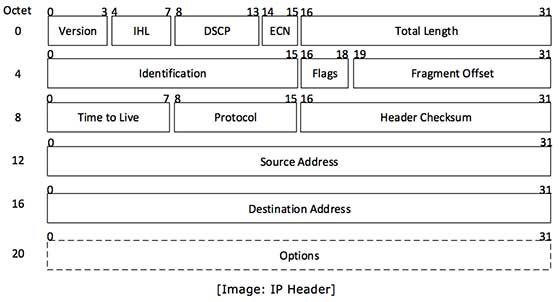
Addresses in IPv4 are 32-bits long. This allows for a maximum of 4,294,967,296 (232) unique addresses. Addresses in IPv6 are 128-bits, which allow for 3.4 x 1038 (2128) unique addresses.

The total usable address pool of both versions is reduced by various reserved addresses and other considerations.

IP addresses are binary numbers but are typically expressed in decimal form (IPv4) or hexadecimal form (IPv6) to make reading and using them easier for humans.



**Structure of IP frame header**



IP header includes many relevant information including Version Number, which, in this context, is 4. Other details are as follows −

* **Version** − Version no. of Internet Protocol used (e.g. IPv4). 0100/0101
* **IHL** − Internet Header Length; Length of entire IP header.
* **DSCP** − Differentiated Services Code Point; this is Type of Service.
* **ECN** − Explicit Congestion Notification; It carries information about the congestion seen in the route.
* **Total Length** − Length of entire IP Packet (including IP header and IP Payload).
* **Identification** − If IP packet is fragmented during the transmission, all the fragments contain same identification number to identify original IP packet they belong to.
* **Flags** − As a required by the network resources, if IP Packet is too large to handle, these ‘flags’ tells if they can be fragmented or not.
* **Fragment Offset** − This is a offset tells the exact position of the fragment in the original IP Packet.
* **Time to Live** − **To avoid looping in the network, every packet is sent with some TTL value set, which tells the network how many routers (hops) this packet can cross. At each hop, its value is decremented by one and when the value reaches zero, the packet is discarded.**
* **Protocol** − Tells the Network layer at the destination host, to which Protocol this packet belongs to, i.e. the next level Protocol. For example protocol number of ICMP is 1, TCP is 6 and UDP is 17.
* **Header Checksum** − This field is used to keep checksum value of entire header which is then used to check if the packet is received error-free.
* **Source Address** − 32-bit address of the Sender (or source) of the packet.
* **Destination Address** − 32-bit address of the Receiver (or destination) of the packet.
* **Options** − This is optional field, which is used if the value of IHL is greater than 5. These options may contain values for options such as Security, Record Route, Time Stamp, etc

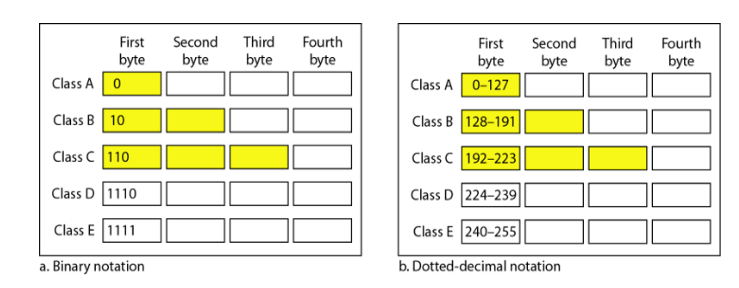
### **Classful Addressing**

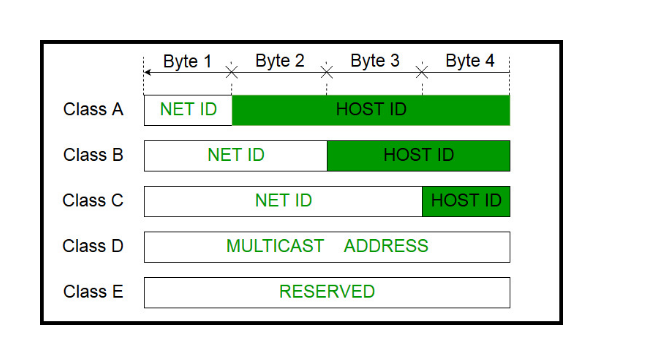
In Classful addressing, the address space is divided into five classes: A, B, C, D, and E. Each of these classes has a valid range of IP addresses. Classes D and E are reserved for multicast and experimental purposes respectively. The order of bits in the first octet determines the classes of IP address.

IPv4 address is divided into two parts:

Net-id: The net-id denotes the address of the network.  
Host-id: The hoist-id denotes the address of the host attached to the corresponding network.

The class of IP address is used to determine the bits used for network ID and host ID and the number of total networks and hosts possible in that particular class. Each ISP or network administrator assigns an IP address to each device that is connected to its network.





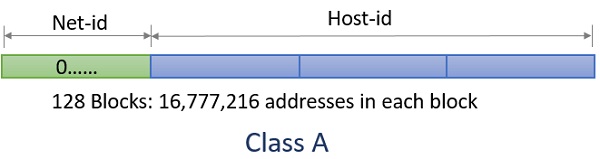
**01111111 . 11111111. 11111111. 11111111**

### Classes of Classful address

#### Class A

The **network id** of class A is defined by the **first byte** of the 32-bit IPv4 address. In class A, the **first bit** of the **net-id**stays ‘**0′**to define that the IPv4 address belongs to the class A and the other **7 bits** of the net-id can be changed to defines different blocks in class A. As the first bit is preserved the remaining seven bits calculate the number of blocks in the class A i.e. **27= 128 blocks**. There are 128 blocks in class A, as the addressing would start from 0 the range of blocks will be from 0-127.

The **host-id**in class A is defined by the **remaining three bytes** of the IPv4 address which is equal to 24 bits. So, we can calculate the **number of hosts for each block** as**224=16,777,216.** So, we conclude that we can assign 128 blocks from class A to 128 organizations where each organization can have 16,777,216 hosts connected to the network.



00000000 0

011111111 127

2^7 networks 128 126

2^24 **16,777,216-2**

**2^31**

**64.0.0.0 64.255.255.255**

64.0.0.8

01000000.00000000.00000000.00001000

**255.0.0.0 Default Mask for class A**

**11111111.00000000.00000000.00000000**

**010000000.0000000.0000000.0000000**

**64.0.0.0**

Now, as we have calculated the number of blocks and the number of addresses in each block of class A. Let us count the total number of addresses in class A which can be calculated as follow:

As we have seen above the **first bit** of the entire **32-bit addresses** of **class A** stays ‘**0**’. The remaining **31 bits** of 32-bit addresses can be changed to define the **address space** of **class A** i.e. **231= 2,147,483,648.**

**0-127 128 126 00000000 127.0.0.1 127.255.255.255**

**1-126**

**91.0.0.1**

**50.0.0.1 0-999**

**2^7**

**16,777,216-1000**

**2^7 =128-2**

**126**

* + - 1. **16,777,216**

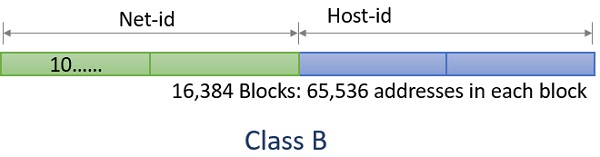
**1011**

#### Class B

128-191

The **network id** or the net-id of **class B** is defined using the **first two bytes** of the IPv4 address. The first **two bits** of **net-id** stays ‘**10**’ to define that the IPv4 address belongs to the**class B** and the remaining **14 bits** of net-id can be changed to calculate the number of **blocks** in class B i.e. **214= 16,384.**

The **next two bytes** to of IPv4 address denotes the**host id** in class B which is **16 bits**. The number of hosts can be calculated as **216= 65,536**. So, we conclude that we can assign 16,384 blocks from class B to 16,384 organizations where each organization can have 65,536 hosts connected to the network.



128-191 64

128 16384

129 16384

**2^14**

**2^16**

**128.0.0.8 10000000.00000000.0000000.00001000**

**255.255.0.0 11111111.11111111.00000000.00000000**

**10000000.0.0.0**

**128.0.0.0**

Now, as we have calculated the number of blocks and the number of addresses in each block of class B. Let us count the total number of addresses in class B which can be calculated as follow:

As we have seen above the **first two bits** of the entire 32-bit addresses of class B stays ‘**10**’ to define the class. The remaining **30 bits** of entire 32-bit addresses can be changed to define the **address space** of **class B** i.e. **230= 1,073,741,824**.

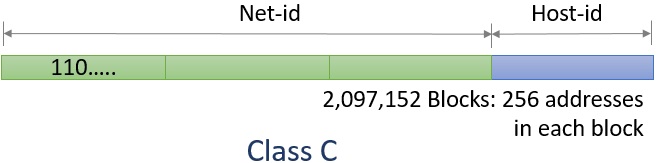
#### Class C

**192-223**

In class C the **network id is** defined by the **first 3 bytes** of the IPv4 address. The **first 4 bits** in **network id** stay ‘**11**0’ to define the**class** and the remaining **21 bits defines the number of block**s in class B. The number of blocks can be calculated as **221= 2,097,152.**

The **last byte** of the IPv4 address in class C defines the **host-id**. The **number of hosts** can be calculated as **28= 256**. So, we conclude that we can assign 2,097,152 blocks from class C to 2,097,152 organizations where each organization can have 256 hosts connected to the network.

2^8-2



Now, as we have calculated the number of blocks and the number of addresses in each block of class C. Let us count the total number of addresses in class C which can be calculated as follow:

As we have seen above the**first three bits** of the entire 32-bit addresses of **class C** stays ‘**110**’ to define the class. The remaining **29 bits** of entire 32-bit addresses can be changed to define the address space of class C i.e. **229= 536,870,912**.

#### Class D

223-239

Like class A, B & C, class D does **not divide** IPv4 into **net-id** and **host-id**. **All the addresses** of class D are of **one single block**. The class D addresses are designed for **multicasting**. The **first four-bit** of entire 32-bit addresses of **class D** stays ‘**1110**’ to define the class.

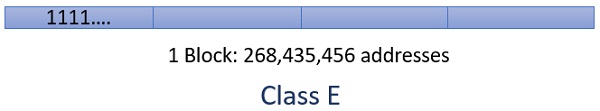
The remaining **28** bits from the 32-bit addresses of class D can be changed to define the**address space** of class D. So, the number of addresses in class D is **228=2,68,435,456.**



#### Class E

240-255

Like class D, Class E addresses are one block addresses. The addresses in class E are not split into net-id and host-id. The addresses in class E are **reserved for future** use. The **first four bits** of entire 32-bit IPv4 addresses of class E stays ‘**1111**’. The remaining **28-bit** changes to define the number of addresses in **class E** i.e. **228=2,68,435,456**.



**192.168.2.1**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **128** | **64** | **32** | **16** | **8** | **4** | **2** | **1** |
| **1** | **1** | **0** | **0** | **0** | **0** | **0** | **0** |
| **1** | **0** | **1** | **0** | **1** | **0** | **0** | **0** |
| **0** | **0** | **0** | **0** | **0** | **0** | **1** | **0** |
| **0** | **0** | **0** | **0** | **0** | **0** | **0** | **1** |

## ****Classless Addressing-****

* Classless Addressing is an improved IP Addressing system.
* It makes the allocation of IP Addresses more efficient.
* It replaces the older classful addressing system based on classes.
* It is also known as **Classless Inter Domain Routing (CIDR)**.

## ****CIDR Block-****

When a user asks for specific number of IP Addresses,

* CIDR dynamically assigns a block of IP Addresses based on certain rules.
* This block contains the required number of IP Addresses as demanded by the user.
* This block of IP Addresses is called as a **CIDR block**.

## ****Rules For Creating CIDR Block-****

A CIDR block is created based on the following 3 rules-

## ****Rule-01:****

* All the IP Addresses in the CIDR block must be contiguous.

## ****Rule-02:****

* The size of the block must be presentable as power of 2.
* Size of the block is the total number of IP Addresses contained in the block.
* Size of any CIDR block will always be in the form 21, 22, 23, 24, 25 and so on.

## ****Rule-03:****

* First IP Address of the block must be divisible by the size of the block.

**Example**

**200.10.20.40/28**

**11001000.00001010.00010100.00101000**

|  |  |
| --- | --- |
| **28** | **4** |

**11111111.11111111.11111111.11111100**

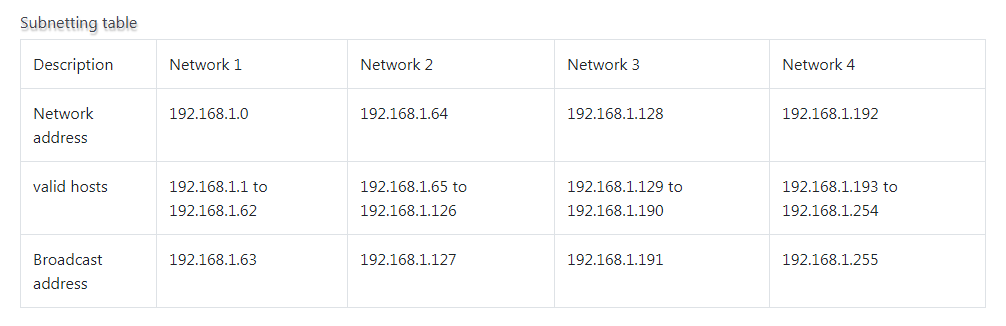
**255.255.255.252**

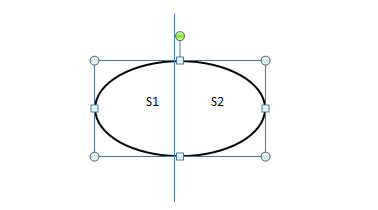
**2^2**

## What is Subnetting?

Subnetting is a process of dividing a single large network in multiple smaller networks. A single large network is just like a town without any sector and street address. In such a town, a postman may take 3 to 4 days in finding a single address. While if town is divided in sectors and streets, he can easily find any address in less than one hour.







**Router**

**Classful Addressing**

200.10.20.0 200.10.20.255

2^8=256

200.10.20.00000000………… 200.10.20.01111111

S1 0-127

1-126

200.10.20.0 200.10.20.127

S2

200.10.20.0

200.10.20.10000000………………200.10.20.11111111

S2 128-255

129-254

200.10.20.128 200.10.20.255

Class c default mask is 255.255.255.0

11111111.11111111.11111111.00000000

255.255.255.128 because reserved bit is 1

11111111.11111111.11111111.10000000

255.255.255.10000000

**Example**

200.10.20.15

200.10.20.00001111

ANDing

255.255.255.128 AND 200.10.20.15

11111111.11111111.11111111.10000000 DM

11001000.00001010.00010100.00001111 IP

11001000.00001010.00010100. 00000000

200.10.20.0

So send to the s1 part.

**Classless Addressing**

**195.10.20.128/26**

**26 bit for netid and 6 bit for hosted**

**2^6=64**

**11111111.11111111.11111111.11000000**

**255.255.255.192**

**195.10.20.10 000000**

**S1**

**195.10.20.10 000000**

**000001**

**011111**

**195.10.20.128 195.10.20.159**

**128-159**

**S2**

**195.10.20.10 100000**

**100001**

**111111**

**195.10.20.160 195.10.20.191**

**160-191**

# Supernetting

**Supernetting** is the opposite of [Subnetting](https://www.geeksforgeeks.org/ip-addressing-classless-addressing/" \t "_blank). In subnetting, a single big network is divided into multiple smaller subnetworks. In Supernetting, multiple networks are combined into a bigger network termed as a Supernetwork or Supernet.

ABCD

200.1.0.0, A

200.1.1.0, B

200.1.2.0, C

200.1.3.0 D

Build a bigger network which have a single Network Id.

**Explanation –** Before Supernetting routing table will be look like as:

| **NETWORK ID** | **SUBNET MASK** | **INTERFACE** |
| --- | --- | --- |
| 200.1.0.0 | 255.255.255.0 | A |
| 200.1.1.0 | 255.255.255.0 | B |
| 200.1.2.0 | 255.255.255.0 | C |
| 200.1.3.0 | 255.255.255.0 | D |

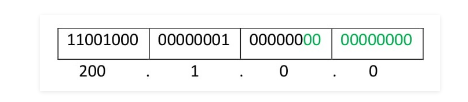
First, lets check whether three condition are satisfied or not:

1. **Contiguous:** You can easily see that all network are contiguous all having size 256 hosts.  
   Range of first Network from 200.1.0.0 to 200.1.0.255. If you add 1 in last IP address of first network that is 200.1.0.255 + 0.0.0.1, you will get the next network id that is 200.1.1.0. Similarly, check that all network are contiguous.
2. **Equal size of all network:** As all networks are of class C, so all of the have a size of 256 which in turn equal to 28.
3. **First IP address exactly divisible by total size:** When a binary number is divided by 2n then last n bits are the remainder. Hence in order to prove that first IP address is exactly divisible by while size of Supernet Network. You can check that if last n v=bits are 0 or not.

In given example first IP is 200.1.0.0 and whole size of supernet is

4\*28 = 210. If last 10 bits of first IP address are zero then IP will be divisible.

Last 10 bits of first IP address are zero (highlighted by green color). So 3rd condition is also satisfied.



200.1.00000000.00000000

200.1.00000001.00000000

200.1.00000010.00000000

200.1.00000011.00000000

11111111.1111111.11111100.00000000

255.255.252.0

200.1.0.0

Therefore, you can join all these 4 networks and can make a Supernet. New Supernet Id will be 200.1.0.0.

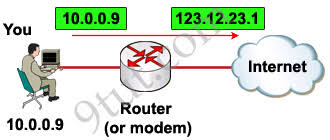
## ****NAT (Network Address Translation)****

2^128

Primarily NAT was introduced to the world of IT and networking due to the lack of IP addresses, or looking at it from another view, due to the vast amount of growing IT technology relying on IP addresses. To add to this, NAT adds a layer of security, by hiding computers, servers and other IT equipment from the outside world.

### ****How NAT works****

When computers and servers within a network communicate, they need to be identified to each other by a unique address, in which resulted in the creation of a 32 bit number, and the combinations of these 32 bits would accommodate for over 4 billion unique addresses, known as IP address. This was named IPv4, and although over 4 billion addresses sounds a lot, it really is not considering how fast the world of computers and the internet has grown.



To circumvent this problem, a temporary solution was produced known as NAT. NAT resulted in two types of IP addresses, public and private. A range of private addresses were introduced, which anyone could use, as long as these were kept private within the network and not routed on the internet. The range of private addresses known as RFC 1918 are;

Class A 10.0.0.0 - 10.255.255.255

Class B 172.16.0.0 - 172.31.255.255

Class C 192.168.0.0 - 192.168.255.255

NAT allows you to use these private IP address on the internal network. So within your private network you would assign a unique IP address to all your computers, servers and other IP driven resources, usually done via DHCP. Another company can use the same private IP addresses as well, as long as they are kept internal to their network. So two companies maybe using the same range of IP addresses but because they are private to their network, they are not conflicting with each other.

**IPV6**

1. IPv6 is the next generation Internet Protocol designed as a successor to the IP version 4.
2. IPv6 was designed to enable high performance, scalable internet.
3. This was achieved by overcoming many of the weaknesses of IPv4 protocol and by adding several new features.
4. In IPv6, there are 2128 possible ways (about 3.4 x 1038 addresses).
5. IPv6 is written in hexadecimal and consists of 8 groups, containing 4 hexadecimal digits or 8 groups of 16 bits each.
6. The IPv6 header is a static header of 40 bytes in length and has only 8 fields. Option information is carried by the extension header which is placed after the IPv6 header.
7. IPv6 has no header checksum because checksums are for example above the TCP/IP protocol suite and above the Token Ring, Ethernet etc.
8. The IPv6 header contains an 8 bit field called the Traffic Class Field. This field allows the traffic source to identify the desired delivery priority of its packets.
9. The IPv6 has both a stateful and stateless address auto-configuration mechanism.
10. IPv6 has been designed to satisfy the growing and expanded need for network security.
11. Source and destination addresses are 128 bits (16 bytes) in length.
12. Packet flow identification for QoS handling by routers is included in the IPv6 header using the Flow Label Field.
13. ARP request frames are replaced with multicast neighbour solicitation messages.
14. ICMP router discovery is replaced with ICMPv6 router solicitation and router advertisement messages are required.
15. IPv6 has three different types of addresses.

**Unicast:** A unicast address defines a single computer. A packet sent to a unicast address is delivered to that specific computer.

**Anycast:** This is a type of address that defines a group of computers with addresses which have the same prefix. A packet sent to an anycast address must be delivered to exactly one of the members of the group which is closest or the most easily accessible.

**Multicast:** A multicast address defines a group of computers which may or may not share the same prefix and may or may not be connected to the same physical network. A packet sent to a multicast address must be delivered to each member of the set.

1. **Advantages of IPv6:**

• **Larger address space:** IPv6 has 128 bit address space which is 4 times wider in bits compared to IPv4’s 32 bit address space. So there is a huge increase in the address space.

• **Better header format:** IPv6 uses a better header format. In its header format the option are separated from the base header. The options are inserted when needed between the base header and upper layer data. This helps in speeding up the routing process.

• **New options:** New options have been added in IPv6 to increase the functionality.

• **Possibility of extension:** IPv6 has been designed in such a way that there is a possibility of extension of protocol if required.

• **More security:** IPv6 includes security in the basic specification. It includes encryption of packets (ESP: Encapsulated Security Protocol) and authentication of the sender of packets (AH: Authentication Header)

• **Support to resource allocation:** To implement better support for real time traffic (such as video conference), IPv6 includes flow label in the specification. With flow label mechanism, routers can recognize to which end-to-end flow the packets belongs.

• **Plug and play:** IPv6 includes plug and play in the standard specification. It therefore must be easier for novice users to connect their machines to the network, it will be done automatically.

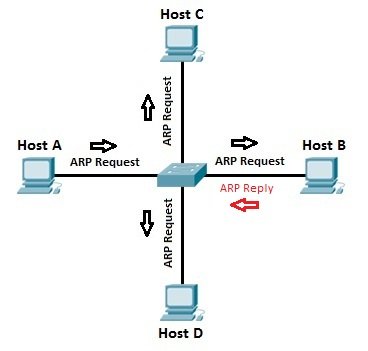
• **Clearer specification and optimization:** IPv6 follows good practices of IPv4 and rejects minor flaws/obsolete items of IPv4.

# ARP (Address Resolution Protocol)

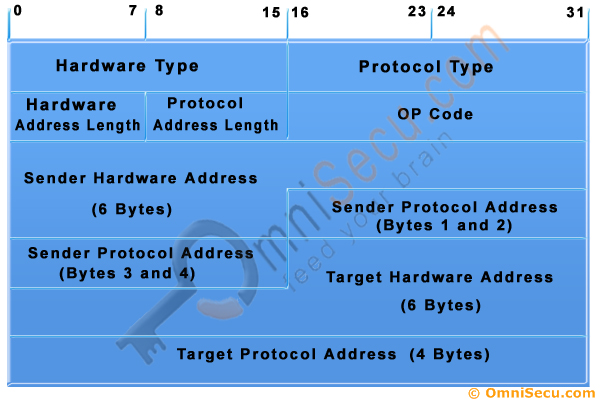
**ARP (Address Resolution Protocol)** is a network protocol used to find out the hardware (MAC) address of a device from an IP address. It is used when a device wants to communicate with some other device on a local network (for example on an Ethernet network that requires physical addresses to be known before sending packets). The sending device uses ARP to translate IP addresses to MAC addresses. The device sends an ARP request message containing the IP address of the receiving device. All devices on a local network segment see the message, but only the device that has that IP address responds with the ARP reply message containing its MAC address. The sending device now has enough information to send the packet to the receiving device.

ARP request packets are sent to the broadcast addresses (FF:FF:FF:FF:FF:FF for the Ethernet broadcasts and 255.255.255.255 for the IP broadcast).

Here is the explanation otf the ARP process:

[](https://study-ccna.com/wp-content/images/arp_process.jpg)

### Address Resolution Protocol (ARP) Message Format



**Address Resolution Protocol (ARP) Message Format**

Following are the fields in the Address Resolution Protocol (ARP) Message Format.

**Hardware Type** : Hardware Type field in the Address Resolution Protocol (ARP) Message specifies the type of hardware used for the local network transmitting the Address Resolution Protocol (ARP) message. [Ethernet](https://www.omnisecu.com/basic-networking/lan-technologies-ethernet.php) is the common Hardware Type and he value for [Ethernet](https://www.omnisecu.com/basic-networking/lan-technologies-ethernet.php) is 1. The size of this field is 2 bytes.

**Protocol Type** : Each protocol is assigned a number used in this field. [IPv4](https://www.omnisecu.com/tcpip/ipv4-protocol-and-ipv4-header.php) .

**Hardware Address Length** : Hardware Address Length in the Address Resolution Protocol (ARP) Message is length in bytes of a [hardware (MAC) address](https://www.omnisecu.com/tcpip/media-access-control-mac-addresses.php). [Ethernet MAC addresses](https://www.omnisecu.com/tcpip/media-access-control-mac-addresses.php) are 6 bytes long.

**Protocol Address Length** : Length in bytes of a [logical address (IPv4 Address)](https://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php). [IPv4 addresses](https://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php) are 4 bytes long.

**Opcode** : Opcode field in the Address Resolution Protocol (ARP) Message specifies the nature of the ARP message. 1 for ARP request and 2 for ARP reply.

**Sender Hardware Address** : Layer 2 address ([MAC Address](https://www.omnisecu.com/tcpip/media-access-control-mac-addresses.php)) of the device sending the message.

**Sender Protocol Address** : The [protocol address (IPv4 address)](https://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php) of the device sending the message

**Target Hardware Address** : Layer 2 ([MAC Address](https://www.omnisecu.com/tcpip/media-access-control-mac-addresses.php)) of the intended receiver. This field is ignored in requests.

**Target Protocol Address** : The [protocol address (IPv4 Address)](https://www.omnisecu.com/tcpip/internet-layer-ip-addresses.php) of the intended receiver.

### Reverse Address Resolution Protocol (RARP) –

Reverse ARP is a networking protocol used by a client machine in a local area network to request its Internet Protocol address (IPv4) from the gateway-router’s ARP table. The network administrator creates a table in gateway-router, which is used to map the MAC address to corresponding IP address.  
When a new machine is setup or any machine which don’t have memory to store IP address, needs an IP address for its own use. So the machine sends a RARP broadcast packet which contains its own MAC address in both sender and receiver hardware address field.

A special host configured inside the local area network, called as RARP-server is responsible to reply for these kind of broadcast packets. Now the RARP server attempt to find out the entry in IP to MAC address mapping table. If any entry matches in table, RARP server send the response packet to the requesting device along with IP address.

* LAN technologies like Ethernet, Ethernet II, Token Ring and Fiber Distributed Data Interface (FDDI) support the Address Resolution Protocol.
* RARP is not being used in today’s networks. Because we have much great featured protocols like BOOTP (Bootstrap Protocol) and DHCP( Dynamic Host Configuration Protocol).

What is the Internet Control Message Protocol (ICMP)?

The Internet Control Message Protocol is an internet layer protocol used by network devices to diagnose network communication issues. ICMP is mainly used to determine whether or not data is reaching its intended destination in a timely manner. Commonly, the ICMP protocol is used on network devices, such as routers.

What Is ICMP Used For?

The primary purpose of ICMP is for error reporting. When two devices connect over the Internet, the ICMP generates errors to share with the sending device in the event that any of the data did not get to its intended destination.

A secondary use of ICMP protocol is to perform network diagnostics; the commonly used terminal utilities traceroute and ping both operate using ICMP. The traceroute utility is used to display the routing path between two Internet devices. The routing path is the actual physical path of connected routers that a request must pass through before it reaches its destination. The journey between one router and another is known as a ‘hop’, and a traceroute also reports the time required for each hop along the way. This can be useful for determining sources of network delay.

ICMP shares error reporting and devices status by messages. Messages created by ICMP are divided into 2 categories:

**1) Error Reporting Messages**

These are messages from which the ICMP reports to errors. The list of common error reporting messages is being given below.

* **Destination Unreachable:**   
  If a router cannot locate the path for a packet, then in such an event the packet is discarded and the destination is sent to the destination unreachable message.
* **Source Quench:**   
  As you know the IP does not have the flow control. The Sending device does not have any information about whether the speed forwarded by the data forwarded by the router forwarding and processing host is according to the host. The IP discards some packets when sending speed is high. In this situation ICMP provides flow control and sends source quench messages to source device.
* **Redirect:**   
  Because routing is a dynamic process and only routers participate in it, so a host has information about only one router (gateway). So when this host sends a data, then the data will go through that router to the correct router. In this situation, gateway router redirection message will send so that the host's routing information can be updated and the host can send data directly to the correct router.
* **Time Exceeded:**   
  A router uses the routing table to forward any packet. If the routing table is not correct and there are errors, then in such an event, the packet drops only in the loop. To avoid this situation, every packet contains a time to live field. The value of this field decreases on every router. As soon as the value of this field is zero, this packet is discarded by the router. In this situation, the Time Exceeded message is sent to the router source.
* **Parameter Unintelligible:**   
  If a router or a destination host packet finds a field empty, then discards that packet and sends the parameter unintelligible message to the source.

**2) Query Messages**

These are the messages from which the ICMP queries for the status of a host. The list of common query messages is being given below with their code.

* **Echo Request & Echo reply:**   
  This pair of Query Messages is used to diagnose problems in the network. Both of these messages determine whether two hosts can communicate with each other or not.
* **Time-stamp Request & Time stamp Reply:**   
  The time it takes to travel a host from one host to another host is detected by time stamp request & reply messages.
* **Address Mask Request & Address Mask Reply:**   
  A host may be aware of its own IP address but it is not necessary that it is aware of its own subnet mask. To know your subnet mask sends the address mask request to the host router and the router sends this host's subnet mask address as the mask reply message. If the host is aware of the router's address, then he sends the request directly to the router, otherwise this request is broadcast.

**IGMP (Internet Group Management Protocol)**

Internet Group Management Protocol is a group management [protocol](https://ecomputernotes.com/computernetworkingnotes/computer-network/protocol) that mainly manages the group membership in a multicast network.

• In a multicast network, multicast routers are used to route packets to all the computers that are having membership of a particular group.

• The multicast routers use the [information](https://ecomputernotes.com/fundamental/information-technology/what-do-you-mean-by-data-and-information) from IGMP to determine which hosts are having membership of which group.

• A multicast router generally receives thousands of multicast packets that have to be transmitted to various groups. If a router has no knowledge about the group membership, it will broadcast packet to every host and this will increase the load on the network.

• In order to save the network from such a problem, a list of groups IS maintained when members of the group are present in the network.

• Thus, IGMP helps the multicast router to create and update this list.

• This protocol uses three different messages: query message, membership report and leave report.

## ****Working of IGMP****

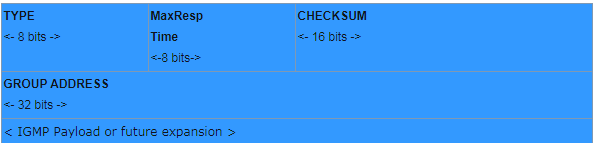
• The multicast router of the network has a list of multicast address for which the network is having any member.

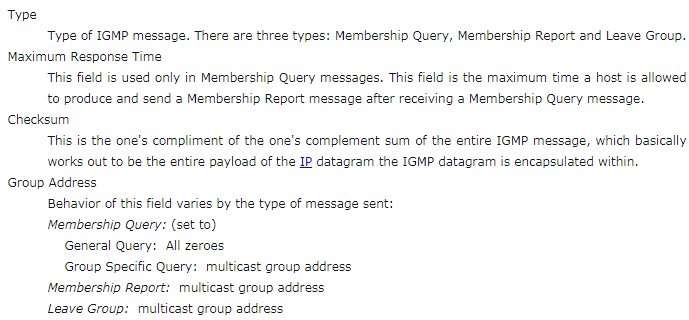
• There is one multicast router for each group that distributes multicast packet to members of that group. It means the network will have two multicast routers, if there are two multicast groups.

• A host or a multicast router can be a member of the group.

• When a host is having membership, it means that any process running on that host is a member of the group and when a router is having membership of group, it means one of the networks connected to the router is having membership of the group.

**IGMP Message format**





**Joining a Group**

• Both the host and a router can join a group.

• When a process on the host wants to join a group it sends the request to the host.

• The host adds the name of the process and group name to its list.

• If this is the first entry for that particular group, the host sends *membership report*message to the multicast router of the group.

• If it is not the first entry for the requested group there is no need of sending such a message.

**Leaving a Group**

• Whenever a host sees no process interested in a group, it sends a *leave report*message.

• The membership is not purged by the multicast router of the group, rather it immediately transmits query packets repeatedly to see if anyone still interested.

• If the response comes in the form of membership report message, the membership of the host or network is preserved.

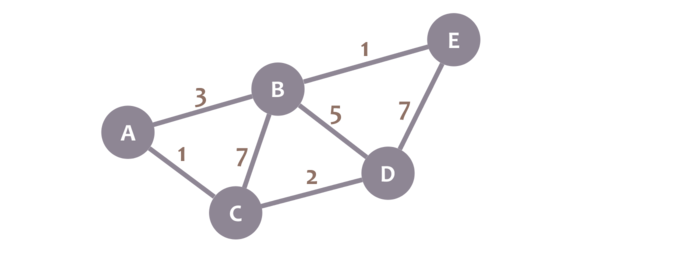
**Routing algorithm**

* In order to transfer the packets from source to the destination, the network layer must determine the best route through which packets can be transmitted.
* Whether the network layer provides datagram service or virtual circuit service, the main job of the network layer is to provide the best route. The routing protocol provides this job.
* The routing protocol is a routing algorithm that provides the best path from the source to the destination. The best path is the path that has the "least-cost path" from source to the destination.
* Routing is the process of forwarding the packets from source to the destination but the best route to send the packets is determined by the routing algorithm.

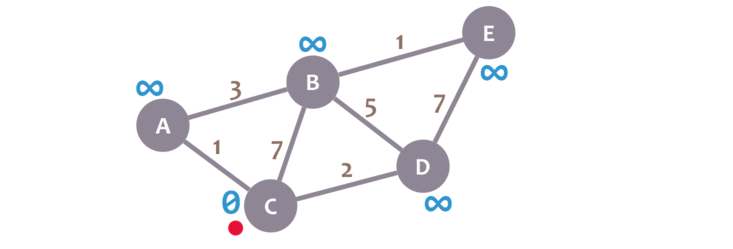
**Shortest path algorithm**

## Dijkstra's Algorithm

Dijkstra's Algorithm allows you to calculate the shortest path between one node (you pick which one) and every other node in the graph. You'll find a description of the algorithm at the end of this page, but, let's study the algorithm with an explained example! Let's calculate the shortest path between node C and the other nodes in our graph:

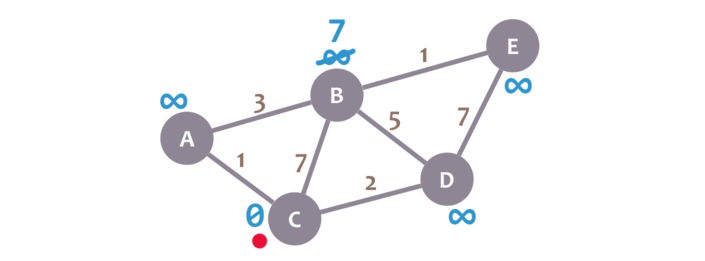


During the algorithm execution, we'll mark every node with its minimum distance to node C (our selected node). For node C, this distance is 0. For the rest of nodes, as we still don't know that minimum distance, it starts being infinity (∞):

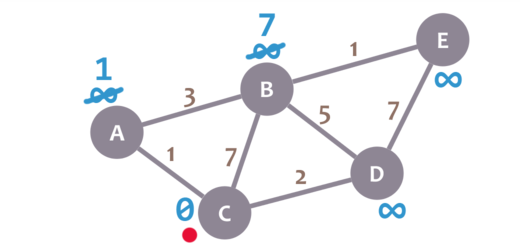


We'll also have a current node. Initially, we set it to C (our selected node). In the image, we mark the current node with a red dot.

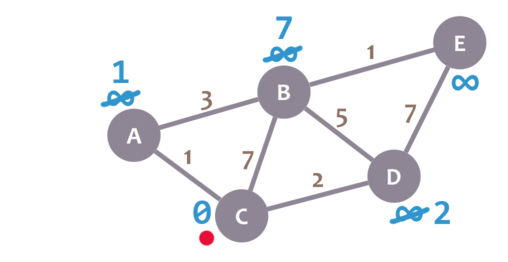
Now, we check the neighbours of our current node (A, B and D) in no specific order. Let's begin with B. We add the minimum distance of the current node (in this case, 0) with the weight of the edge that connects our current node with B (in this case, 7), and we obtain 0 + 7 = 7. We compare that value with the minimum distance of B (infinity); the lowest value is the one that remains as the minimum distance of B (in this case, 7 is less than infinity):



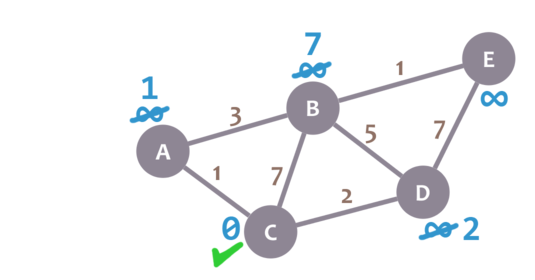
So far, so good. Now, let's check neighbour A. We add 0 (the minimum distance of C, our current node) with 1 (the weight of the edge connecting our current node with A) to obtain 1. We compare that 1 with the minimum distance of A (infinity), and leave the smallest value:



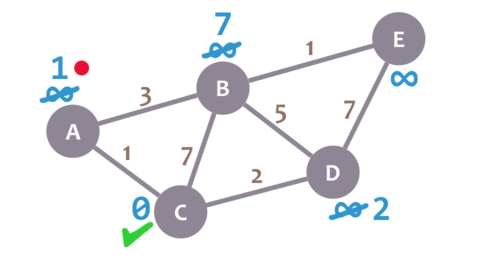
OK. Repeat the same procedure for D:



Great. We have checked all the neighbours of C. Because of that, we mark it as visited. Let's represent visited nodes with a green check mark:

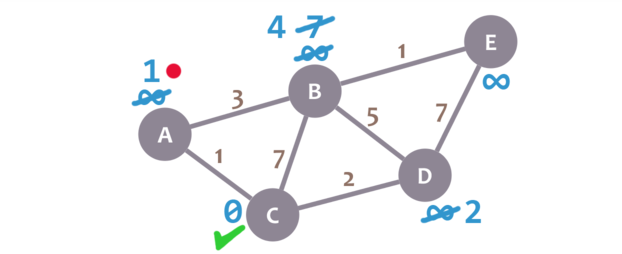


We now need to pick a new current node. That node must be the unvisited node with the smallest minimum distance (so, the node with the smallest number and no check mark). That's A. Let's mark it with the red dot:

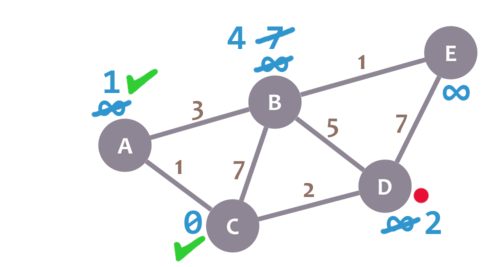


And now we repeat the algorithm. We check the neighbours of our current node, ignoring the visited nodes. This means we only check B.

For B, we add 1 (the minimum distance of A, our current node) with 3 (the weight of the edge connecting A and B) to obtain 4. We compare that 4 with the minimum distance of B (7) and leave the smallest value: 4.



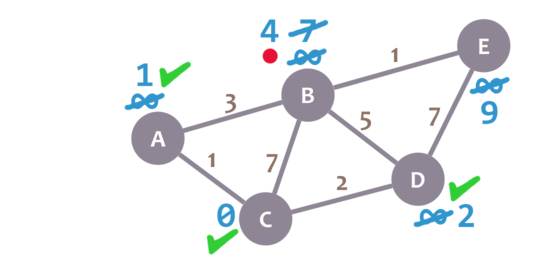
Afterwards, we mark A as visited and pick a new current node: D, which is the non-visited node with the smallest current distance.



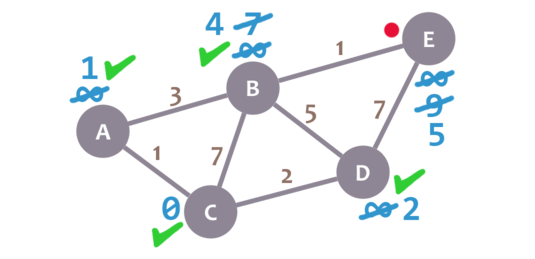
We repeat the algorithm again. This time, we check B and E.

For B, we obtain 2 + 5 = 7. We compare that value with B's minimum distance (4) and leave the smallest value (4). For E, we obtain 2 + 7 = 9, compare it with the minimum distance of E (infinity) and leave the smallest one (9).

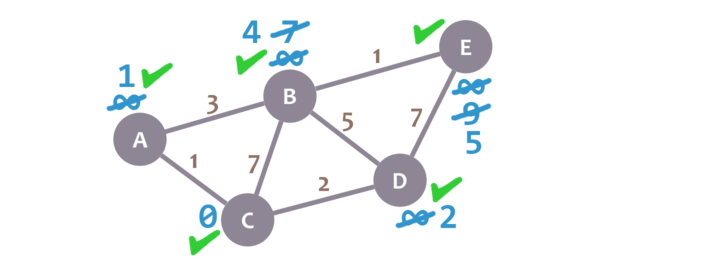
We mark D as visited and set our current node to B.



Almost there. We only need to check E. 4 + 1 = 5, which is less than E's minimum distance (9), so we leave the 5. Then, we mark B as visited and set E as the current node.



E doesn't have any non-visited neighbours, so we don't need to check anything. We mark it as visited.



As there are not univisited nodes, we're done! The minimum distance of each node now actually represents the minimum distance from that node to node C (the node we picked as our initial node)!

C-----A=1

C-----D=2

C----A----B=4

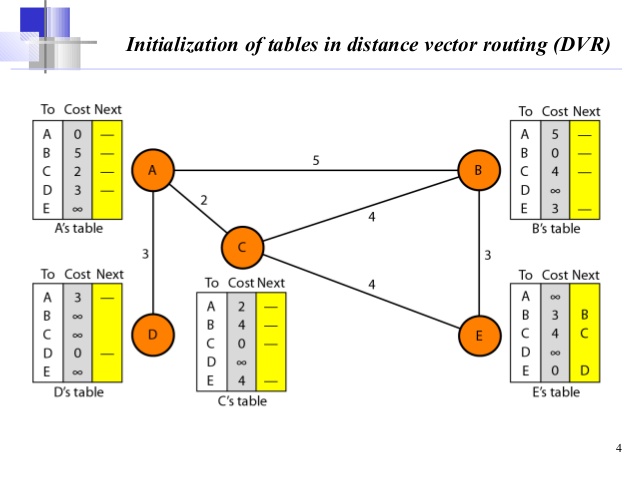
C----A---B----E=5

# Distance Vector Routing Algorithm

* **The Distance vector algorithm is iterative, asynchronous and distributed.**
  + **Distributed:** It is distributed in that each node receives information from one or more of its directly attached neighbors performs calculation and then distributes the result back to its neighbors.
  + **Iterative:** It is iterative in that its process continues until no more information is available to be exchanged between neighbors.
  + **Asynchronous:** It does not require that all of its nodes operate in the lock step with each other.
* The Distance vector algorithm is a dynamic algorithm.
* It is mainly used in ARPANET, and RIP.
* Each router maintains a distance table known as **Vector**.

### Three Keys to understand the working of Distance Vector Routing Algorithm:

* **Knowledge about the whole network:** Each router shares its knowledge through the entire network. The Router sends its collected knowledge about the network to its neighbors.
* **Routing only to neighbors:** The router sends its knowledge about the network to only those routers which have direct links. The router sends whatever it has about the network through the ports. The information is received by the router and uses the information to update its own routing table.
* **Information sharing at regular intervals:** Within 30 seconds, the router sends the information to the neighboring routers.

****

# Link State Routing

Link state routing is a technique in which each router shares the knowledge of its neighborhood with every other router in the internetwork.

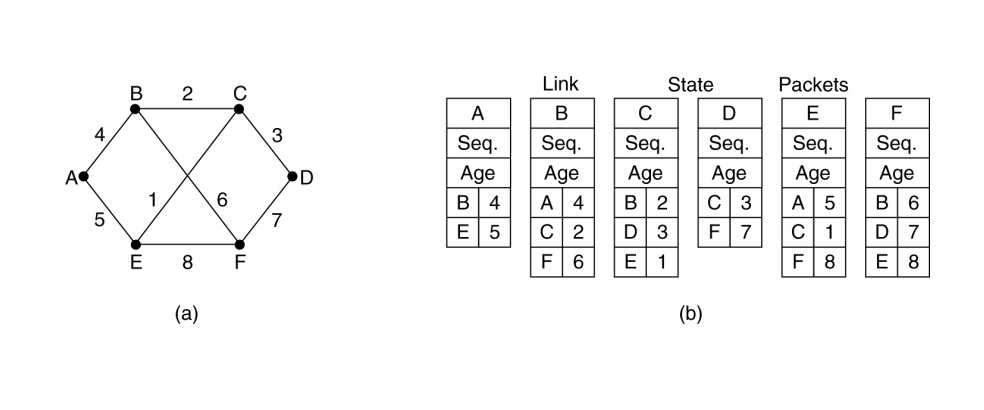
**The three keys to understand the Link State Routing algorithm:**

* **Knowledge about the neighborhood:** Instead of sending its routing table, a router sends the information about its neighborhood only. A router broadcast its identities and cost of the directly attached links to other routers.
* **Flooding:** Each router sends the information to every other router on the internetwork except its neighbors. This process is known as Flooding. Every router that receives the packet sends the copies to all its neighbors. Finally, each and every router receives a copy of the same information.
* **Information sharing:** A router sends the information to every other router only when the change occurs in the information.

## Link State Routing has two phases:

### Reliable Flooding

* **Initial state:** Each node knows the cost of its neighbors.
* **Final state:** Each node knows the entire graph.



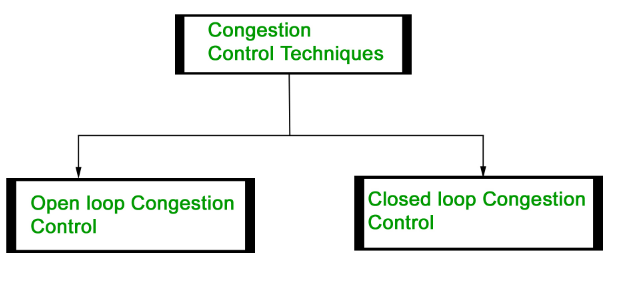
**A---B---C—D 4+2+3=9**

**A---E---F 5+8=13**

# Congestion Control techniques in Computer Networks

### Open Loop Congestion Control

Open loop congestion control policies are applied to prevent congestion before it happens. The congestion control is handled either by the source or the destination.



**Policies adopted by open loop congestion control –**

1. **Retransmission Policy:**

It is the policy in which retransmission of the packets are taken care. If the sender feels that a sent packet is lost or corrupted, the packet needs to be retransmitted. This transmission may increase the congestion in the network.  
To prevent congestion, retransmission timers must be designed to prevent congestion and also able to optimize efficiency.

1. **Window Policy:**

The type of window at the sender side may also affect the congestion. Several packets in the Go-back-n window are resent, although some packets may be received successfully at the receiver side. This duplication may increase the congestion in the network and making it worse.  
Therefore, Selective repeat window should be adopted as it sends the specific packet that may have been lost.

1. **Discarding Policy:**

A good discarding policy adopted by the routers is that the routers may prevent congestion and at the same time partially discards the corrupted or less sensitive package and also able to maintain the quality of a message.  
In case of audio file transmission, routers can discard less sensitive packets to prevent congestion and also maintain the quality of the audio file.

1. **Acknowledgment Policy:**

Since acknowledgement is also the part of the load in network, the acknowledgment policy imposed by the receiver may also affect congestion. Several approaches can be used to prevent congestion related to acknowledgment.  
The receiver should send acknowledgement for N packets rather than sending acknowledgement for a single packet. The receiver should send an acknowledgment only if it has to send a packet or a timer expires.

1. **Admission Policy:**

In admission policy a mechanism should be used to prevent congestion. Switches in a flow should first check the resource requirement of a network flow before transmitting it further. If there is a chance of congestion or there is congestion in the network, router should deny establishing a virtual network connection to prevent further congestion.

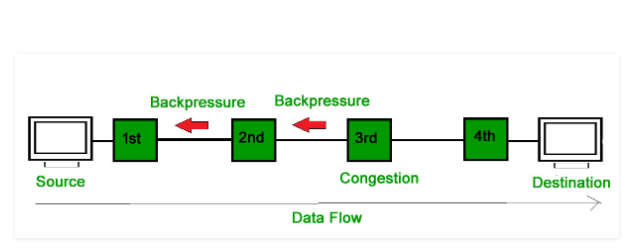
All the above policies are adopted to prevent congestion before it happens in the network.

### Closed Loop Congestion Control

Closed loop congestion control technique is used to treat or alleviate congestion after it happens. Several techniques are used by different protocols; some of them are:

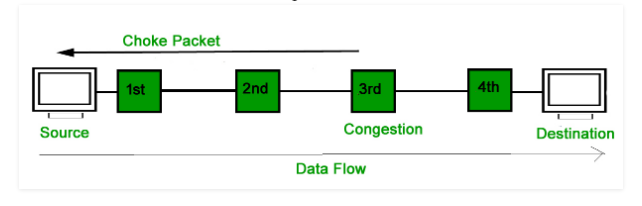
1. **Backpressure:**

Backpressure is a technique in which a congested node stops receiving packet from upstream node. This may cause the upstream node or nodes to become congested and rejects receiving data from above nodes. Backpressure is a node-to-node congestion control technique that propagates in the opposite direction of data flow. The backpressure technique can be applied only to virtual circuit where each node has information of its above upstream node.



In above diagram the 3rd node is congested and stops receiving packets as a result 2nd node may be get congested due to slowing down of the output data flow. Similarly 1st node may get congested and informs the source to slow down.

1. **Choke Packet Technique:**

Choke packet technique is applicable to both virtual networks as well as datagram subnets. A choke packet is a packet sent by a node to the source to inform it of congestion. Each router monitor its resources and the utilization at each of its output lines. Whenever the resource utilization exceeds the threshold value which is set by the administrator, the router directly sends a choke packet to the source giving it a feedback to reduce the traffic. The intermediate nodes through which the packets has traveled are not warned congestion.  


1. **Implicit Signaling:**

In implicit signaling, there is no communication between the congested nodes and the source. The source guesses that there is congestion in a network. For example when sender sends several packets and there is no acknowledgment for a while, one assumption is that there is congestion.

1. **Explicit Signaling:**

In explicit signaling, if a node experiences congestion it can explicitly sends a packet to the source or destination to inform about congestion. The difference between choke packet and explicit signaling is that the signal is included in the packets that carry data rather than creating different packet as in case of choke packet technique.

**Quality of Services are given below:**  
  
**1. Reliability**  
If a packet gets lost or acknowledgement is not received (at sender), the re-transmission of data will be needed. This decreases the reliability.  
The importance of the reliability can differ according to the application.  
**For example:**  
E- mail and file transfer need to have a reliable transmission as compared to that of an audio conferencing.  
  
**2. Delay**  
Delay of a message from source to destination is a very important characteristic. However, delay can be tolerated differently by the different applications.  
**For example:**  
The time delay cannot be tolerated in audio conferencing (needs a minimum time delay), while the time delay in the e-mail or file transfer has less importance.  
  
**3. Jitter**  
The jitter is the variation in the packet delay.  
If the difference between delays is large, then it is called as **high jitter.** On the contrary, if the difference between delays is small, it is known as **low jitter.**  
**Example:**  
**Case1:** If 3 packets are sent at times 0, 1, 2 and received at 10, 11, 12. Here, the delay is same for all packets and it is acceptable for the telephonic conversation.  
**Case2:** If 3 packets 0, 1, 2 are sent and received at 31, 34, 39, so the delay is different for all packets. In this case, the time delay is not acceptable for the telephonic conversation.  
  
**4. Bandwidth**  
Different applications need the different bandwidth.  
**For example:**  
Video conferencing needs more bandwidth in comparison to that of sending an e-mail.

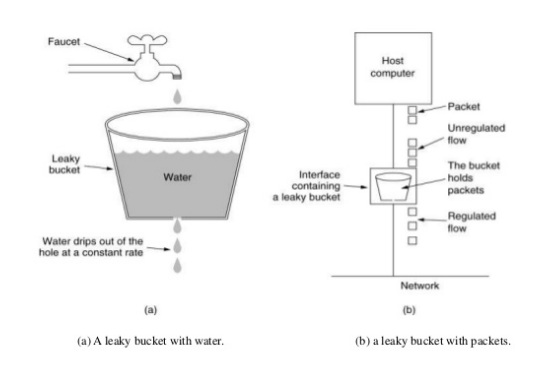
In the network layer, before the network can make Quality of service guarantees, it must know what traffic is being guaranteed. One of the main causes of congestion is that traffic is often busty.

To understand this concept first we have to know little about traffic shaping. Traffic Shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Approach of congestion management is called Traffic shaping. Traffic shaping helps to regulate rate of data transmission and reduces congestion.

**Leaky Bucket Algorithm**

Let us consider an example to understand

Imagine a bucket with a small hole in the bottom. No matter at what rate water enters the bucket, the outflow is at constant rate. When the bucket is full with water additional water entering spills over the sides and is lost.



Similarly, each network interface contains a leaky bucket and the following steps are involved in leaky bucket algorithm:

* When host wants to send packet, packet is thrown into the bucket.
* The bucket leaks at a constant rate, meaning the network interface transmits packets at a constant rate.
* Busty traffic is converted to a uniform traffic by the leaky bucket.
* In practice the bucket is a finite queue that outputs at a finite rate.

**Token bucket Algorithm**

The leaky bucket algorithm enforces output pattern at the average rate, no matter how busty the traffic is. So in order to deal with the busty traffic we need a flexible algorithm so that the data is not lost. One such algorithm is token bucket algorithm.

**Token Bucket Algorithm:**

* Host is connected to the network by an interface. This interface is actually a bucket. A token is generated in the bucket every ∆T seconds.
* The host sends an unregulated flow to the bucket. For a packet to be transmitted to the network, it must capture and destroy a token present in the bucket.
* If the lost is not sending packets to the bucket the tokens keep getting accumulated in the bucket. Generally there is a maximum amount of tokens that can be accumulated in the bucket.
* Due to this feature of tokens getting accumulated, bursts can be handled better. Therefore in this case the rate increases if tokens are saved in the bucket, whereas in leaky bucket the rate will always be constant (1 packet per clock tic).

[](https://tutorialspoint.dev/image/leakybuk.jpg)