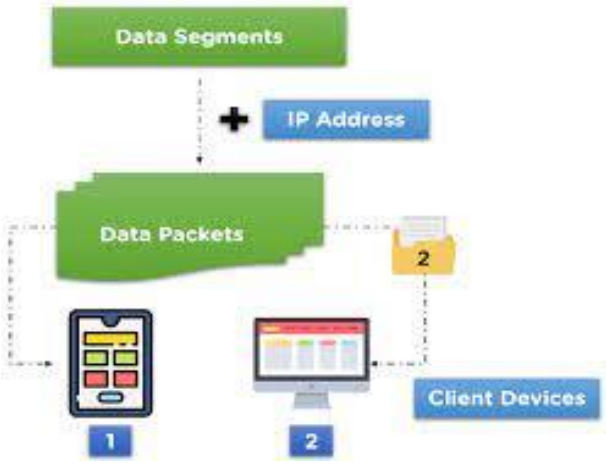
UNIT– III: Network Layer

The network layer is an essential component of the OSI (Open Systems Interconnection) model and the TCP/IP (Transmission Control Protocol/Internet Protocol) protocol suite. It is the third layer in the OSI model and the Internet layer in the TCP/IP model. The primary function of the network layer is to enable communication and data transfer between different networks by providing routing and addressing capabilities.



Key features and responsibilities of the network layer include:

1. Logical Addressing: The network layer uses logical addresses (such as IP addresses in the case of TCP/IP) to uniquely identify devices on a network. These addresses are used to route data packets from the source to the destination across multiple networks.
2. Routing: Routing is one of the major functions of the network layer. It involves determining the best path for data packets to travel from the source to the destination across a series of interconnected networks. Routers, which operate at the network layer, use routing protocols and routing tables to make these decisions.
3. Packet Forwarding: The network layer is responsible for forwarding data packets between different networks. As a packet arrives at a router, the network layer examines the destination address and determines the next hop for the packet to continue its journey toward the final destination.
4. Fragmentation and Reassembly: The network layer can fragment large data packets into smaller segments to accommodate the maximum transmission unit (MTU) size of the underlying network technologies. At the destination, the network layer reassembles the fragments into the original packet.
5. Quality of Service (QoS): Some network layer protocols support QoS features, allowing the network to prioritize certain types of traffic over others. This ensures that critical data, such as video or voice packets, receive preferential treatment and lower latency during transmission.
6. Tunneling: The network layer can encapsulate data from higher-layer protocols within its own packets, a process known as tunneling. This is commonly used in virtual private networks (VPNs) to securely transmit data over public networks.

In the TCP/IP model, the Internet Protocol (IP) is the primary network layer protocol, responsible for logical addressing, routing, and packet forwarding. Other network layer protocols in the TCP/IP suite include ICMP (Internet Control Message Protocol), IGMP (Internet Group Management Protocol), and IPsec (Internet Protocol Security).

Network Layer Design Issues

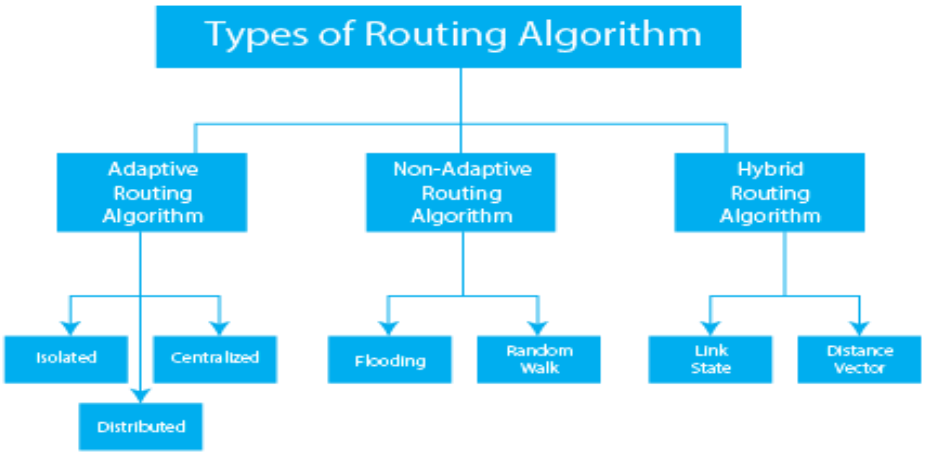
Designing the network layer involves addressing various issues and considerations to ensure efficient and reliable communication between devices and networks. Here are some key design issues that network layer designers need to address:

1) **Addressing Scheme:** Choosing an appropriate addressing scheme is crucial for the network layer. It involves defining how devices on the network are uniquely identified. For example, in the case of the Internet, the IP (Internet Protocol) addressing scheme is used to assign unique IP addresses to devices.
2) **Routing Algorithms**: Selecting the right routing algorithms is essential for efficient data packet forwarding. Different algorithms, such as distance-vector, link-state, and path-vector, have their advantages and trade-offs in terms of convergence speed, scalability, and overhead.

3) **Routing Protocol Selection**: Determining the appropriate routing protocols to be used within the network is vital. Different network environments may require different routing protocols. For instance, OSPF (Open Shortest Path First) is often used within enterprise networks, while BGP (Border Gateway Protocol) is used to manage routes between different autonomous systems on the Internet.
4) **Scalability:** Network designs must be scalable to accommodate growth in the number of devices and network traffic. Scalability involves ensuring that the network can handle increased demands without compromising performance or stability.
5) **Interoperability**: The network layer should be designed to ensure interoperability with different devices and networking technologies. This includes supporting various link-layer technologies and being able to communicate with networks based on different protocols.
6) **Error Handling and Recovery:** The network layer must handle errors that occur during packet transmission. This may involve implementing error detection and correction mechanisms or relying on higher layers to handle error recovery.
7) **Fragmentation and Reassembly:** The network layer should address issues related to packet fragmentation and reassembly. It may need to split large packets into smaller fragments to fit the Maximum Transmission Unit (MTU) of the underlying link-layer technology and reassemble them at the destination.
8) **Quality of Service (QoS)**: Designing the network layer to support Quality of Service is crucial for providing different levels of priority and service guarantees for specific types of traffic. This ensures that critical applications, such as real-time voice and video, receive the necessary bandwidth and lower latency.
9) **Security:** Network layer design should incorporate security features to protect against various threats, such as IP spoofing, denial-of-service (DoS) attacks, and unauthorized access. IPsec and other security protocols can be used to provide data confidentiality, integrity, and authentication.
10) **Mobility Support**: In mobile networks, the network layer needs to address issues related to mobile device movement and seamless handover between different access points or networks.
11) **Multicasting and Broadcasting:** Designing the network layer to support multicasting and broadcasting is important for efficiently sending data to multiple recipients or all devices within a network segment.

Routing Algorithms

Routing algorithms are crucial components of the network layer responsible for determining the best paths for data packets to travel from the source to the destination across a network. Different routing algorithms have been developed to address various network topologies, scalability requirements, and optimization goals. Here are some common types of routing algorithms:



1. **Static Routing:** In static routing, network administrators manually configure the routes in the routing table. The routes do not change unless modified explicitly by an administrator. It is suitable for small, stable networks where the topology rarely changes.
2. **Dynamic Routing**: Dynamic routing algorithms automatically calculate and update routing information based on network changes. They use metrics, such as distance, bandwidth, delay, or load, to determine the best path. Dynamic routing protocols are more adaptive and scalable for larger networks. Examples include RIP (Routing Information Protocol), OSPF (Open Shortest Path First), and EIGRP (Enhanced Interior Gateway Routing Protocol).
3. **Distance-Vector Routing:** Distance-vector routing algorithms, like RIP, use simple metrics (hop count) to measure the distance to a destination. Routers exchange routing tables with their neighbors and

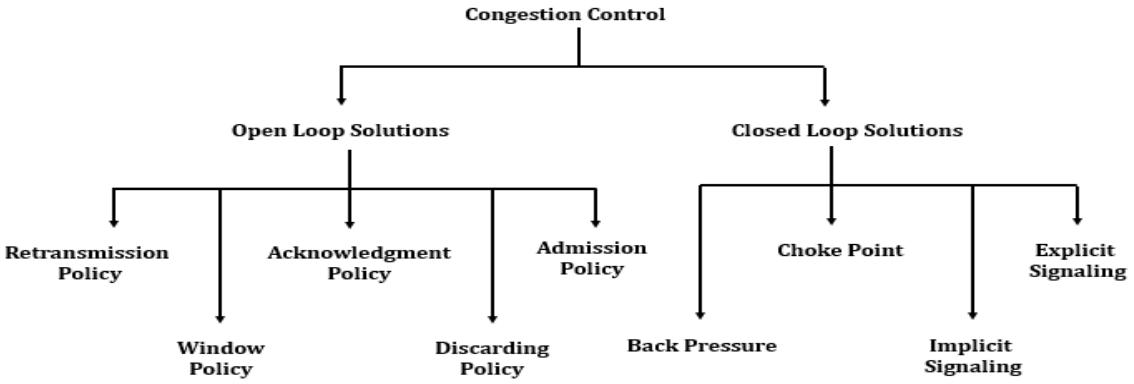
update them based on received information. The algorithm converges slowly and is prone to routing loops.

4. **Link-State Routing**: Link-state routing algorithms, such as OSPF and IS-IS (Intermediate System to Intermediate System), build a complete view of the network by exchanging link-state information with neighboring routers. They calculate the shortest path to each destination using Dijkstra's algorithm. Link-state routing converges faster and offers better scalability than distance-vector routing.
5. **Path-Vector Routing:** Path-vector routing algorithms, like BGP (Border Gateway Protocol), are used for interdomain routing on the Internet. They advertise paths rather than simple metrics, allowing more control over the selection of routes between autonomous systems.
6. **Flooding**: Flooding is a simple but inefficient routing technique where a packet is forwarded to all neighboring nodes except the one it was received from. While this ensures that the packet reaches its destination eventually, it can lead to excessive network traffic and redundancy.
7. **Multicast Routing:** Multicast routing algorithms handle the efficient delivery of multicast traffic to multiple recipients. They create distribution trees to minimize replication and ensure that packets are forwarded only to interested receivers.
8. **Anycast Routing**: Anycast routing allows multiple nodes to share the same IP address, and the routing algorithm directs packets to the nearest or most optimal node offering the service.
9. **Source Routing:** In source routing, the entire path that a packet should follow from the source to the destination is determined by the sender and explicitly included in the packet's header.

Congestion Control Algorithms

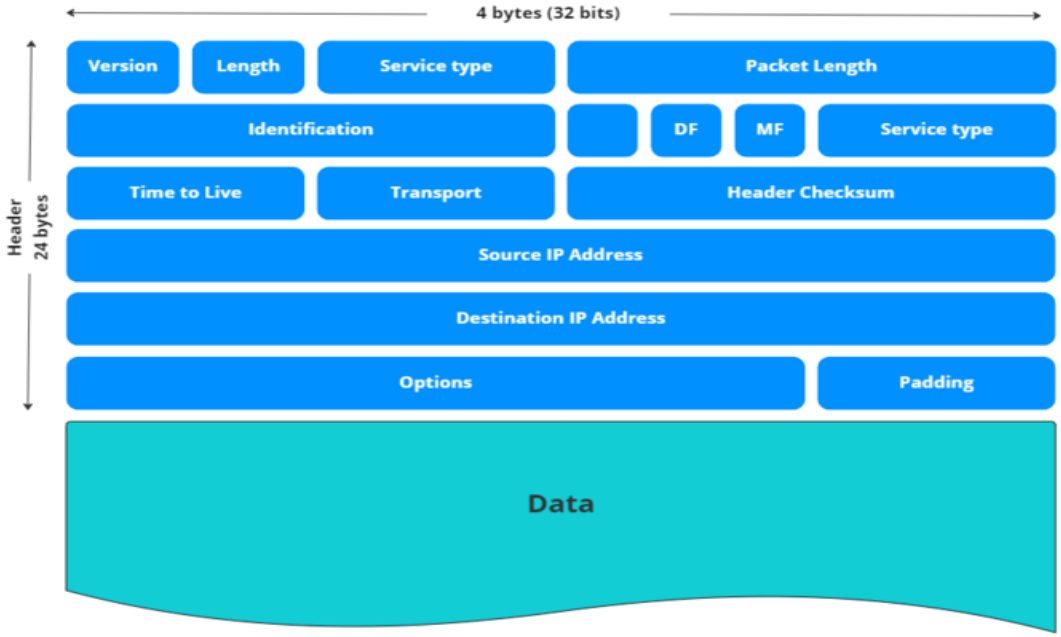
Congestion control algorithms are techniques employed to manage and prevent network congestion, which occurs when the demand for network resources exceeds the available capacity. Congestion can lead to increased packet loss, delays, and reduced overall network performance. To mitigate congestion and maintain network stability, various congestion control algorithms have been developed. Here are some commonly used congestion control algorithms:

1) **Random Early Detection (RED)**: RED is a router-based congestion control algorithm that aims to prevent global congestion collapse. It monitors the average queue length and randomly drops packets before the queue becomes excessively full. By dropping packets early, RED encourages TCP/IP hosts to reduce their transmission rates, thus alleviating congestion.
2) **Explicit Congestion Notification (ECN):** ECN is a mechanism that allows routers to signal congestion to end systems without dropping packets. When a router's queue approaches congestion, it marks the packets instead of dropping them. The end systems read the ECN marks and respond by reducing their transmission rates.
3) **TCP Congestion Control:** TCP (Transmission Control Protocol) itself incorporates several congestion control mechanisms to prevent network congestion. TCP uses algorithms such as Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery to dynamically adjust the sending rate based on network conditions.
4) **Quality of Service (QoS):** QoS mechanisms prioritize certain types of traffic over others, ensuring that critical data (e.g., real-time voice or video) receives preferential treatment. By allocating appropriate bandwidth to high-priority traffic, congestion can be controlled more effectively.
5) **Traffic Shaping and Policing:** Traffic shaping and policing mechanisms control the rate at which data is transmitted into the network. These techniques help smooth out traffic bursts, preventing congestion caused by sudden spikes in data transmission rates.
6) **Active Queue Management (AQM)**: AQM algorithms, like RED mentioned earlier, proactively manage the queue lengths at routers. By dropping or marking packets when queues approach congestion, AQM helps maintain a balanced traffic flow.
7) **Forward Explicit Congestion Notification (FECN) and Backward Explicit Congestion Notification (BECN):** FECN and BECN are congestion notification mechanisms used in Frame Relay networks. FECN is set by a sender to indicate congestion to the receiver, while BECN is set by a switch to indicate congestion back to the sender.
8) **Resource Reservation Protocols:** Resource reservation protocols, like RSVP (Resource Reservation Protocol), allow applications to request and reserve network resources in advance. This ensures that sufficient resources are available to meet the needs of specific flows, reducing the chances of congestion.
9) **Load Balancing:** Load balancing techniques distribute network traffic across multiple paths or links, preventing any single path from becoming overwhelmed and congested.
10) **Traffic Engineering:** Traffic engineering involves the deliberate manipulation of traffic paths to optimize network performance and reduce congestion. This may include route selection based on network conditions or dynamically adjusting path preferences.



Internet Protocol Header

The Internet Protocol (IP) header is a fundamental part of IP packets used for communication over the Internet and other IP-based networks. It is located at the network layer (Layer 3) of the OSI model and contains essential information required for the proper routing and delivery of data packets from the source to the destination. Below is a breakdown of the fields commonly found in an IPv4 header:



1. **Version (4 bits):** Indicates the IP version being used. For IPv4, this field is set to '0100', indicating a 4-bit value of 4.
2. **Header Length (IHL) (4 bits):** Specifies the total length of the IP header in 32-bit words. This value is needed because the IP header may have optional fields, and the header length varies accordingly.
3. **Type of Service (TOS) or Differentiated Services Code Point (DSCP) (8 bits):** Originally used for Quality of Service (QoS) purposes, this field was later redefined as DSCP, which allows for differentiated services in modern networks.
4. **Total Length (16 bits):** Indicates the total length of the IP packet, including both the header and data (payload). The maximum value for this field is 65,535 bytes.
5. **Identification (16 bits):** Used to uniquely identify each IP packet from the sender. This field is essential for reassembling fragmented packets at the destination.
6. **Flags (3 bits):** Used in conjunction with the Fragment Offset field (see below) to handle packet fragmentation. The three flags are: 'Reserved', 'Don't Fragment' (DF), and 'More Fragments' (MF).
7. **Fragment Offset (13 bits):** When an IP packet is fragmented, this field indicates the position of the current fragment relative to the original unfragmented packet.
8. **Time to Live (TTL) (8 bits)**: Represents the maximum number of hops (routers) the packet can traverse before being discarded. It helps prevent packets from circulating indefinitely in the network.
9. **Protocol (8 bits):** Identifies the transport layer protocol (e.g., TCP, UDP, ICMP) that the data payload of the IP packet is associated with.

10. **Header Checksum (16 bits)**: Used to check the integrity of the IP header during transmission. It is recalculated at each hop along the route.
11. **Source IP Address (32 bits)**: Specifies the IP address of the sender (source) of the packet.
12. **Destination IP Address (32 bits):** Specifies the IP address of the intended recipient (destination) of the packet.
13. **Options (variable length):** This field is optional and is used to accommodate additional functionalities or experimentations. It is rarely used in typical IP packets.

The IPv6 header has a different structure, with some of the fields being modified or eliminated to improve efficiency and security. However, the core functionality of addressing and routing remains similar to IPv4. IPv6 uses 128-bit addresses instead of 32-bit addresses in IPv4.

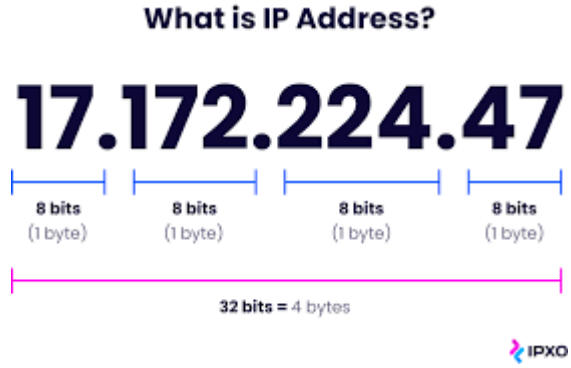
IP Addresses

IP addresses (Internet Protocol addresses) are numerical labels assigned to devices connected to a network that use the Internet Protocol (IP) for communication. They serve as unique identifiers for each device, allowing data packets to be properly addressed and routed across networks. IP addresses play a fundamental role in the functioning of the Internet and other IP-based networks.

There are two versions of IP addresses in use:

IPv4 (Internet Protocol version 4): IPv4 addresses consist of 32 bits and are typically represented in dotted-decimal notation, where each 8-bit segment is expressed as a decimal number (e.g., 192.168.0.1). IPv4 provides around 4.3 billion unique addresses, which has become insufficient with the rapid growth of the Internet and the increasing number of connected devices.

IPv6 (Internet Protocol version 6): IPv6 addresses, introduced to overcome the limitations of IPv4, are 128 bits long. They are represented in hexadecimal format and separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334). IPv6 offers an enormous address space, providing approximately 340 undecillion unique addresses (3.4 x 10^38), ensuring that the world will have an ample supply of IP addresses for the foreseeable future.



IP addresses can be further divided into two parts: the network portion and the host portion. The network portion identifies the specific network to which the device is connected, while the host portion identifies the individual device within that network.

Types of IP addresses:

Public IP Addresses: Public IP addresses are assigned to devices connected directly to the Internet, allowing them to be reachable from anywhere on the global network. Internet Service Providers (ISPs) assign public IP addresses to their customers.

Private IP Addresses: Private IP addresses are used within private networks, such as local area networks (LANs), and are not routable over the Internet. They are used for internal communication among devices within the network. Some common private IP address ranges include:

- 10.0.0.0 to 10.255.255.255
- 172.16.0.0 to 172.31.255.255
- 192.168.0.0 to 192.168.255.255

Dynamic IP Addresses: Dynamic IP addresses are assigned to devices by the network's DHCP (Dynamic Host Configuration Protocol) server. These addresses may change over time, as devices connect and disconnect from the network.

Static IP Addresses: Static IP addresses are manually configured for a device and do not change. They are typically used for devices that need to maintain a consistent, fixed address, such as servers or network equipment.

IP addresses are essential for identifying and locating devices on the Internet and enabling the routing of data packets from the source to the destination

subnetting and super netting

Subnetting and supernetting are techniques used in IP addressing to efficiently allocate and manage IP address space, especially in scenarios where a network needs to be divided into smaller subnetworks or when multiple smaller networks are combined into a larger one.

Subnetting:

Subnetting is the process of dividing a larger network into smaller subnetworks, or subnets. This is typically done to improve network efficiency, reduce broadcast domains, and manage IP address allocation more effectively. Subnetting involves borrowing bits from the host portion of the IP address to create subnetworks.

In IPv4, each IP address is divided into two parts: the network portion and the host portion. The subnet mask determines the boundary between these two portions. By changing the subnet mask, you can create different-sized subnets from a given IP address range. Subnetting allows organizations to make more efficient use of available IP addresses.

For example, consider the IP address 192.168.1.0 with the default subnet mask 255.255.255.0 (or /24 in CIDR notation). This allows for 256 possible host addresses (from 192.168.1.1 to 192.168.1.254). However, by subnetting this network, you can create multiple subnets, each with a smaller range of IP addresses. For instance, using a subnet mask of 255.255.255.224 (or /27), you create eight subnets, each with 30 usable host addresses.

Supernetting (CIDR - Classless Inter-Domain Routing):

Supernetting, or CIDR (Classless Inter-Domain Routing), is the opposite of subnetting. It is the process of combining multiple smaller contiguous subnets into a larger network. Supernetting is used to summarize multiple IP address ranges and represent them as a single, larger address block. It is commonly used in routing to reduce the size of routing tables and make routing more efficient.

CIDR notation expresses the size of the network prefix using a slash followed by the number of bits in the network portion of the IP address. For example, an IP address represented as 192.168.1.0/24 indicates that the first 24 bits are the network portion, and the remaining 8 bits are the host portion.

Supernetting is often used by Internet Service Providers (ISPs) and large organizations to aggregate multiple smaller networks into larger blocks, making it easier to advertise routing information and reducing the size of routing tables in the global Internet routing infrastructure.

Both subnetting and supernetting are essential techniques for effectively managing IP address space and optimizing network operations, particularly in the context of IPv4 address exhaustion and the migration to IPv6, which provides a vastly larger address space.