UNIT 1: -

Internetworking:-

Internetworking Basics:-Internetworking is the practice of interconnecting multiple computer networks, such that any pair of hosts in the connected networks can exchange messages irrespective of their hardware-level networking technology.

Introduction of Internetworking

Internetworking is combined of 2 words, inter and networking which implies an association between totally different nodes or segments. This connection area unit is established through intercessor devices akin to routers or gateway. The first term for associate degree internetwork was catenet. This interconnection is often among or between public, private, commercial, industrial, or governmental networks. Thus, associate degree internetwork could be an assortment of individual networks, connected by intermediate networking devices, that function as one giant network. Internetworking refers to the trade, products, and procedures that meet the challenge of making and administering internet works.

To enable communication, every individual network node or phase is designed with a similar protocol or communication logic, that is Transfer Control Protocol (TCP) or Internet Protocol (IP). Once a network communicates with another network having constant communication procedures, it's called Internetworking. Internetworking was designed to resolve the matter of delivering a packet of information through many links.

There is a minute difference between extending the network and Internetworking. Merely exploitation of either a switch or a hub to attach 2 local area networks is an extension of LAN whereas connecting them via the router is an associate degree example of Internetworking. Internetworking is enforced in Layer three (Network Layer) of the OSI-ISO model. The foremost notable example of internetworking is the Internet.

There is chiefly 3 units of Internetworking:

1. Extranet

2. Intranet
3. Internet

Intranets and extranets might or might not have connections to the net. If there is a connection to the net, the computer network or extranet area unit is usually shielded from being accessed from the net if it is not authorized. The net isn't thought-about to be a section of the computer network or extranet, though it should function as a portal for access to parts of the associate degree extranet.

1. **Extranet –** It's a network of the internetwork that's restricted in scope to one organization or entity however that additionally has restricted connections to the networks of one or a lot of different sometimes, however not essential. It's the very lowest level of Internetworking, usually enforced in an exceedingly personal area. Associate degree extranet may additionally be classified as a Man, WAN, or different form of network however it cannot encompass one local area network i.e. it should have a minimum of one reference to associate degree external network.
2. **Intranet –** This associate degree computer network could be a set of interconnected networks, which exploits the Internet Protocol and uses IP-based tools akin to web browsers and FTP tools, that are underneath the management of one body entity. That body entity closes the computer network to the remainder of the planet and permits solely specific users. Most typically, this network is the internal network of a corporation or different enterprise. An outsized computer network can usually have its own internet server to supply users with browsable data.
3. **Internet –** A selected Internetworking, consisting of a worldwide interconnection of governmental, academic, public, and personal networks based mostly upon the Advanced analysis comes Agency Network (ARPANET) developed by ARPA of the U.S. Department of Defense additionally home to the World Wide Web (WWW) and cited as the 'Internet' to differentiate from all different generic Internetworks. Participants within the web, or their service suppliers, use IP Addresses obtained from address registries that manage assignments.

Internetworking Models:-

In this article I describe about internetworking model in computer network for CCNA exam. Internetworking model in computer network followed by all the manufacturer to provide the compatibility among the various hardware. There are many manufacturer of computer machine in the market. Initially When computers became single user public computer. The computers communicate with only same brand machines. It happens because there was no any fix standard for data transfer between different devices. It is very difficult to make communication with each other when the hardware are of different brands or company.

Internetworking model in computer network followed by various vendors to overcome this problem. After implementation of internetworking model in computer network, equality maintains by all manufacturer. In 1970 the Open Systems Interconnection (OSI) reference model was created by the

International Organization for Standardization (ISO). The OSI model was meant to create inter-operable network with different manufactured devices. In this article I describe all layered approach of internetworking model in computer network.

Importance of internetworking model in computer network

Not only hardware, software also not supported for work the different computer brand. It became very difficult for all computer users to working without implementation of internetworking model in computer network. It is necessary then to make some common protocols for all vendors of computer. Before implementation of internetworking model in computer network, all vendors implements their own protocols on computer hardware and software.

In networking OSI reference model became helpful. OSI reference model describes the flow of data between nodes in any network. Data from one computer application to another computer application transfer by following some common protocols. The OSI reference layer also become beneficial for troubleshooting the network problems. TCP/IP and Three layered hierarchic model of Cisco became more helpful alongside the OSI reference model.

The Layered Approach for internetworking model in computer network.

The Layered approach was the best way to make equality for all computer devices. Layers are not physical but following some protocols. Protocols are for connectivity, connections, data transfer and more. All manufacturer begin to follow the layered approach for internetworking model in computer network. Later in the briefing of OSI layer architecture article we approach layers briefly. Later the OSI reference model change in TCP/IP reference model. OSI layer architecture have 7 layers. TCP/IP reference model convert these 7 layers into only four layers.

OSI Reference internetworking model in computer network

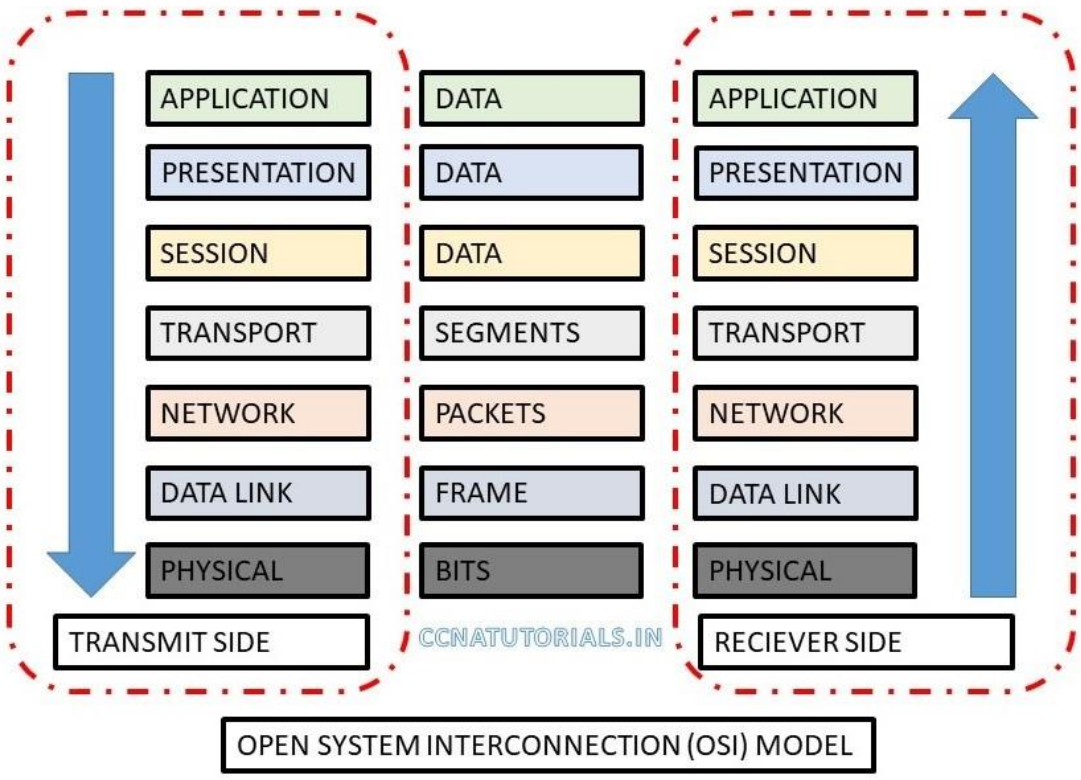
OSI is acronym for open system interconnection. The OSI is a logical reference internetworking model in computer network. OSI model helps for data flow between different devices and operating systems. All manufacturer used their own architecture before invention of OSI reference model. It was very difficult to establish data communication between different devices. To overcome this problem international organization for standardization (ISO) created the open systems interconnection (OSI) reference model. OSI reference model make data flow possible between different operating system, devices and hardware.

Structure of OSI reference internetworking model in computer network

OSI reference internetworking model in computer network consist of 7 layers. These 7 layers further divided into two groups. First 3 layers works for application communication and remaining 4 layers works for data flow. Application, presentation and session layers define the application communication. Transport, network, data link and physical layers define the data flow. Networking protocols works only on last four layers.

Layers of OSI reference internetworking model in computer network

Below image shows the seven layers of OSI reference model. Transmit and receive side shown in diagram.



Let's take an example of email sending and receiving. There are two users for email flow on internet. A sender composes and send mail at application layer. At transmit side data flow from application to physical layer. Physical media carry the data for receiver. At receiver end data flow from physical to application layer. At application layer receiver got the email. During this process a lots of protocol functions. In below diagram some protocols example shown at various layers.

The application layer in OSI reference internetworking model in computer network

Application layer provides interface to user for working on a computer. Take an example of outlook emailing application. What a user can see on the computer screen is the application layer functionality. Application layer also interact with the presentation layer. Some examples of application layer are internet browsing, email, file transfers, work on remote computer, printing etc. Standalone software or application like notepad doesn't lie in OSI reference model. The main function of application layer is to provide an interface to user for Working on computer. Some common protocols work on application layer are DNS, HTTP, HTTPS, FTP, POP3 etc.

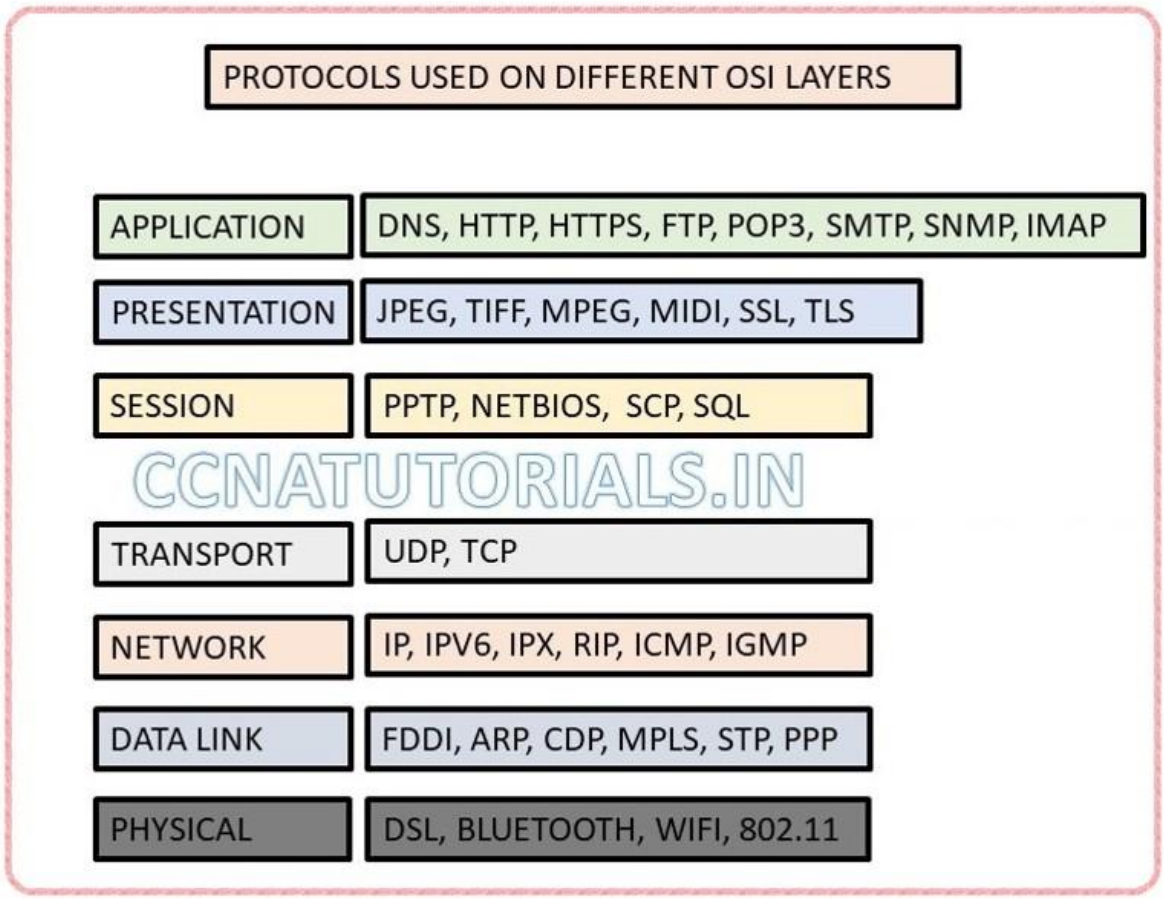
The presentation layer in OSI reference internetworking model in computer network

Presentation layer works as a translator for data. Presentation layer translate the data in prescribed format for user. At sender computer presentation layer translate the data received from application layer. This translated data forwarded to session layer. At receiver computer reverse process done at presentation layer. The best example of translation in converting the ASCII code into EBCDIC code.

You may also like to read -- Access point in networking

Presentation layer also responsible for data compression in internetworking model in computer network, encryption and decryption etc. suppose a user is watching online video. The data translated to required video format. Similarly, for image data converted to required format like jpeg, tiff etc. Presentation layer integrate all formats of data into a standard format. The presentation layer translates data of application layer to network format The presentation layer is responsible for the following:

- Data encryption/decryption
- Character/string conversion
- Data compression
- Graphic handling



The session layer in OSI reference internetworking model in computer network

Function of session layer is to manage and maintain session between client to client. Session layer make and close the session between two end devices. PPTP, SQL are protocols used by session layer. Some common functions for session layer are dialog control, Token management, synchronization. This layer maintain session in the form or simplex, half-duplex and full duplex. Simplex transmission is one side transmission. Half-duplex is both side transmission but one side at a time. Full duplex is both side transmission in real time like telephone.

The transport layer in OSI reference internetworking model in computer network

TCP and UDP protocols are integral to transport layer in internetworking model in computer network. Transport layer create segments of the data stream received from presentation later at transmission stage. At receiver stage transport layer assemble the segments and send the data stream to presentation layer. Transport layer establish a logical connection between the end equipment on an

internetwork. The Transport layer is responsible for multiplexing upper layer applications, create sessions, sequencing data, acknowledgments etc.

Transport layer works on connection oriented communication. It is reliable for data transmission. In connection oriented communication first step is handshake. After handshaking data transfer starts until the complete data transferred. If data transferring using TCP protocol, then acknowledgment of complete data transfer is must. In case complete data not transferred same data re-transmit again. Data flow depends on speed of the network. During transfer the data congestion can occur. To overcome data congestion flow control term used. Flow control ensure data integrity at transport layer. Data overflowing and buffers prevented by flow control.

Connection oriented communication uses sequencing, acknowledgements, flow control, congestion control and windowing. When a device send data and doesn't need acknowledgement, it is called windowing. Online video uses windowing. Window size defined when the acknowledgement required. Suppose you set the window size 2. For window size 2 transmitting computer will check acknowledgement after every 2 segments transmitted. Minimum the windowing size increase the congestion because acknowledgment required more.

The network layer in OSI reference internetworking model in computer network

Network layer commonly known as layer 3. Network layer works on IP address system. Internetworking in fully depends on network layer. Segments received from transport layer breaks in packets and forwarded to data link layer. Similarly, packets reassembled received from data link layer and forwarded to transport layer. Routers are layer 3 devices. Router configured at layer 3 for internetworking between different LANs.

IP, IPv6, IPX, RIP protocols functions on layer 3 or network layer. Protocols used for data traffic at layer 3 are called routed protocol. Whereas protocols used to keep update the routing table of neighboring router are called routing protocol.

Suppose a packet received on a router interface. Router will check the destination address on this packet. Router check it's ip address in routing table and forward it to related exit interface. If router doesn't have the destination address in its routing table, then router will drop the packet. There are Route update packets used to keep update the neighboring router. Router maintain individual routing protocol for each protocol.

Router doesn't send any broadcast or multicast packet itself. The Router use a logical address in a network layer header to forward a packet. Access list used by router to keep control security. Routers provide connections between virtual LAN.

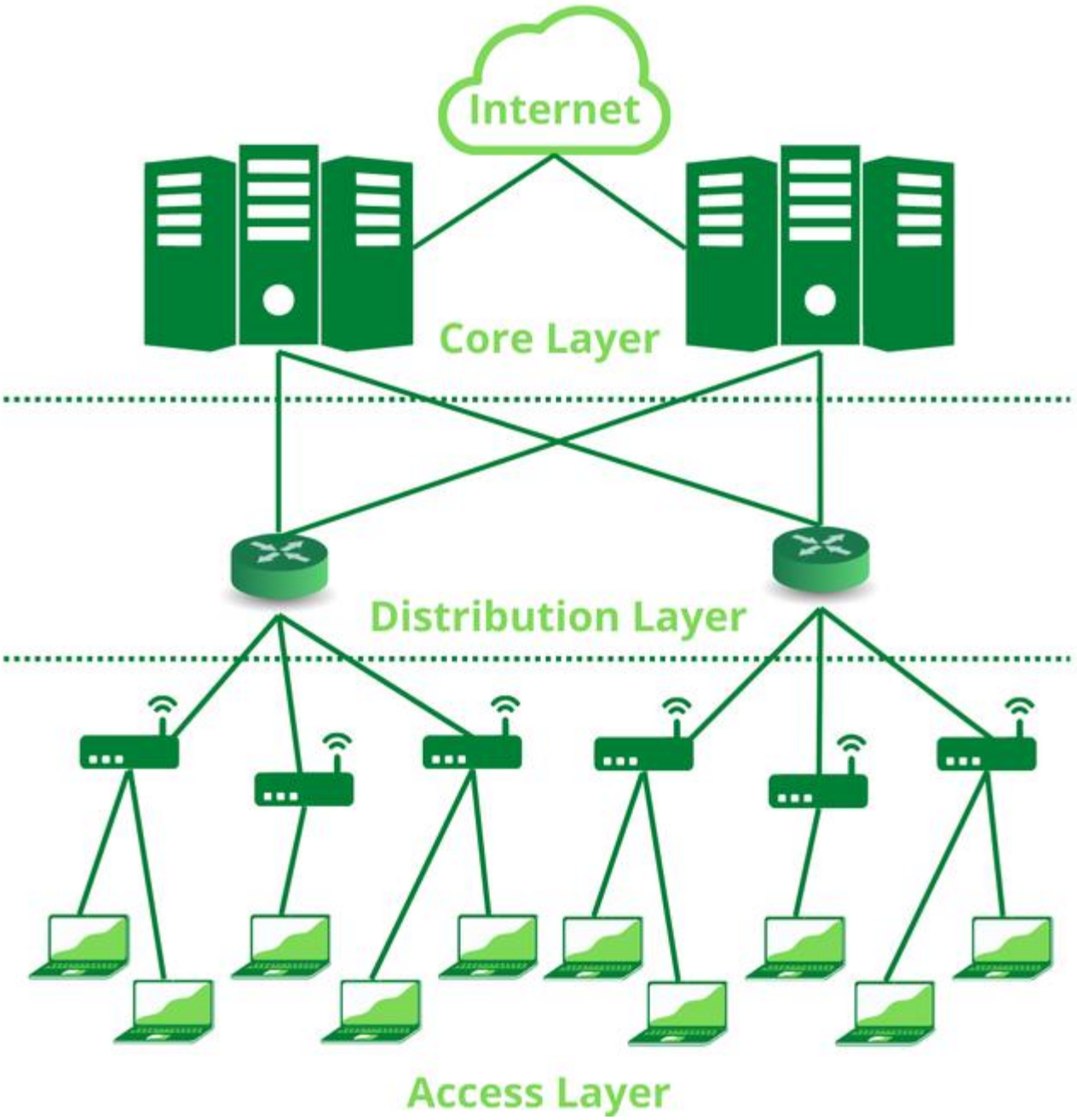
The data link layer in OSI reference internetworking model in computer network

Data link layer works on frame in a OSI reference internetworking model in computer network. Packets received from network layer breaks into frame at data link layer during transmission of data. Frame constructed the packets and forwarded to network layer during receive data. some protocols used at data link layer are FDDI, PPP etc. data link layer also convert the frame into bits and send to physical layer and vice versa. Every frame adds a header which contains the destination and source address. Data link layer use the MAC address or source and destination. It never uses the IP addressing system.

Cisco Three Hierarchical Model: Core Layer, Distribution Layer & Access Layer.:-

In a three-layer hierarchical model for Cisco routers, The first layer is the local area network (LAN) that uses IEEE 802.3 Ethernet technology to connect devices on the same physical segment (or subnet). This low level of networking provides easy sharing of media and files between individual workstations and printers connected to the LAN, as well as providing security against unauthorized access by outsiders. The next layer is the wide area network (WAN), which offers faster data transfer rates than LANs but can be more expensive due to its reliance on leased lines or satellite links. WANs typically use TCP/IP protocols at this higher level, allowing them to communicate with other networks across corporate boundaries or over long distances.
This model consists of three layers:

1. The Access Layer
2. The Distribution Layer
3. The Core Layer



For more details, you can also refer to the article TCP/IP Model.

Access Layer:

The Access Layer is the part of the network which enables the users to connect to the wired Ethernet Network. It enables the users to share data and resources on the local network. The devices used in this layer include Ethernet Switches and Hubs.

Hubs are basically multiport repeaters. They are devices that cannot decode the data packets received by them because they lack circuitry and logic to decode the data packets. Hubs cannot

determine which host must receive the data packet. They simply repeat the electronic signals received on one interface to all other interfaces on the hub, thus all the hosts connected to the hub receive the data packet. Hubs have a fatal issue of collision. if two hosts transmit data packets at the same time, they would “collide” and be rendered useless. The hosts must retransmit the packets again.

Another device used in the Access Layer is the Ethernet Switch. An Ethernet Switch is far more capable than hubs. They can decode the data packets and determine the interface to which the data packet must be forwarded. they use the MAC address, also known as the Physical Address, assigned to the host to forward the data packets. This reduces the issue of collision faced while using hubs. The development of Switches has rendered Hubs obsolete. Devices like *Cisco 2390XR* are used at this layer.

Distribution Layer:

When a network grows beyond a certain size, it must be divided into multiple local (Access Layer) networks. the distribution layer connects these networks together. It ensures that local traffic remains confined to local networks and governs traffic control between these networks.

This layer uses Routers to connect multiple networks together. Routers and other devices on this layer are meant to connect multiple networks together, and not individual hosts. In order to navigate traffic between hosts on different networks, IP Address, also known as Logical Address, is used. The Router maintains a Routing Table to determine the interface on which to forward the received data packet.

This layer also acts as an intermediary between the Access Layer and the Core Layer. Devices like the *Cisco C9300* are used at this layer.

Core Layer:

This layer is considered the backbone of a network, as it is used to connect multiple Distribution Layer devices together. This layer uses the most powerful devices to manage the traffic between the networks. The speed at which data flows in this layer is upwards of 10 Gigabit Ethernet. This layer has the maximum number of redundant connections (*Redundancy is the process of introducing extra connections between the same network points to ensure reliable data transfer even if one of the connections is down*) in order to ensure reliable connectivity.

Devices like *Cisco Catalyst 9600* are used at this layer with high-speed and high-bandwidth transmission media like optical fiber cable.

Advantages:

- Larger, more complex networks are divided into smaller, manageable subnetworks.
- Local traffic remains local, which increases network efficiency.
- Makes the network scalable. The addition of new networks does not affect the performance of existing ones.

Conclusion:

The Three-Layered Hierarchical Model in Cisco divides a network into the following three layers:

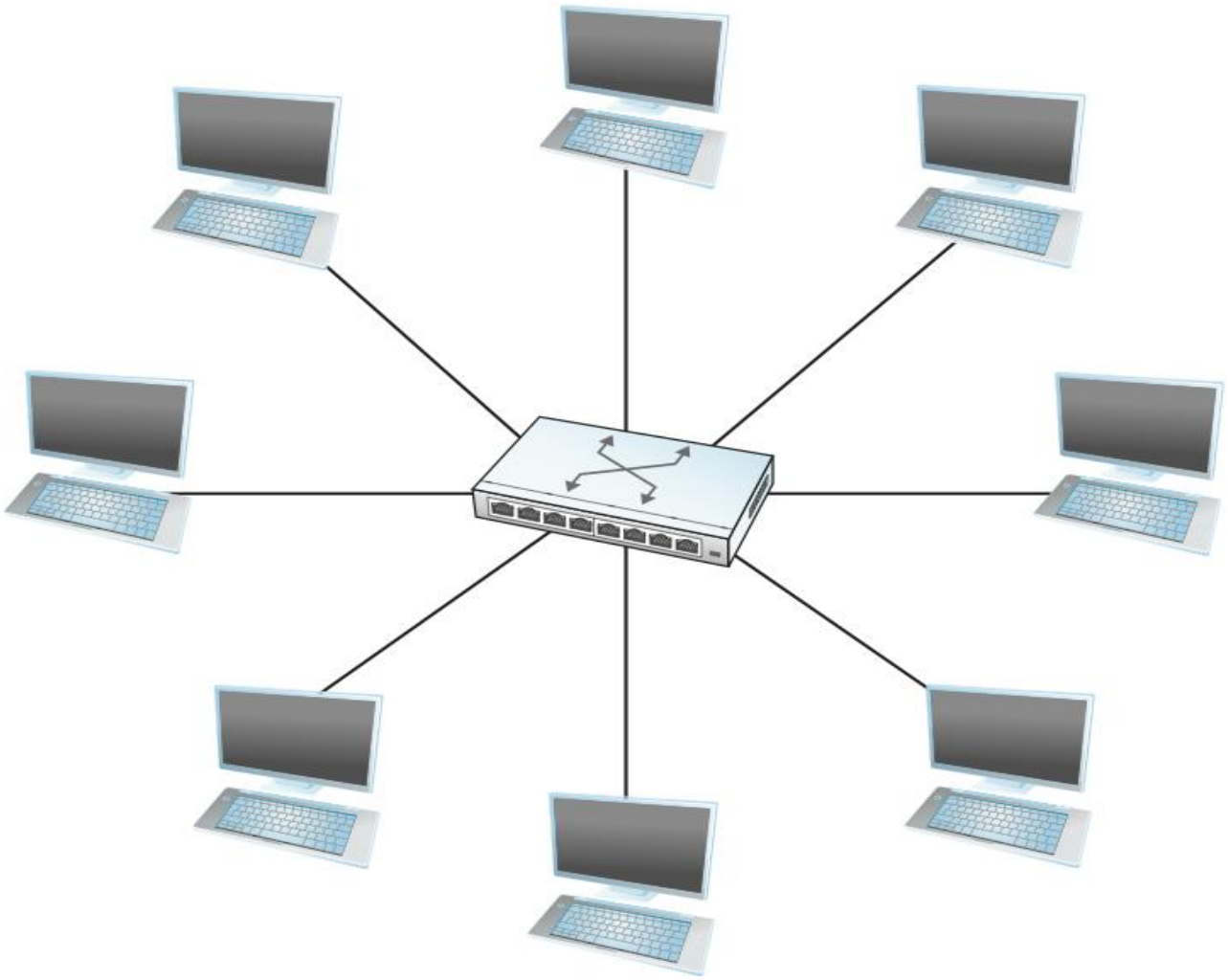
1. The Access Layer: Provides access points for hosts to connect to the network.
2. The Distribution Layer: Acts as an intermediary between the Core Layer and the Access Layer, and keeps local traffic confined to local networks.

3. The Core Layer: Handles and transports huge amounts of data quickly and reliably, and connects multiple end networks together.

Bridging / Switching: Switching Services:-

In the simplest terms, a switch is a mechanism that allows us to interconnect links to form a larger network. A switch is a multi-input, multi-output device that transfers packets from an input to one or more outputs. Thus, a switch adds the star topology (see Figure 1) to the set of possible network structures. A star topology has several attractive properties:

- Even though a switch has a fixed number of inputs and outputs, which limits the number of hosts that can be connected to a single switch, large networks can be built by interconnecting a number of switches.
- We can connect switches to each other and to hosts using point-to-point links, which typically means that we can build networks of large geographic scope.
- Adding a new host to the network by connecting it to a switch does not necessarily reduce the performance of the network for other hosts already connected.



A switch provides a star topology.

This last claim cannot be made for the shared-media networks discussed in the last chapter. For example, it is impossible for two hosts on the same 10-Mbps Ethernet segment to transmit continuously at 10 Mbps because they share the same transmission medium. Every host on a switched network has its own link to the switch, so it may be entirely possible for many hosts to transmit at the full link speed (bandwidth), provided that the switch is designed with enough aggregate capacity. Providing high aggregate throughput is one of the design goals for a switch; we return to this topic later. In general, switched networks are considered more *scalable* (i.e., more capable of growing to large numbers of nodes) than shared-media networks because of this ability to support many hosts at full speed.

A switch is connected to a set of links and, for each of these links, runs the appropriate data link protocol to communicate with the node at the other end of the link. A switch's primary job is to receive incoming packets on one of its links and to transmit them on some other link. This function is sometimes referred to as either *switching* or *forwarding,* and in terms of the Open Systems Interconnection (OSI) architecture, it is the main function of the network layer, otherwise known as *Layer 2*.

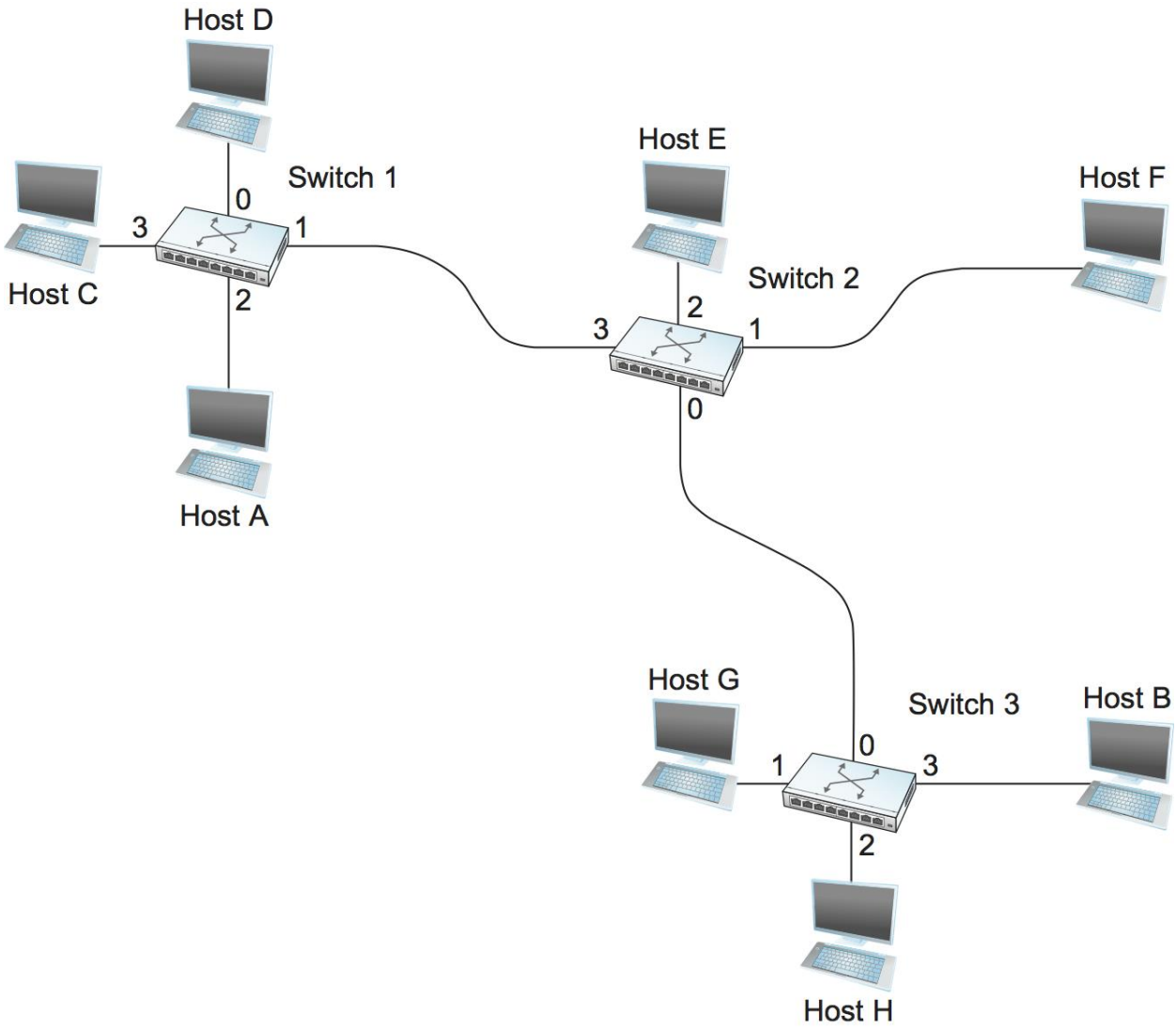
The question, then, is how does the switch decide which output link to place each packet on? The general answer is that it looks at the header of the packet for an identifier that it uses to make the decision. The details of how it uses this identifier vary, but there are two common approaches. The first is the *datagram* or *connectionless* approach. The second is the *virtual circuit* or *connection-oriented* approach. A third approach, *source routing*, is less common than these other two, but it does have some useful applications.

One thing that is common to all networks is that we need to have a way to identify the end nodes. Such identifiers are usually called *addresses*. We have already seen examples of addresses, such as the 48-bit address used for Ethernet. The only requirement for Ethernet addresses is that no two nodes on a network have the same address. This is accomplished by making sure that all Ethernet cards are assigned a *globally unique* identifier. For the following discussion, we assume that each host has a globally unique address. Later on, we consider other useful properties that an address might have, but global uniqueness is adequate to get us started.

Another assumption that we need to make is that there is some way to identify the input and output ports of each switch. There are at least two sensible ways to identify ports: One is to number each port, and the other is to identify the port by the name of the node (switch or host) to which it leads. For now, we use numbering of the ports.

Datagrams

The idea behind datagrams is incredibly simple: You just include in every packet enough information to enable any switch to decide how to get it to its destination. That is, every packet contains the complete destination address. Consider the example network illustrated in Figure 2, in which the hosts have addresses A, B, C, and so on. To decide how to forward a packet, a switch consults a *forwarding table* (sometimes called a *routing table*), an example of which is depicted in Table 1. This particular table shows the forwarding information that switch 2 needs to forward datagrams in the example network. It is pretty easy to figure out such a table when you have a complete map of a simple network like that depicted here; we could imagine a network operator configuring the tables statically. It is a lot harder to create the forwarding tables in large, complex networks with dynamically changing topologies and multiple paths between destinations. That harder problem is known as *routing* and is the topic of a later section. We can think of routing as a process that takes place in the background so that, when a data packet turns up, we will have the right information in the forwarding table to be able to forward, or switch, the packet.



Datagram forwarding: an example network.Forwarding Table for Switch 2.

Destination	Port
A	3
B	0

Destination	Port
C	3
D	3
E	2
F	1
G	0
H	0

Datagram networks have the following characteristics:

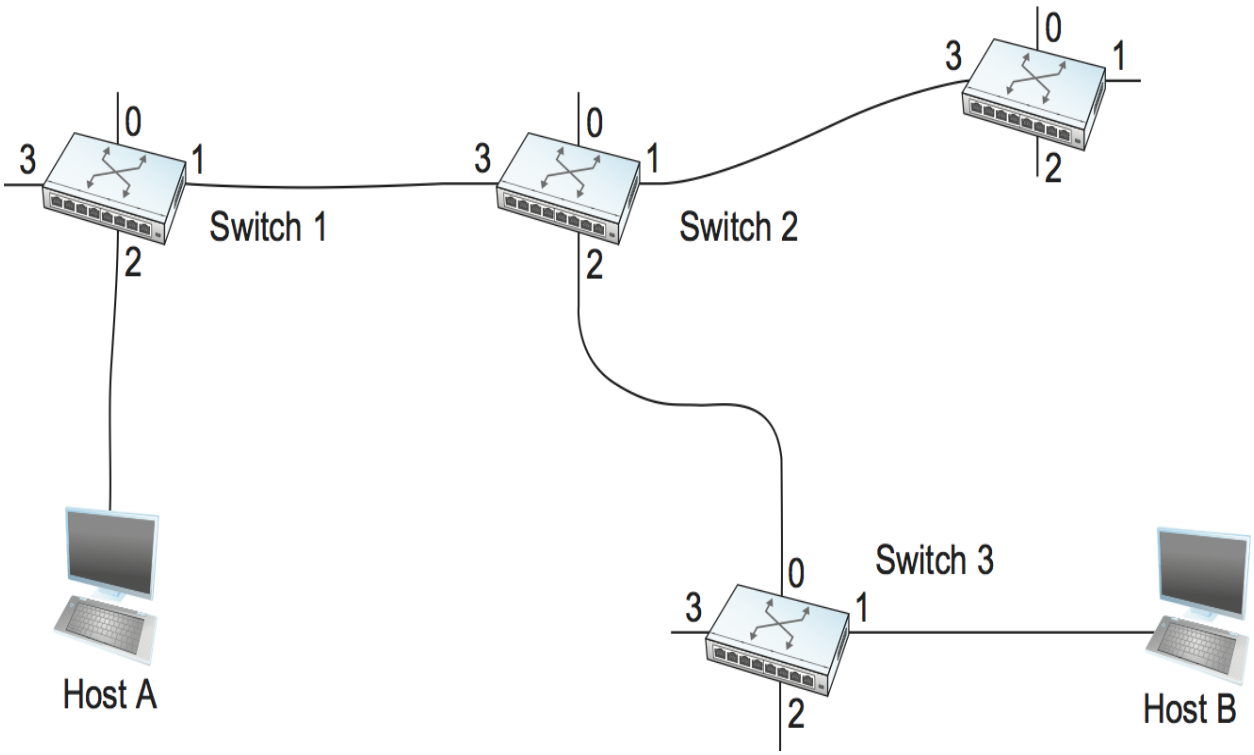
- A host can send a packet anywhere at any time, since any packet that turns up at a switch can be immediately forwarded (assuming a correctly populated forwarding table). For this reason, datagram networks are often called *connectionless*; this contrasts with the *connection-oriented* networks described below, in which some *connection state* needs to be established before the first data packet is sent.
- When a host sends a packet, it has no way of knowing if the network is capable of delivering it or if the destination host is even up and running.
- Each packet is forwarded independently of previous packets that might have been sent to the same destination. Thus, two successive packets from host A to host B may follow completely different paths (perhaps because of a change in the forwarding table at some switch in the network).
- A switch or link failure might not have any serious effect on communication if it is possible to find an alternate route around the failure and to update the forwarding table accordingly.

This last fact is particularly important to the history of datagram networks. One of the important design goals of the Internet is robustness to failures, and history has shown it to be quite effective at meeting this goal.

Virtual Circuit Switching

A second technique for packet switching, which differs significantly from the datagram model, uses the concept of a *virtual circuit* (VC). This approach, which is also referred to as a *connection-oriented model*, requires setting up a virtual connection from the source host to the destination host before any data is sent. To understand how this works, consider Figure 3, where host A again wants to send packets to host B. We can think of this as a two-stage process. The first stage is "connection setup." The second is data transfer. We consider each in turn.

An example of a virtual circuit network.



In the connection setup phase, it is necessary to establish a "connection state" in each of the switches between the source and destination hosts. The connection state for a single connection consists of an entry in a "VC table" in each switch through which the connection passes. One entry in the VC table on a single switch contains:

- A *virtual circuit identifier* (VCI) that uniquely identifies the connection at this switch and which will be carried inside the header of the packets that belong to this connection
- An incoming interface on which packets for this VC arrive at the switch
- An outgoing interface in which packets for this VC leave the switch
- A potentially different VCI that will be used for outgoing packets

The semantics of one such entry is as follows: If a packet arrives on the designated incoming interface and that packet contains the designated VCI value in its header, then that packet

should be sent out the specified outgoing interface with the specified outgoing VCI value having been first placed in its header.

Note that the combination of the VCI of packets as they are received at the switch *and* the interface on which they are received uniquely identifies the virtual connection. There may of course be many virtual connections established in the switch at one time. Also, we observe that the incoming and outgoing VCI values are generally not the same. Thus, the VCI is not a globally significant identifier for the connection; rather, it has significance only on a given link (i.e., it has *link-local scope*).

Whenever a new connection is created, we need to assign a new VCI for that connection on each link that the connection will traverse. We also need to ensure that the chosen VCI on a given link is not currently in use on that link by some existing connection.

There are two broad approaches to establishing connection state. One is to have a network administrator configure the state, in which case the virtual circuit is "permanent." Of course, it can also be deleted by the administrator, so a permanent virtual circuit (PVC) might best be thought of as a long-lived or administratively configured VC. Alternatively, a host can send messages into the network to cause the state to be established. This is referred to as *signalling*, and the resulting virtual circuits are said to be *switched*. The salient characteristic of a switched virtual circuit (SVC) is that a host may set up and delete such a VC dynamically without the involvement of a network administrator. Note that an SVC should more accurately be called a *signalled* VC, since it is the use of signalling (not switching) that distinguishes an SVC from a PVC.

Let's assume that a network administrator wants to manually create a new virtual connection from host A to host B. First, the administrator needs to identify a path through the network from A to B. In the example network of Figure 3, there is only one such path, but in general, this may not be the case. The administrator then picks a VCI value that is currently unused on each link for the connection. For the purposes of our example, let's suppose that the VCI value 5 is chosen for the link from host A to switch 1, and that 11 is chosen for the link from switch 1 to switch 2. In that case, switch 1 needs to have an entry in its VC table configured as shown in Table 2.

Example Virtual Circuit Table Entry for Switch 1.

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	5	1	11

Similarly, suppose that the VCI of 7 is chosen to identify this connection on the link from switch 2 to switch 3 and that a VCI of 4 is chosen for the link from switch 3 to host B. In that case, switches 2 and 3 need to be configured with VC table entries as shown in Table 3. Note that the "outgoing" VCI value at one switch is the "incoming" VCI value at the next switch.

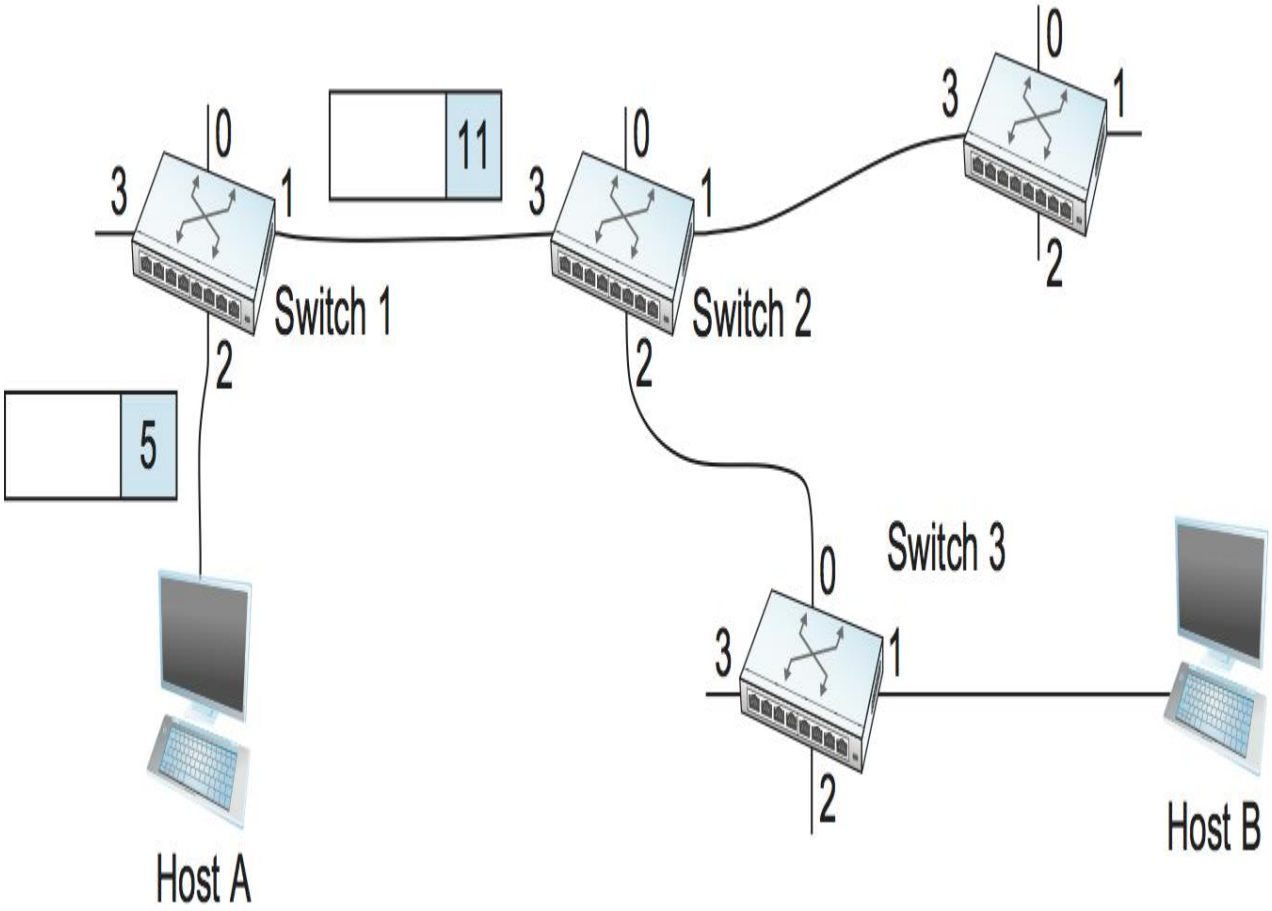
Virtual Circuit Table Entries for Switches 2 and 3.

VC Table Entry at Switch 2:

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
3	11	2	7

VC Table Entry at Switch 3:

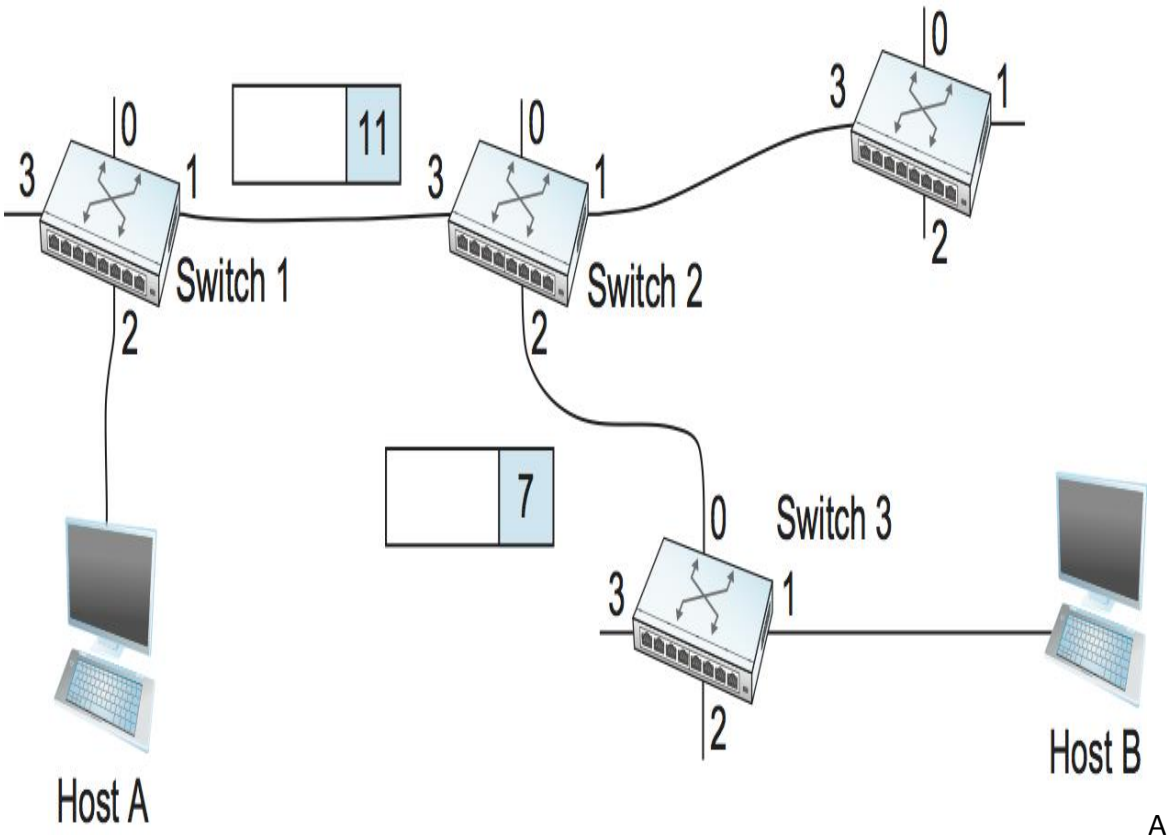
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
0	7	1	4



A packet is sent into a virtual circuit network.

Once the VC tables have been set up, the data transfer phase can proceed, as illustrated in Figure 4. For any packet that it wants to send to host B, A puts the VCI value of 5 in the header of the packet and sends it to switch 1. Switch 1 receives any such packet on interface 2, and it uses the combination of the interface and the VCI in the packet header to find the appropriate VC table entry. As shown in Table 2, the table entry in this case tells switch 1 to forward the packet out of interface 1 and to put the VCI value 11 in the header when the packet is sent. Thus, the packet will arrive at switch 2 on interface 3 bearing VCI 11. Switch 2 looks up interface 3 and VCI 11 in its VC table (as shown in Table 3) and sends the packet on to switch 3 after updating the VCI value in the packet header appropriately, as shown in Figure 5. This process continues until it arrives at host B with the VCI value of 4 in the packet. To host B, this identifies the packet as having come from host A.

In real networks of reasonable size, the burden of configuring VC tables correctly in a large number of switches would quickly become excessive using the above procedures. Thus, either a network management tool or some sort of signalling (or both) is almost always used, even when setting up "permanent" VCs. In the case of PVCs, signalling is initiated by the network administrator, while SVCs are usually set up using signalling by one of the hosts. We consider now how the same VC just described could be set up by signalling from the host.



A packet makes its way through a virtual circuit network.

To start the signalling process, host A sends a setup message into the network—that is, to switch 1. The setup message contains, among other things, the complete destination address of host B. The setup message needs to get all the way to B to create the necessary connection state in every switch along the way. We can see that getting the setup message to B is a lot like getting a datagram to B, in that the switches have to know which output to send the setup message to so that it eventually reaches B. For now, let's just assume that the switches know enough about the network topology to figure out how to do that, so that the setup message flows on to switches 2 and 3 before finally reaching host B.

When switch 1 receives the connection request, in addition to sending it on to switch 2, it creates a new entry in its virtual circuit table for this new connection. This entry is exactly the same as shown previously in Table 2. The main difference is that now the task of assigning an unused VCI value on the interface is performed by the switch for that port. In this example, the switch picks the value 5. The virtual circuit table now has the following information: "When packets arrive on port 2 with identifier 5, send them out on port 1." Another issue is that, somehow, host A will need to learn that it should put the VCI value of 5 in packets that it wants to send to B; we will see how that happens below.

When switch 2 receives the setup message, it performs a similar process; in this example, it picks the value 11 as the incoming VCI value. Similarly, switch 3 picks 7 as the value for its

incoming VCI. Each switch can pick any number it likes, as long as that number is not currently in use for some other connection on that port of that switch. As noted above, VCIs have link-local scope; that is, they have no global significance.

Finally, the setup message arrives as host B. Assuming that B is healthy and willing to accept a connection from host A, it too allocates an incoming VCI value, in this case 4. This VCI value can be used by B to identify all packets coming from host A.

Now, to complete the connection, everyone needs to be told what their downstream neighbor is using as the VCI for this connection. Host B sends an acknowledgment of the connection setup to switch 3 and includes in that message the VCI that it chose (4). Now switch 3 can complete the virtual circuit table entry for this connection, since it knows the outgoing value must be 4. Switch 3 sends the acknowledgment on to switch 2, specifying a VCI of 7. Switch 2 sends the message on to switch 1, specifying a VCI of 11. Finally, switch 1 passes the acknowledgment on to host A, telling it to use the VCI of 5 for this connection.

At this point, everyone knows all that is necessary to allow traffic to flow from host A to host B. Each switch has a complete virtual circuit table entry for the connection. Furthermore, host A has a firm acknowledgment that everything is in place all the way to host B. At this point, the connection table entries are in place in all three switches just as in the administratively configured example above, but the whole process happened automatically in response to the signalling message sent from A. The data transfer phase can now begin and is identical to that used in the PVC case.

When host A no longer wants to send data to host B, it tears down the connection by sending a teardown message to switch 1. The switch removes the relevant entry from its table and forwards the message on to the other switches in the path, which similarly delete the appropriate table entries. At this point, if host A were to send a packet with a VCI of 5 to switch 1, it would be dropped as if the connection had never existed.

There are several things to note about virtual circuit switching:

- Since host A has to wait for the connection request to reach the far side of the network and return before it can send its first data packet, there is at least one round-trip time (RTT) of delay before data is sent.
- While the connection request contains the full address for host B (which might be quite large, being a global identifier on the network), each data packet contains only a small identifier, which is only unique on one link. Thus, the per-packet overhead caused by the header is reduced relative to the datagram model. More importantly, the lookup is fast because the virtual circuit number can be treated as an index into a table rather than as a key that has to be looked up.
- If a switch or a link in a connection fails, the connection is broken and a new one will need to be established. Also, the old one needs to be torn down to free up table storage space in the switches.

- The issue of how a switch decides which link to forward the connection request on has been glossed over. In essence, this is the same problem as building up the forwarding table for datagram forwarding, which requires some sort of *routing algorithm*. Routing is described in a later section, and the algorithms described there are generally applicable to routing setup requests as well as datagrams.

Spanning Tree Protocol (STP)

What is STP and how does it work?

Spanning Tree Protocol (STP) is a Layer 2 network protocol used to prevent looping within a network topology. STP was created to avoid the problems that arise when computers exchange data on a local area network (LAN) that contains redundant paths. If the flow of traffic is not carefully monitored and controlled, the data can be caught in a loop that circles around network segments, affecting performance and bringing traffic to a near halt.

Networks are often configured with redundant paths when connecting network segments. Although redundancy can help protect against disaster, it can also lead to bridge or switch looping. Looping occurs when data travels from a source to a destination along redundant paths and the data begins to circle around the same paths, becoming amplified and resulting in a broadcast storm.

STP can help prevent bridge looping on LANs that include redundant links. Without STP, it would be difficult to implement that redundancy and still avoid network looping. STP monitors all network links, identifies redundant connections and disables the ports that can lead to looping.

LANs are often divided into multiple network segments, and they use bridges to connect the individual segment pairs. Each message, called a frame, goes through the bridge before being sent to the intended destination. The bridge determines whether the message is for a destination within the same segment as the sender's or for another segment and then forwards the message accordingly. When used in the context of STP, the term *bridge* can also refer to a network switch.

A bridge looks at the destination address and, based on its understanding of which computers are on which segments, forwards the data on the right path via the correct outgoing port. Network segmentation and bridging can reduce the amount of competition for a network path by half -- assuming each segment has the same number of computers. As a result, the network is much less likely to come to a halt.

A segmented LAN is often designed with redundant bridges and paths to ensure that communications can continue in the event that a network link becomes unavailable. However, this makes the network more susceptible to looping, so a system must be put into place to prevent this possibility, which is where STP comes in.

When STP is enabled, each bridge learns which computers are on which segment by sending a first-time message to network segments. Through this process, the bridge discovers the computers' locations and records the details in a table. When subsequent messages are sent, the bridge uses the table to determine which segment to forward them to. Enabling the bridge to learn about the network on

its own is known as *transparent bridging*, a process that eliminates the need for an administrator to set up bridging manually.

In a network that contains redundant paths, bridges need to continually understand the topology of the network to control the flow of traffic and prevent looping. To do this, they exchange bridge protocol data units (BPDUs) via an extended LAN that uses a spanning tree protocol. BPDUs are data messages that provide the bridges with network information that's used to carry out STP operations.

At the heart of STP is the spanning tree algorithm that runs on each STP-enabled bridge. The algorithm was specifically designed to avoid bridge loops when redundant paths exist. It uses the BPDUs to identify redundant links and select the best data path for forwarding messages. The algorithm also controls packet forwarding by setting the port state.

What are STP port states?

When STP is enabled on a network bridge, each port is set to one of five states to control frame forwarding:

1. **Disabled.** The port does not participate in frame forwarding or STP operations.
2. **Blocking.** The port does not participate in frame forwarding and discards frames received from the attached network segment. However, the port continues to listen for and process BPDUs.
3. **Listening.** From the blocking state, the port transitions to the listening state. The port discards frames from the attached network segment or forwarded from another port. However, it receives BPDUs and redirects them to the switch module for processing.

4. **Learning.** The port moves from the listening state to the learning state. It listens for and processes BPDUs but discards frames from the attached network segment or forwarded from another port. It also starts updating the address table with the information it's learned. In addition, it processes user frames but does not forward those frames.

5. **Forwarding.** The port moves from the learning state to the forwarding state and starts forwarding frames across the network segments. This includes frames from the attached network segment and those forwarded from another port. The port also continues to receive and process BPDUs, and the address table continues to be updated.

LAN Switch types- Cut Through

Cut-through is a packet-switching method, where the switch forwards a packet as soon as the destination address is processed without waiting for the entire packet to be received. The next packet is sent as soon as the previous one has been verified as reaching the recipient without waiting for the complete transmission of the previous packet.

Working :

In cut-through switching, whenever a packet arrives at the switching device, data transmission is started as soon as the destination address is processed. The switch performs a look–up operation in the address table to check whether the destination address is valid or not. If the address is valid and the outgoing link is available then the switching device immediately transmits the frame to the destination port.

Switch use Cyclic Redundancy Check (CRC) on incoming packets for error detection and marks corrupted frame EOF field as invalid. It relies on the destination devices for error handling of the corrupted data. The destination devices detect the invalid flag and drop the frame.

Types :

Primarily Cut-through switching is characterized into two types:

1. Rapid Frame Forwarding :

- Switch forwards the frame as soon as it has looked up the destination MAC address of the frame in its MAC address table. It does not wait to receive the rest of the frame.
- It has low latency and high data transfer speed.
- It has a high error rate because frames with errors are forwarded to other segments.

2. Fragment Free :

- Switch waits for the collision window (64 bytes) to pass before forwarding the frame. It checks the data field to ensure no fragmentation has occurred.
- It has a comparatively lesser data transfer speed.
- It has a low error rate because it monitors the integrity of each frame while forwarding.

Advantages :

1. **Low- latency –**
   Switching device does not wait for the entire packet to arrive for transmission which reduces the latency (time required to process the data) to pass through the switch.
2. **High speed –**
   Low latency leads to higher transfer speeds i.e. speed at which data packets are transmitted.
3. **Less storage requirement –**
   Switching devices do not require storing of data packets which reduces the internal storage requirement.

Disadvantage :

1. **Integrity issues –**
   Switches forward corrupted frames since they don't wait to check if the checksum at the end of each frame is valid or not.

Uses :

1. Fiber channel communication
2. SCSI traffic transmission
3. ATM networks
4. InfiniBand networks
5. Bitcoin

Fragment Free, Store-and-forward:-

"Fragment Free" and "Store-and-forward" are two different network transmission modes used in networking and data communication. Let me explain each of them:

1. Fragment Free:

Fragment Free is a specific transmission mode used in network switches for data transmission. It is a variation of the Ethernet switching method that attempts to strike a balance between the two common transmission modes: "Cut-Through" and "Store-and-forward" (which I'll explain next).

In "Fragment Free" mode, the network switch performs some level of error checking before forwarding data frames. When a frame arrives at the switch, it checks the first 64 bytes of the frame, known as the preamble and the first 64 bytes of the frame, known as the header. The switch can determine whether the frame has a minimum size (typically 64 bytes). If the frame appears to be valid (non-damaged), it will forward the frame. The idea behind Fragment Free is to avoid forwarding frames that are excessively short, which could potentially indicate collisions or other transmission issues.

2. Store-and-forward:

Store-and-forward is a different network transmission mode commonly used in networking switches and routers. In this mode, when a data frame arrives at the switch or router, the device stores the entire frame in its memory and checks the frame for errors and integrity before forwarding it to the next destination. The switch/router checks the frame's integrity by verifying the frame's checksum or CRC (Cyclic Redundancy Check) value.

Store-and-forward is a more thorough transmission method compared to "Cut-Through" (another transmission mode) as it waits for the entire frame to be received and checked before forwarding it. This method ensures that corrupt or

damaged frames are not propagated further into the network, reducing the likelihood of errors being propagated through the network.

In summary, "Fragment Free" and "Store-and-forward" are different modes of data transmission within a network switch or router. Fragment Free aims to avoid forwarding excessively short frames, while Store-and-forward ensures the frame's integrity is verified before forwarding it to its destination.

Configuration of Switches:-

Just like riding a bicycle, nobody's born knowing how to setup a network switch. And this process is a little more advanced than, say, setting up your home Internet or even a plug-and-play type switch. But, with the right guidance, a can-do attitude, and a dash of bravery, even newbie IT professionals can integrate a new Cisco switch into their business environment. As your virtual training wheels, we've broken down the task into its simplest parts so you can successfully create client VLANS, build DHCP systems, and assign access ports without skinning your knees.

Step 1: Inspect your hardware

Check the model number of your shiny new switch. Or, if you are using a spare, check the device hardware and its connected cables for any damages. If everything checks out, power on the switch and verify that all the indicator lights are in working order. Next, use a rollover cable to console into the switch from your computer. To do this, you will need to download and install Putty (or a similar, fun-named software tool). Run Putty and select the 9600 speed serial connection. You are now connected to the switch and ready to check the output of the following commands:

- show version
- show running-config
- show VLAN brief
- show VTP status

- (config)# IP domain-name routerfreak.com
- (config)# hostname Switch01
- (config)# interface VLAN1
- (config)# description Management VLAN
- (config)# IP address 192.168.101.1 255.255.255.0
- vtp [client | server | transparent]
- vtp domain name
- description *** DESCRIPTION ***
- switchport access vlan ###
- sswitchport mode access
- power inline consumption ###
- queue-set 2
- mls qos trust dscp
- storm-control multicast level 50.00
- no cdp enable
- spanning-tree portfast
- spanning-tree bpduguard enable
- Interface GigabitEthernet1/0/1
- description *** UPLINK ***
- switchport trunk encapsulation dot1q
- switchport mode trunk
- speed 1000
- duplex full
- Switch01(config)# crypto key generate rsa
- The name for the keys will be:
- Switch01.routerfreak.com
- How many bits in the modulus [512]: 1024
- % Generating 1024 bit RSA keys, keys will be non-exportable...[OK]
- # line vty 0 4

- (config-line)# transport input ssh
- (config-line)# login local
- (config-line)# password routerfreak
- (config-line)# exit
- # line console 0
- (config-line)# logging synchronous
- (config-line)# login local
- Switch01# service password-encryption
- remote-computer# ssh 192.168..101.1
- Log in as: username
- Password:
- Switch01>en
- Password:
- Switch01#

For spare switches, make sure to delete the flash:vlan.dat file to erase the previous configuration.

Step 2: Set up management IP

Unlike with that punny name you gave your home Wi-Fi network, when setting up the hostname for your switch you should probably stick to a more professional and standard naming convention. Follow any preset naming assignment your company is using and then assign an IP address on the management VLAN. Next, make sure your switch has a set hostname and domain name:

Step 3: Check VTP revision number

Hit the show vtp status command to reveal your Virtual Trunking Protocol (VTP) revision numbers. The VTP revision numbers determine which updates are to be used in a VTP domain. When you set a VTP domain name, the revision number is set to zero—after which each change to the VLAN database increases the revision number by one. Your switch will only process data from a neighboring switch coming from the same domain and if the revision number of the neighboring switch is higher than its own. This means that the switches will update their VLAN configuration based on the VTP information being sent by the switch with the highest revision number.

So, before you add your switch to the network, you're going to want to set its revision number to zero. To easily reset the domain back to zero, change the config mode to transparent:

Step 4: Configure access ports

You might already have a template ready for access port configuration, but in case you don't, here are some commands you should use:

Step 5: Configure trunk ports

Enter the command sh int g0/1 capabilities and check the trunking protocol supported. If ISL is supported, you have to issue the switchport trunk encapsulation dot1q on the trunk port configuration. If not, simply type switchport mode trunk. It means there is no other encapsulation supported so there is no need for an encapsulation command. It only supports 802.1Q.

Step 6: Configure access ports

After already performing basic network switch configurations, it's time to generate RSA keys to be used during the SSH process, using the crypto commands shown here:

Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys. Choosing a key modulus greater than 512 may take a few minutes.

Step 7: Set up VTY line config

If you have not set the console line yet, you can easily input these values:

Set the enable password using the enable secret password command. Then, set the privilege exec password with username name privilege 15 secret password. Make sure that the password-encryption service is activated.

Verify SSH access by typing 'sh ip ssh' to confirm that the SSH is enabled. You can now try to log in from a remote machine to verify that you can ssh to your Cisco switch.

Finishing touches

You've made it through the learning process with (hopefully) minimum bumps and bruises, and you're just about ready to ride off. All that's left is to test your access,

reload the switch, and ready the cables. Once that's done, label your switch, rack it up, and go enjoy doing anything that doesn't involve switch configuration!

Virtual LANs:-

Virtual LAN (VLAN) is a concept in which we can divide the devices logically on layer 2 (data link layer). Generally, layer 3 devices divide the broadcast domain but the broadcast domain can be divided by switches using the concept of VLAN.
A broadcast domain is a network segment in which if a device broadcast a packet then all the devices in the same broadcast domain will receive it. The devices in the same broadcast domain will receive all the broadcast packets but it is limited to switches only as routers don't forward out the broadcast packet. To forward out the packets to different VLAN (from one VLAN to another) or broadcast domains, inter Vlan routing is needed. Through VLAN, different small-size sub-networks are created which are comparatively easy to handle.

VLAN ranges:

- **VLAN 0, 4095:** These are reserved VLAN which cannot be seen or used.
- **VLAN 1:** It is the default VLAN of switches. By default, all switch ports are in VLAN. This VLAN can't be deleted or edit but can be used.
- **VLAN 2-1001:** This is a normal VLAN range. We can create, edit and delete these VLAN.
- **VLAN 1002-1005:** These are CISCO defaults for fddi and token rings. These VLAN can't be deleted.
- **Vlan 1006-4094:** This is the extended range of Vlan.

Configuration –
We can simply create VLANs by simply assigning the vlan-id and Vlan name.
#switch1(config)#vlan 2

#switch1(config-vlan)#vlan accounts

Here, 2 is the Vlan I'd and accounts is the Vlan name. Now, we assign Vlan to the switch ports.e.g-

Switch(config)#int fa0/0

Switch(config-if)#switchport mode access

Switch(config-if)#switchport access Vlan 2

Also, switchport range can be assigned to required vlans.

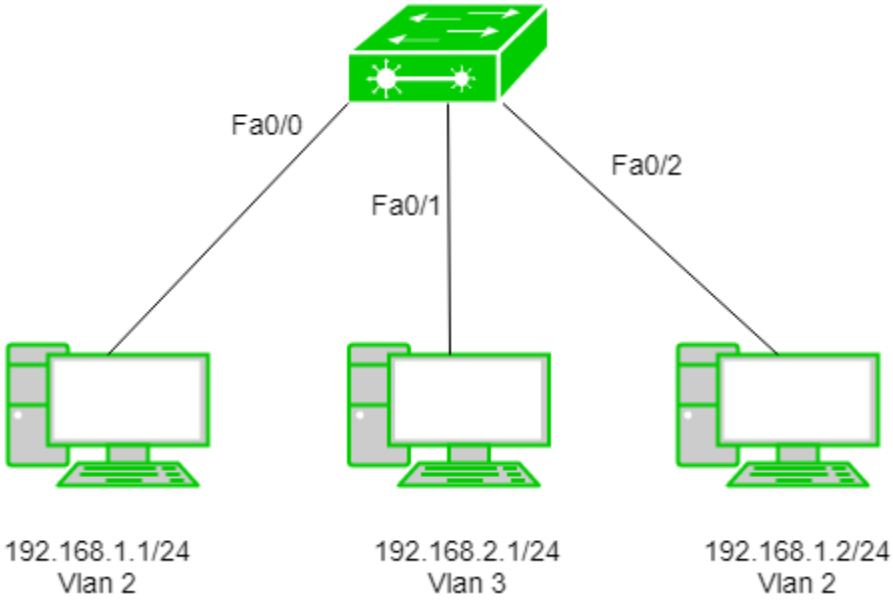
Switch(config)#int range fa0/0-2

Switch(config-if)#switchport mode access

Switch(config-if) #switchport access Vlan 2

By this, switchport fa0/0, fa0/1, fa0-2 will be assigned Vlan 2.

Example –



Assigning IP address 192.168.1.1/24, 192.168.1.2/24 and 192.168.2.1/24 to the PC's. Now, we will create Vlan 2 and 3 on switch.

Switch(config)#vlan 2

Switch(config)#vlan 3

We have made VLANs but the most important part is to assign switch ports to the VLANs.

Switch(config)#int fa0/0

Switch(config-if)#switchport mode access

Switch(config-if) #switchport access Vlan 2

Switch(config)#int fa0/1

Switch(config-if)#switchport mode access

Switch(config-if) #switchport access Vlan 3

Switch(config)#int fa0/2

Switch(config-if)#switchport mode access

Switch(config-if) #switchport access Vlan 2

As seen, we have assigned Vlan 2 to fa0/0, fa0/2, and Vlan 3 to fa0/1.

VLANs offer several features and benefits, including:

- **Improved network security:** VLANs can be used to separate network traffic and limit access to specific network resources. This improves security by preventing unauthorized access to sensitive data and network resources.
- **Better network performance:** By segregating network traffic into smaller logical networks, VLANs can reduce the amount of broadcast traffic and improve network performance.
- **Simplified network management:** VLANs allow network administrators to group devices together logically, rather than physically, which can simplify network management tasks such as configuration, troubleshooting, and maintenance.
- **Flexibility:** VLANs can be configured dynamically, allowing network administrators to quickly and easily adjust network configurations as needed.
- **Cost savings:** VLANs can help reduce hardware costs by allowing multiple virtual networks to share a single physical network infrastructure.
- **Scalability:** VLANs can be used to segment a network into smaller, more manageable groups as the network grows in size and complexity.

Some of the key features of VLANs include:

- **VLAN tagging:** VLAN tagging is a way to identify and distinguish VLAN traffic from other network traffic. This is typically done by adding a VLAN tag to the Ethernet frame header.
- **VLAN membership:** VLAN membership determines which devices are assigned to which VLANs. Devices can be assigned to VLANs based on port, MAC address, or other criteria.
- **VLAN trunking:** VLAN trunking allows multiple VLANs to be carried over a single physical link. This is typically done using a protocol such as IEEE 802.1Q.
- **VLAN management:** VLAN management involves configuring and managing VLANs, including assigning devices to VLANs, configuring VLAN tags, and configuring VLAN trunking.

Types of connections in VLAN –

There are three ways to connect devices on a VLAN, the type of connections are based on the connected devices i.e. whether they are VLAN-aware(A device that understands VLAN formats and VLAN membership) or VLAN-unaware(A device that doesn't understand VLAN format and VLAN membership).

1. **Trunk Link –**
   All connected devices to a trunk link must be VLAN-aware. All frames on this should have a special header attached to it called tagged frames.
2. **Access link –**
   It connects VLAN-unaware devices to a VLAN-aware bridge. All frames on the access link must be untagged.
3. **Hybrid link –**
   It is a combination of the Trunk link and Access link. Here both VLAN-unaware and VLAN-aware devices are attached and it can have both tagged and untagged frames.

Advantages –

- **Performance –**
  The network traffic is full of broadcast and multicast. VLAN reduces the need to send such traffic to unnecessary destinations. e.g.-If the traffic is intended for 2 users but as 10 devices are present in the same broadcast domain, therefore, all will receive the traffic i.e. wastage of bandwidth but if

we make VLANs, then the broadcast or multicast packet will go to the intended users only.

- **Formation of virtual groups –**
  As there are different departments in every organization namely sales, finance etc., VLANs can be very useful in order to group the devices logically according to their departments.
- **Security –**
  In the same network, sensitive data can be broadcast which can be accessed by the outsider but by creating VLAN, we can control broadcast domains, set up firewalls, restrict access. Also, VLANs can be used to inform the network manager of an intrusion. Hence, VLANs greatly enhance network security.
- **Flexibility –**
  VLAN provide flexibility to add, remove the number of host we want.
- **Cost reduction –**
  VLANs can be used to create broadcast domains which eliminate the need for expensive routers.
  By using Vlan, the number of small size broadcast domain can be increased which are easy to handle as compared to a bigger broadcast domain.

Disadvantages of VLAN

1. **Complexity:** VLANs can be complex to configure and manage, particularly in large or dynamic cloud computing environments.
2. **Limited scalability:** VLANs are limited by the number of available VLAN IDs, which can be a constraint in larger cloud computing environments.
3. **Limited security**: VLANs do not provide complete security and can be compromised by malicious actors who are able to gain access to the network.
4. **Limited interoperability**: VLANs may not be fully compatible with all types of network devices and protocols, which can limit their usefulness in cloud computing environments.
5. **Limited mobility**: VLANs may not support the movement of devices or users between different network segments, which can limit their usefulness in mobile or remote cloud computing environments.
6. **Cost:** Implementing and maintaining VLANs can be costly, especially if specialized hardware or software is required.
7. **Limited visibility:** VLANs can make it more difficult to monitor and troubleshoot network issues, as traffic is isolated in different segments.

Real-Time Applications of VLAN

Virtual LANs (VLANs) are widely used in cloud computing environments to improve network performance and security. Here are a few examples of real-time applications of VLANs:

1. **Voice over IP (VoIP)** : VLANs can be used to isolate voice traffic from data traffic, which improves the quality of VoIP calls and reduces the risk of network congestion.
2. **Video Conferencing** : VLANs can be used to prioritize video traffic and ensure that it receives the bandwidth and resources it needs for high-quality video conferencing.
3. **Remote Access** : VLANs can be used to provide secure remote access to cloud-based applications and resources, by isolating remote users from the rest of the network.
4. **Cloud Backup and Recovery** : VLANs can be used to isolate backup and recovery traffic, which reduces the risk of network congestion and improves the performance of backup and recovery operations.
5. **Gaming** : VLANs can be used to prioritize gaming traffic, which ensures that gamers receive the bandwidth and resources they need for a smooth gaming experience.
6. **IoT** : VLANs can be used to isolate Internet of Things (IoT) devices from the rest of the network, which improves security and reduces the risk of network congestion.

Routing and Configuration of VLAN:-

Routing and configuration of VLANs (Virtual Local Area Networks) are essential aspects of network design and management. VLANs allow you to logically segment a physical network into multiple virtual networks, which can improve security, manageability, and overall network performance. Below are the key steps involved in routing and configuring VLANs:

1. VLAN Creation:

- Access your network switch or router's management interface. Most modern network devices support VLAN configuration through a web-based GUI or a command-line interface (CLI).

- Create the VLANs you need. Assign each VLAN a unique identifier called a VLAN ID (VLAN number) and a name to identify its purpose.

2. Assigning Ports to VLANs:

- Determine which physical ports on the switch or router should be part of each VLAN. These ports are known as "access ports."

- Configure each access port to belong to a specific VLAN.

3. Trunk Ports:

- Trunk ports are used to carry traffic for multiple VLANs between switches or routers. These ports need to be configured to allow traffic for multiple VLANs, and they tag the frames with VLAN IDs.

- Make sure to configure the same VLANs on both ends of the trunk link (inter-switch or inter-router links).

4. VLAN Membership Modes:

- Depending on the switch manufacturer and model, there are different VLAN membership modes for access ports:

  - **Untagged**: The port allows traffic for a single VLAN and does not tag the frames with VLAN IDs. Typically used for end devices (e.g., computers, printers) that do not understand VLAN tags.

  - **Tagged**: The port allows traffic for multiple VLANs and tags each frame with the appropriate VLAN ID. Usually used for trunk ports between switches or routers.

5. IP Addressing and Routing:

- For communication between devices on different VLANs, you need to configure IP addressing and routing.

- Assign a subnet to each VLAN. This means each VLAN will have its own IP address range.

- Configure the router or Layer 3 switch with IP addresses on each VLAN interface (SVI - Switched Virtual Interface).

- Set up inter-VLAN routing to allow traffic to flow between VLANs. This can be done through routing protocols or static routes.

6. Security:

- VLANs can be used to improve network security by segregating sensitive or critical devices from the rest of the network.

- Use features like VLAN Access Control Lists (VACLs) or VLAN-based firewall rules to control traffic between VLANs.

7. Testing:

- After configuring VLANs and routing, test the connectivity between devices in different VLANs to ensure everything is working as expected.

It's essential to plan your VLAN design carefully and follow best practices to achieve a secure and efficient network configuration. Also, keep in mind that the exact steps and commands may vary depending on the brand and model of the networking equipment you are using.