MODULE 3..

Network Layer

1. Network layer design issues
2. LOGICAL ADDRESSING of IPV4 &IPV6
3. Address Mapping
4. Routing Algorithm
5. Congestion Control Algorithm
6. Internetworking: The network layers of internet
7. Quality of Service

The Network layer 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: Maintains the address at the frame header of both source and destination and performs addressing to detect various devices in the network.
- Packeting: This is performed by Internet Protocol. The network layer converts the packets from its upper layer.
- 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 that 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 the Transport Layer

Through the network/transport layer interface, the network layer transfers its patterns services to the transport layer. These services are described below. But before providing these services to the transfer layer, the following goals must be kept in mind:-

- Offering services must not depend on router technology.
- The transport layer needs to be protected from the type, number, and topology of the available router.

- The network addresses for the transport layer should use uniform numbering patterns, also at LAN and WAN connections.

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

- Connectionless - The routing and insertion of packets into the subnet are 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

Packets are termed as "datagrams" and corresponding subnets 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 the router via. a few protocols. Each data packet has a 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 the 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.

## LOGICAL ADDRESSING

Usually, computers communicate through the Internet. The packet transmitted by the sending computer may pass through several LANs or WANs before reaching the destination computer.

For this level of communication, we need a global addressing scheme called

logical addressing.

The Internet addresses are 32 bits in length; this gives us a maximum of 232 addresses. These addresses are referred to as IPv4 (IP version 4) addresses or simply IP addresses. The new generation of IP or IPv6 (IP version 6) can accommodate more addresses. In this version, the Internet uses 128-bit addresses that give much greater flexibility in address allocation.

3.1.1. IPV4 ADDRESSES

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device to the Internet.

IPv4 addresses are unique. They are unique, which means two devices on the Internet can never have the same address at the same time.

The IPv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet..

Address Space A protocol such as IPv4 that defines addresses has an address space. An address space is the total number of addresses used by the protocol. If a protocol uses N bits to define an address, the address space is 2 N because each bit can have two different values (0 or 1) and N bits can have 2N values.

IPv4 uses 32-bit addresses, which means that the address space is 232 or 4,294,967,296 (more than 4 billion). This means that, if there were no restrictions, more than 4 billion devices could be connected to the Internet.

**Notations**

There are two notations to show an IPv4 address: **binary notation and dotted-decimal** notation..

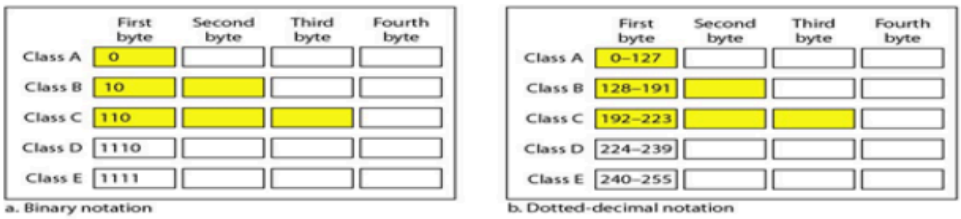
Binary Notation In binary notation, the IPv4addressis displayed as32 bits or a 4 -byte address.

The following is an example of an IPv4 address in binary notation: 01110001 10010001 00011111 00000110

Dotted-Decimal Notation Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. The following is the dotted-decimal notation of the above address: 113.145.31.6 Because each byte (octet) is 8 bits, each number in dotted-decimal notation is a value ranging from 0 to 255.

**Classful Addressing** In classful addressing, the address space is divided into five classes: A, B, C, D, and E.

**Figure 3.1: Classes in binary and dotted decimal notation**




**Class D: multicast**
**Class E: reserved**

---

**Classes and Blocks**

One problem with classful addressing is that each class is divided into a fixed number of blocks with each block having a fixed size as shown in Table below.

**Table 3.1: Number of blocks and block size in classful IPv4 addressing**

| Class | Number of Blocks | Block Size | Application |
|---|---|---|---|
| A | 128 | 16,777,216 | Unicast |
| B | 16,384 | 65,536 | Unicast |
| C | 2,097,152 | 256 | Unicast |
| D | 1 | 268,435,456 | Multicast |
| E | 1 | 268,455,456 | Reserved |

Previously, when an organization requested a block of addresses, it was granted one in class A, B, or C. Class A addresses were designed for large organizations with a large number of attached hosts or routers. Class A block is too large for any organization.

Class B addresses was designed for midsize organizations with tens of thousands of attached hosts or routers. A block in class B is also very large and resulted in wasted addresses.

Class C addresses were designed for small organizations with a small number of attached hosts or routers. A block in class C is probably too small for many organizations.

Class D addresses were designed for multicasting. Each address in this class is used to define one group of hosts on the Internet. The Internet authorities wrongly predicted a need for 268,435,456 groups. This never happened and many addresses were wasted here too.

And lastly, the class E addresses were reserved for future use; only a few were used, resulting in another waste of addresses.

### Netid and Hostid

In classful addressing, an IP address in class A, B, or C is divided into netid and hostid. These parts are of varying lengths, depending on the class of the address. In Figure 3.1the netid is in color, the hostid is in white. This concept does not apply to classes D and E.

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

### Sub-netting

If an organization was granted a large block in class A or B, it could divide the addresses into several contiguous groups and assign each group to smaller networks called sub-nets or, in some cases, share part of the addresses with neighbors. Sub-netting increases the number of 1s in the mask.

### Super-netting

The class A and class B addresses were meant for very large and large organizations. There was a huge demand for midsize blocks. The size of a class C block with a maximum number of 256 addresses did not satisfy the needs of most organizations. Even a midsize organization needed more addresses. One solution was super-netting. In super-netting, an organization can combine several class C blocks to create a larger range of addresses. In other words, several networks are combined to create a super-network or a super net. For example, an organization that needs 1000 addresses can be granted four contiguous class C blocks. The organization can then use these addresses to create one super-network. Super-netting decreases the number of 1s in the mask, For example, if an organization is given four class C addresses, the mask changes from /24 to /22.

### Classless Addressing

To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented. In this scheme, there are no classes, but the addresses are still granted in blocks,

## 3.1.3. IPV6 ADDRESSES

Despite all short-term solutions, such as classless addressing, Dynamic Host Configuration Protocol (DHCP), and NAT, address depletion is still a long-term problem for the Internet. This and other problems in the IP protocol itself have been the motivation for IPv6.

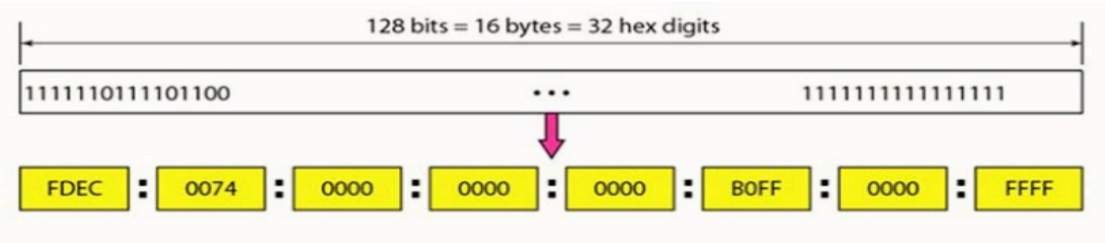
---

An IPv6 address consists of 16 bytes (octets); it is 128 bits long.

### Hexadecimal Colon Notation

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

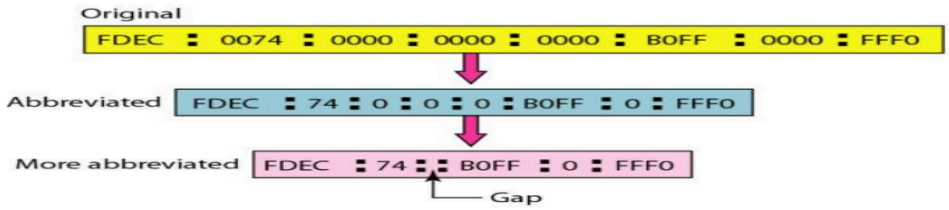
**Figure 3.3: Address in binary and hexadecimal colon notation**



*Abbreviation*

In the IP address, sometimes, many of the digits are zeros. In this case, we can abbreviate the address. The leading zeros of a section (four digits between two colons) can be omitted. Only the leading zeros can be dropped, not the trailing zeros.

**Figure 3.4: Abbreviated Ipv6 addresses**




*Re expansion of the abbreviated address is very simple:* Align the unabbreviated portions and insert zeros to get the original expanded address.

**Address Space**

IPv6 has a much larger address space; $2^{128}$ addresses are available. IPv6 address is divided into several categories. A few leftmost bits, called the *type prefix,* in each address define its category. The type prefix is variable in length, but it is designed such that no code is identical to the first part of any other code. When an address is given, the type prefix can easily be determined. Table 3.3 shows the prefix for each type of address. The third column shows the fraction of each type of address relative to the whole address space.

| IPV4 | IPV6 |
| --- | --- |
| Addresses are 32 bits (4 bytes) in length. | Addresses are 128 bits (16 bytes) in length |
| Address (A) resource records in DNS to map host names to IPv4 addresses. | Address (AAAA) resource records in DNS to map host names to IPv6 addresses. |
| Pointer(PTR)resource records in the IN-ADDR.ARPA DNS domain to map IPv4 addresses to host names. | Pointer(PTR)resource records in the IP6.ARPA DNS domain to map IPv6 addresses to host names. |
| IPSec is optional and should be supported externally | IPSec support is not optional |
| Header does not identify packet flow or QoS handling by routers. | Header contains Flow Label field, which Identifies packet flow for QoS handling by router. |
| Both routers and the sending host fragment packets. | Routers do not support packet fragmentation. Sending host fragments packets |
| Header includes a checksum. | Header does not include a checksum. |
| Header includes options. | IPV6 Optional data is supported as extension headers. |
| ARP uses broadcast ARP request to resolve IP to MAC/Hardware address | Multicast Neighbor Solicitation messages resolve IP addresses to MAC addresses. |
| Internet Group Management Protocol (IGMP) manages membership in local subnet groups. | Multicast Listener Discovery (MLD) messages manage membership in local subnet groups. |
| Broadcast addresses are used to send traffic to all nodes on a subnet. | IPv6 uses a link-local scope all-nodes multicast address. |
| Must support a 576-byte packet size (possibly fragmented) | Must support a 1280-byte packet size (without fragmentation) |
| Configured either manually or through DHCP. | Does not require manual configuration or DHCP. |

## 3.2. INTER NETWORKING

Inter networking is the practice of connecting a computer network with other networks through the use of gateways that provide a common method of routing information packets between the networks. The resulting systems of interconnected networks are called an **inter-network**, or simply an internet.

The physical and data link layers of a network operate locally. These two layers are jointly responsible for data delivery on the network from one node to the next. . The frame (in data link or physical layer) does not carry any routing information.

The network layer is responsible for host-to-host delivery and for routing the packets through the routers or switches.

- The network layer at the source is responsible for creating a packet from the data coming from another protocol (such as a transport layer protocol or a routing protocol). The header of the packet contains, among other information, the logical addresses of the source and destination. The network layer is responsible for checking its routing table to find the routing information (such as the outgoing interface of the packet or the physical address of the next node).
- The network layer at the switch or router is responsible for routing the packet.
- The network layer at the destination is responsible for address verification; it makes sure that the destination address on the packet is the same as the address of the host.

### Internet as a Datagram Network

The Internet, at the network layer, is a packet-switched network. In general, switching can be divided into three broad categories: circuit switching, packet switching, and message switching. Packet switching uses either the virtual circuit approach or the datagram approach.

The Internet uses the datagram approach to switching in the network layer. It uses the universal addresses defined in the network layer to route packets from the source to the destination.

### Internet as a Connectionless Network

Delivery of a packet can be accomplished by using either a connection—oriented or a connectionless network service. In a connection-oriented service, the source first makes a connection with the destination before sending a packet. When the connection is established, a sequence of packets from the same source to the same destination can be sent one after another. When all packets of a message have been delivered, the connection is terminated.

In connectionless service, the network layer protocol treats each packet independently, with each packet having no relationship to any other packet. The packets in a message may or may not travel the same path to their destination. This type of service is used in the datagram approach to packet switching. The Internet has chosen this type of service at the network layer.

The reason for this decision is that the Internet is made of so many heterogeneous networks that it is almost impossible to create a connection from the source to the destination without knowing the nature of the networks in advance.

# 3.4. ADDRESS MAPPING

## 3.4.1. MAPPING LOGICAL TO PHYSICAL ADDRESS: ADDRESS RESOLUTION PROTOCOL .

A host or a router has an IP datagram to send to another host or router, it has the logical (IP) address of the receiver .

The logical (IP) address is obtained from the DNS if the sender is the host or it is found in a routing table if the sender is a router.

• The IP datagram must be an encapsulated in a frame to be able to pass through the physical network. This means that sender needs the physical address of the receiver.

• The host or the router sends an ARP query packet. The packet includes the physical and IP address of the sender and the IP address of the receiver. .

• Because the sender does not know the physical address of the receiver, the query is broadcast over the network.

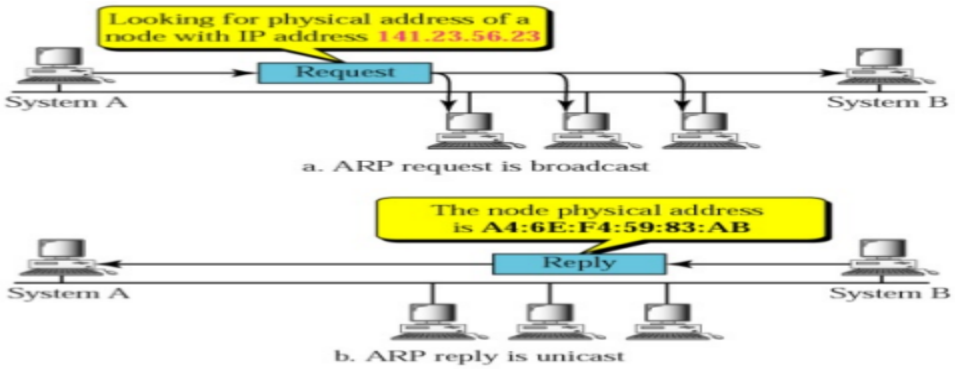
• Every host or router on the network receives and processes the ARP query packet, but only the intended recipient recognizes its IP address and sends back an ARP response packet.

• The response packet contains the recipient's IP and physical addresses.

• The packet is unicast directly to the inquirer by using the physical address received in the query packet.

In the below fig.(a) the system on the left(A) has a packet that needs to be delivered to another system (B) with IP address 141.23.56.23. System A needs to pass the packet to its data link layer for the actual delivery, but it does not know the physical address of the recipient. It uses the services of ARP by asking the ARP protocol to send a broadcast ARP request packet to ask for the physical address of a system with an IP address of 141.23.56.23. .

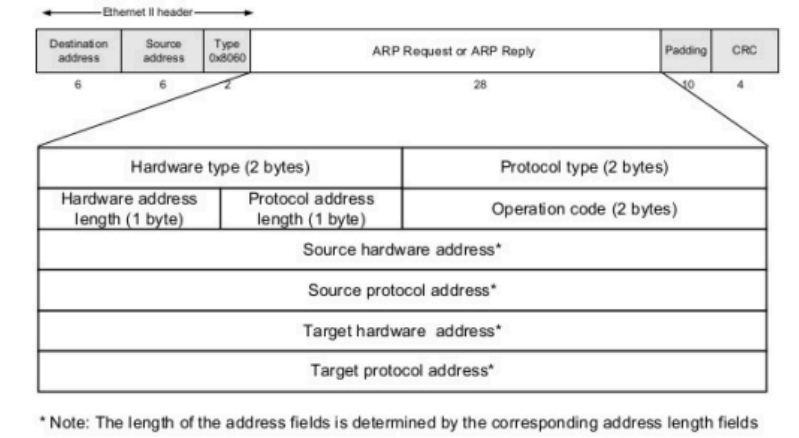
**Figure: 3.6 ARP operation**




## Packet Format

Figure below shows the format of an ARP packet.

**Figure: 3.7 ARP packet format**




* Note: The length of the address fields is determined by the corresponding address length fields

The fields are as follows:

- **Hardware type.** This is a 16-bit field defining the type of the network on which ARP is running. Each LAN has been assigned an integer based on its type. For example, Ethernet is given type 1.
- **Protocol type.** This is a 16-bit field defining the protocol. For example, the value of this field for the IPv4 protocol is $0800_{16}$.
- **Hardware length.** This is an 8-bit field defining the length of the physical address in bytes. For example, for Ethernet the value is 6.
- **Protocol length.** This is an 8-bit field defining the length of the logical address in bytes. For example, for the IPv4 protocol the value is 4.
- **Operation.** This is a 16-bit field defining the type of packet. Two packet types are defined: ARP request (1) and ARP reply (2).
- **Sender hardware address.** This is a variable-length field defining the physical address of the sender. For example, for Ethernet this field is 6 bytes long.
- **Sender protocol address.** This is a variable-length field defining the logical (for example, IP) address of the sender. For the IP protocol, this field is 4 bytes long.
- **Target hardware address.** This is a variable-length field defining the physical address of the target. For example, for Ethernet this field is 6 bytes long. For an ARP request message, this field is all 0s because the sender does not know the physical address of the target.
- **Target protocol address.** This is a variable length field defining the logical (for example, IP) address of the target. For the IPv4 protocol, this field is 4 bytes long.

### 3.4.2. MAPPING PHYSICAL TO LOGICAL ADDRESS: RARP, BOOTP, AND DHCP

There are occasions in which a host knows its physical address, but needs to know its logical address. This may happen in two cases:

- A diskless station is just booted. The station can find its physical address by checking its interface, but it does not know its IP address.
- An organization does not have enough IP addresses to assign to each station; it needs to assign IP addresses on demand. The station can send its physical address and ask for a short time lease.

**Reverse Address Resolution Protocol (RARP)** finds the logical address for a machine that knows only its physical address. Each host or router is assigned one or more logical (IP) addresses, which are unique and independent of the physical (hardware) address of the machine. To create an IP datagram, a host or a router needs to know its own IP address or addresses. The IP address of a machine is usually read from its configuration file stored on a disk file.

The machine can get its physical address which is unique locally. It can then use the physical address to get the logical address by using the RARP protocol. A RARP request is created and broadcast on the local network. Another machine on the local network that knows all the IP addresses will respond with a RARP reply. The requesting machine must be running a RARP client program; the responding machine must be running a RARP server program.

There is a serious problem with RARP: Broadcasting is done at the data link layer. The physical broadcast address, all 1s in the case of Ethernet, does not pass the boundaries of a network. This means that if an administrator has several networks or several subnets, it needs to assign a RARP server for each network or subnet. This is the reason that RARP is almost obsolete. Two protocols, BOOTP and DHCP, are replacing RARP.

**BOOTP**

The Bootstrap Protocol (BOOTP) is a client/server protocol designed to provide physical address to logical address mapping. BOOTP is an application layer protocol. The administrator may put the client on the same network or on different networks. BOOTP messages are encapsulated in a UDP packet, and the UDP packet itself is encapsulated in an IP packet. The client simply uses all 0s as the source address and all 1s as the destination address.

**DHCP**

BOOTP is not a **dynamic configuration protocol.** When a client requests its IP address, the BOOTP server consults a table that matches the physical address of the client with its IP address. This implies that the binding between the physical address and the IP address of the client already exists. The binding is predetermined.

The **Dynamic Host Configuration Protocol (DHCP)** has been devised to provide static and dynamic address allocation that can be manual or automatic.

**Static** Address Allocation In this capacity DHCP acts as BOOTP does. It is backward- compatible with BOOTP, which means a host running the BOOTP client can request a static address from a DHCP server. A DHCP server has a database that statically binds physical addresses to IP addresses.

Dynamic Address Allocation DHCP has a second database with a pool of available IP addresses. This second database makes DHCP dynamic. When a DHCP client requests a temporary IP address, the DHCP server goes to the pool of available (unused) IP addresses and assigns an IP address for a negotiable period of time.

# ROUTING ALGORITHM:: DISTANCE VECTOR AND LINK STATE ROUTING

## Distance Vector

The Distance-Vector (DV) Routing Algorithm

- Distance-vector (DV) algorithm is iterative, asynchronous, and distributed.
- It is distributed in that each node receives some information from one or more of its directly attached neighbour, performs a calculation, and then distributes the results of its calculation back to its neighbour.
- It is iterative in that this process continues on until no more information is exchanged between neighbors.
- The algorithm is asynchronous in that it does not require all of the nodes to operate in lockstep with each other.

FOR EXAMPLE REFER CLASS NOTES

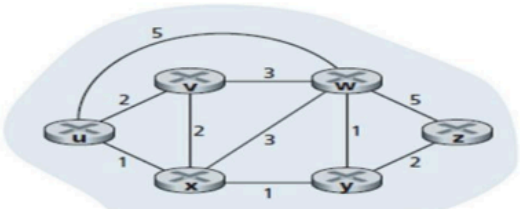
**The Link-State (LS) Routing Algorithm (Dijkstra's algorithm)**

- Dijkstra's algorithm computes the least-cost path from one node (the source, which we will refer to as u) to all other nodes in the network.
- Dijkstra's algorithm is iterative and has the property that after the $k^{th}$ iteration of the algorithm, the least-cost paths are known to k destination nodes, and among the least-cost paths to all destination nodes, these k paths will have the k smallest costs.
- Let us define the following notation:
  - D(v): cost of the least-cost path from the source node to destination v as of this iteration of the algorithm.
  - p(v): previous node (neighbor of v) along the current least-cost path from the source to v.

---

  - N' : subset of nodes; v is in N' if the least-cost path from the source to v is definitively known.
- The global routing algorithm consists of an initialization step followed by a loop.
- The number of times the loop is executed is equal to the number of nodes in the network.
- Upon termination, the algorithm will have calculated the shortest paths from the source node u to every other node in the network.
- As an example, let's consider the network in Figure3.16 and compute the least-cost paths from u to all possible destinations.

**Figure: 3.16 Abstract graph model of a computer network**



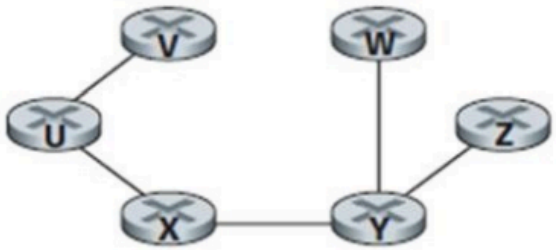

```
1  Initialization:
2    N' = {u}
3    for all nodes v
4      if v is a neighbor of u
5        then D(v) = c(u,v)
6      else D(v) = ∞
7
8  Loop
9    find w not in N' such that D(w) is a minimum
10   add w to N'
11   update D(v) for each neighbor v of w and not in N':
12         D(v) = min( D(v), D(w) + c(w,v) )
13   /* new cost to v is either old cost to v or known
14    least path cost to w plus cost from w to v */
15 until N'= N
```

- Figure3.16. Shows the resulting least-cost paths for u for the network in Figure3.17 .

| step | N' | D(v),p(v) | D(w),p(w) | D(x),p(x) | D(y),p(y) | D(z),p(z) |
|---|---|---|---|---|---|---|
| 0 | u | 2,u | 5,u | 1,u | ∞ | ∞ |
| 1 | ux | 2,u | 4,x | | 2,x | ∞ |
| 2 | uxy | 2,u | 3,y | | | 4,y |
| 3 | uxyv | | 3,y | | | 4,y |
| 4 | uxyvw | | | | | 4,y |
| 5 | uxyvwz | | | | | |

**Table: 3.5 Running the link-state algorithm on the network in Figure**

**Figure:3.17 Least cost path for module u**




## CONGESTION CONTROL ALGORITHM

Congestion control refers to techniques and mechanisms in computer networks that prevent or manage congestion, which occurs when too much data is sent, causing delays and packet loss

TWO TYPES OF ALGORITHM

- LEAKY BUCKET ALGORITHM
- TOKEN BUCKET ALGORITHM

LEAKY BUCKET ALGORITHM

The leaky bucket algorithm is a network traffic shaping method that smooths out bursty traffic by sending packets out at a constant rate, similar to how a leaky bucket drips water at a steady pace. It uses a fixed-size buffer (the bucket) to store incoming packets. If the buffer is not full, packets are added; if the bucket's leak rate is slower than the arrival rate, the excess packets are either discarded or delayed when the bucket overflows

How it works

- **Input**:

Bursty data packets arrive at the network interface at varying rates.

- **Buffer**:

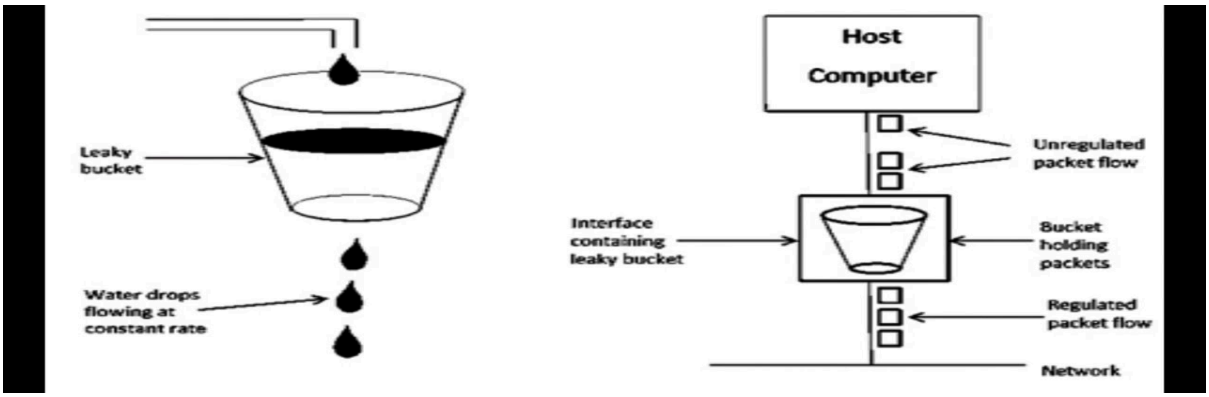
These packets are placed into a buffer (the "bucket"), which has a finite size.

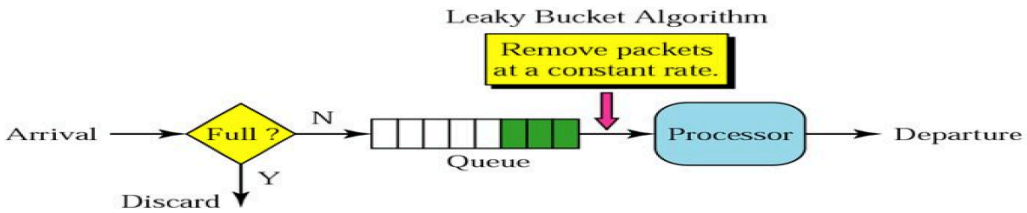
- **Output**:

The algorithm releases packets from the buffer at a constant, steady rate, regardless of the input rate.

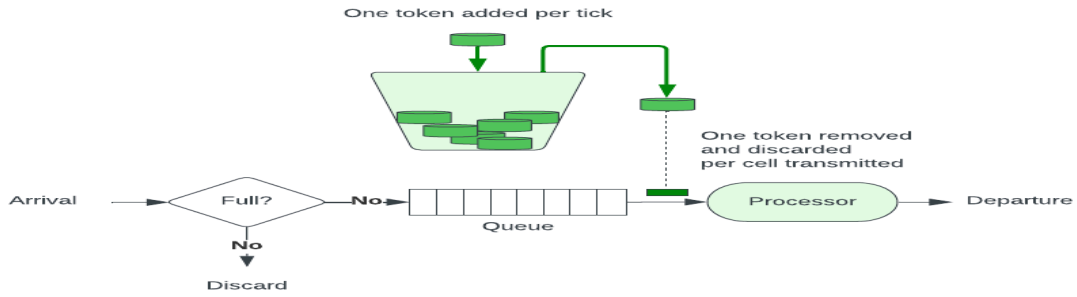
- **Overflow**:

If packets arrive faster than the bucket can leak them, the bucket fills up. Any packets that would cause the bucket to overflow are discarded.



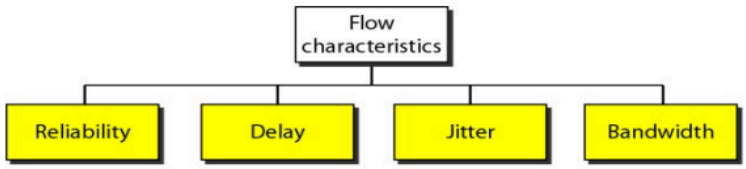

**The token bucket algorithm** is a method used in networking to control network traffic by allowing for bursts of data while ensuring an average transmission rate is not exceeded. It works by filling a "bucket" with "tokens" at a fixed rate; each token represents permission to send a certain amount of data. To send data, tokens are removed from the bucket. If the bucket is full, it stops accepting tokens; if the bucket is empty, data transmission must wait until more tokens are added




## QUALITY OF SERVICE

**Quality of service** (**QoS**) is the overall performance of a telephony or computer network, particularly the performance seen by the users of the network.

### Flow Characteristics




### Reliability

Reliability is a characteristic that a flow needs. Lack of reliability means losing a packet or acknowledgment, which entails retransmission. However, the sensitivity of application programs to reliability is not the same. For example, it is more important that electronic mail, file transfer, and Internet access have reliable transmissions than telephony or audio conferencing.

### Delay

Source-to-destination delay is another flow characteristic. Again applications can tolerate delay in different degrees. In this case, telephony, audio conferencing, video conferencing, and remote log-in need minimum delay, while delay in file transfer or e-mail is less important.

### Jitter

Jitter is the variation in delay for packets belonging to the same flow. For example, if four packets depart at times 0, 1, 2, 3 and arrive at 20, 21, 22, 23, all have the same delay, 20 units of time. On the other hand, if the above four packets arrive at 21, 23, 21, and 28, they will have different delays: 21,22, 19, and 24. For applications such as audio and video, the first case is completely acceptable; the second case is not. For these applications, it does not matter if the packets arrive with a short or long delay as long as the delay is the same for all packets. For this application, the second case is not acceptable. Jitter is defined as the variation in the packet delay.

### Bandwidth

Different applications need different bandwidths. In video conferencing we need to send millions of bits per second to refresh a color screen while the total number of bits in an e-mail may not reach even a million.