Layer 2 Switches

Switch Architecture Facts

Switch Roles

In network environments that contain thousands of interconnected devices, multiple network switches are required. Each of these switches fulfill one of the following roles:

Type	Description
Access switch	An *access switch* is a switch that gives users access to the local area network. User devices are connected to access switches. These switches send data to and from specific computers or nodes that are connected to them. In an office building, each floor may contain one or more access switches with cables that run from the switch to individual rooms or cubicles.
Distribution switch	A *distribution switch* is a switch that resides in the distribution layer and connects to the access and core switches. Typically, each access switch is connected to a distribution switch using one more ports or uplinks. Multiple connections not only increase redundancy, but also increase the maximum bandwidth between the switches. In addition, access switches are oftentimes connected to multiple distribution switches. Distribution switches are linked to each other via high-speed connections, often a 10 Gb Ethernet connection. The number of distribution switches in a network depends on the number of connected devices and the number of access switches in the network.
Core switch	A *core switch* is a switch that resides in the core layer of a two-tier architecture. Most core switches are typically placed in the same location as other distribution switches and connected to the distribution switches.

Switch Architectures

The following table describes switch architectures:

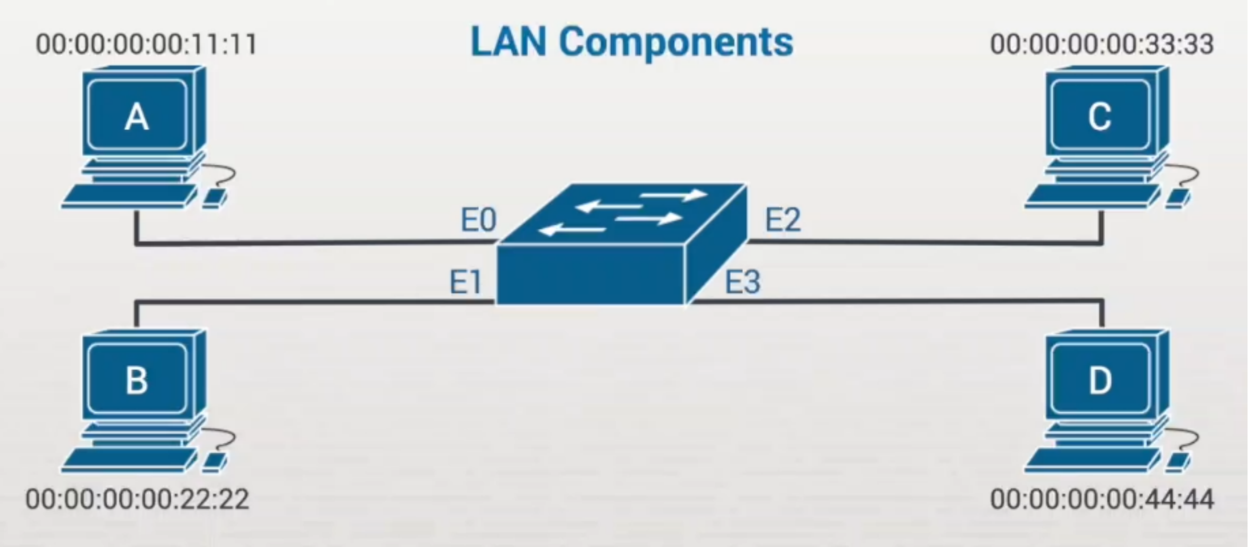
Architecture	Description
Two-tier	A *two-tier architecture*, also known as a collapsed core design, is a switch architecture designed with an access layer, which contains access switches, and a distribution layer, which contains distribution switches. Hundreds or thousands of computers connect to numerous access switches that are connected to multiple distribution switches. There are redundant links between the access switches and the distribution switches, and between the distribution switches themselves. Access to the outside network is managed through one or more routers connected to the distribution layer. The reason this design is sometimes referred to as the collapsed core design is because it doesn't have a distinct third layer, which is called the core layer. The two-tier architecture is commonly used in enterprise networks.
Three-tier	A *three-tier architecture* is a switch architecture designed with an access layer, a distribution layer, and a core layer. The purpose of this core layer is to connect the distribution switches. Connections between core switches and distribution switches, and between multiple core switches, are typically high speed and provide high forwarding rates throughout the campus network. By using a core design, redundant connectivity between distribution switches is still maintained. However, the number of uplinks required to connect each building is reduced considerably. This reduces the cost of connecting buildings. The three-tier architecture is commonly used in enterprise networks.

Two-tier spine-and-leaf	A *two-tier spine-and-leaf architecture* is a switch architecture designed with a leaf layer and a spine layer used in data centers. In the leaf layer, every access switch is connected to each of the switches in the spine layer creating a full mesh topology. The spine layer is made up of switches that perform routing and is the backbone of the network. The spine-and-leaf topology is used in data centers that experience more east-west network traffic than north-south traffic. The traffic load is evenly distributed among the top-tier switches because the traffic path is randomly chosen. If one of the top-tier switches fails, the performance is slightly degraded throughout the data center. To increase capacity to handle more traffic, you can add new leaf switches by connecting them to each spine switch. This allows you to easily scale and expand your network.
Three-tier spine-and-leaf	A *three-tier spine-and-leaf architecture* is a switch architecture designed with a leaf layer, a spine layer, and a core layer used in data centers. The core layer consists of routers added to the leaf and spine layers. A spine-and-leaf architecture allows data flows to take shortcuts from where data is, to where it is going. Data flows within a leaf-spine take the same number of hops on the network no matter where the source and destination are. This is because a leaf-spine architecture is fully-meshed where a three-tier model is partially meshed. While many may think that a fully-meshed architecture creates too many physical interconnects to manage, large Ethernet links reduce the number of physical ports needed. More data can flow across a single link instead of needing multiple links to carry the same load.

Switch Operations

A LAN is composed of three basic components.

- **Network Nodes:** First, we have our individual network nodes--workstations, notebooks, servers, printers, and so on.
- **Central Connecting Device:** Then we have a central connecting device, such as a switch, and we have to connect it to each of the network nodes.
- **Network Media:** To do this, we need some type of network media. This could be Ethernet cabling, fiber optic cabling, or radio waves.



- **MAC Addresses:** The Mac address of the Ethernet card in station A is 1111. Station B is 2222, and so on.
- **Ethernet Port:** You can see that station A connects to the first port on the switch. The first port on most Cisco devices is labeled 0, so we'll label this port E0. E1 is the second Ethernet port, and so on. Each of these ports, E0 through E3, is connected to a different workstation.

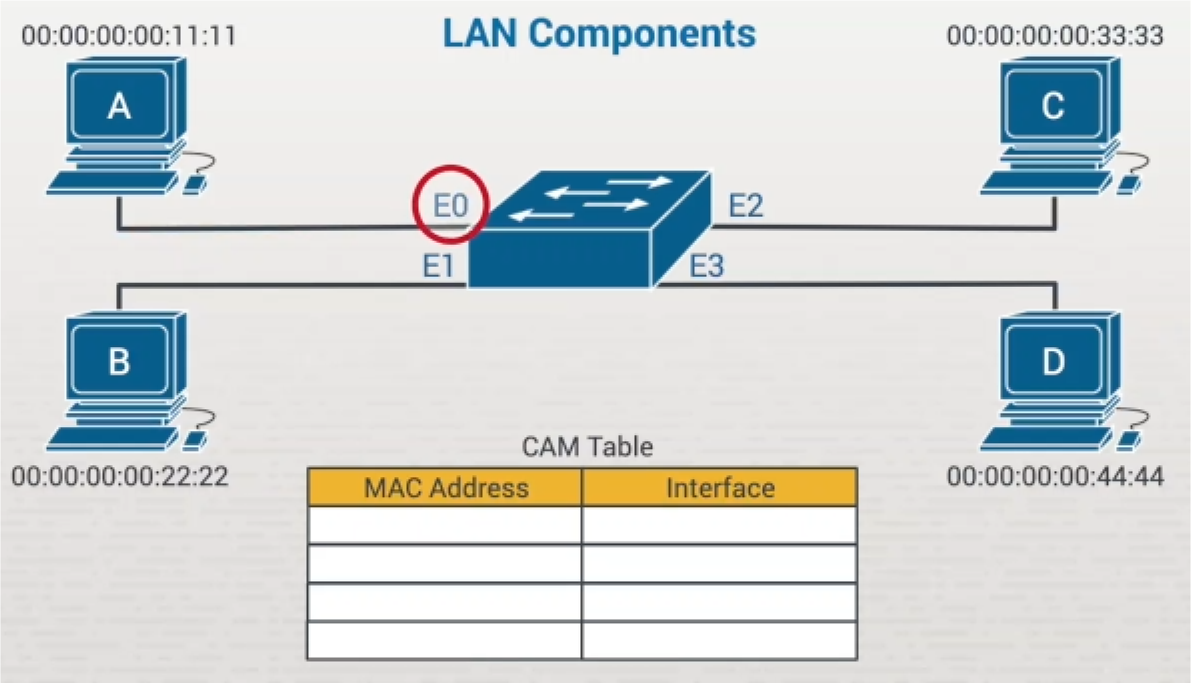
- **Ingress**: When frames enter an interface, that's called ingress.
- **Egress:** When frames exit an interface, it's called egress.
- A switch will never send traffic out of the same interface that it was received on.

CAM Table

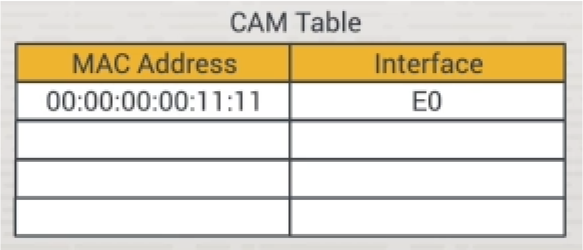
- **Learn and Forward Method:** The switch uses this table to complete a two-step process called the learn and forward method. When a switch is turned on, the Content Addressable Memory table, or CAM table, is empty. Before it can forward packets, the switch needs to learn about the MAC addresses connected to each of its interfaces. The switch examines source addresses and adds them to the CAM. If a destination MAC address is in the CAM table already, it's forwarded out the associated port.

- **Flooding:** If the destination isn't in the CAM table, it's sent through all interfaces except the one that it was received on. This is called flooding.

Example

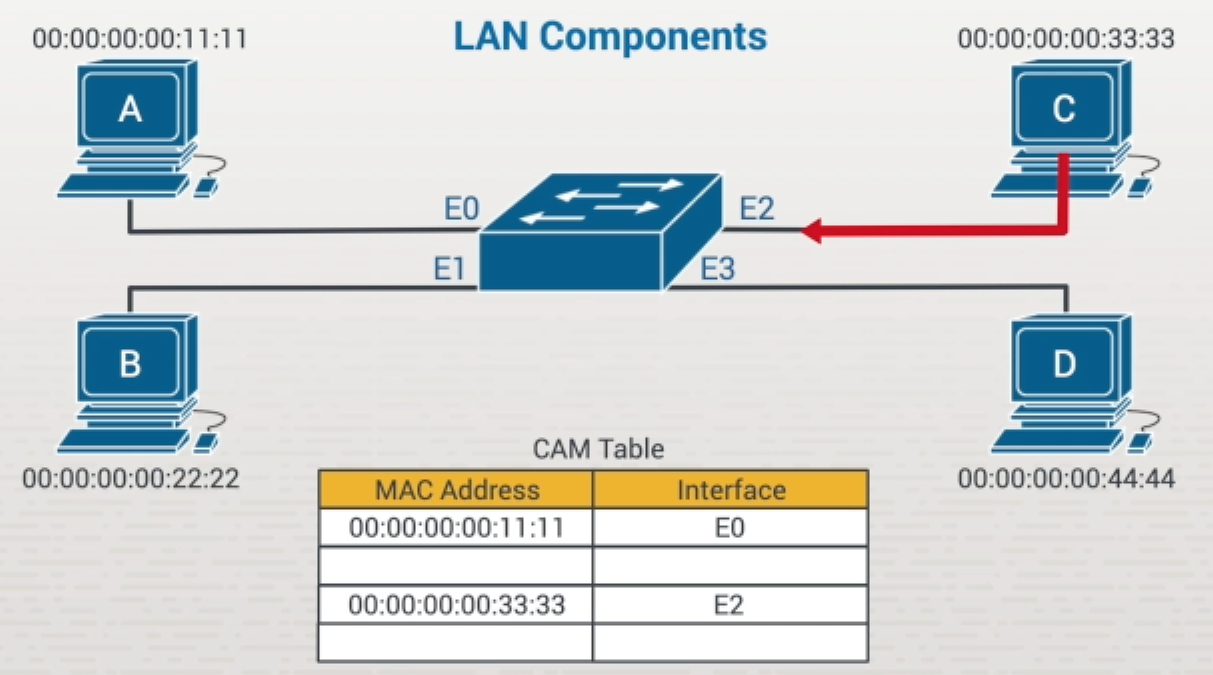


Let's assume that station A has just come up on the network and hasn't received data from any workstations yet. When Station A boots up, it sends out frames. A frame that's sent from Station A arrives on port E0. The frame contains a source MAC address of 1111. Now the switch has learned that a device with the MAC address of 1111 can be accessed through port E0. This MAC address is added to the table alongside port E0.


CAM Table

MAC Address	Interface
00:00:00:00:11:11	E0

The switch only has one entry in the CAM table, the frame's ingress, so it has no idea where to send the packet. To solve this problem, it's going to flood that frame out over every other connection. It's trying to find the host that A is communicating with. Now, let's say that A was trying to communicate with MAC address 3333, which is Station C. Stations B, C, and D receive the frame. Because B and D aren't the intended recipients, they just throw the data away. Station C receives the frame and replies to it. When Station C replies, it'll have a source MAC address of 3333. At this point, the switch will learn that 3333 is connected to port E2.

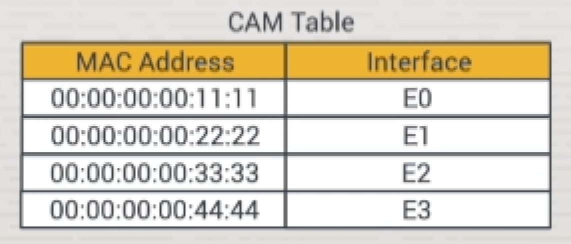


CAM Table

MAC Address	Interface
00:00:00:00:11:11	E0
00:00:00:00:33:33	E2

The switch's CAM table is complete once every device on the network has the chance to send a frame.

Once it is complete, the switch can forward data where it needs to go.



CAM Table

MAC Address	Interface
00:00:00:00:11:11	E0
00:00:00:00:22:22	E1
00:00:00:00:33:33	E2
00:00:00:00:44:44	E3

The next time A tries to send data toward any of these other three devices--for example, if I'm trying to target Station B--the destination MAC address will be 2222. The switch looks at the CAM table, sees that 2222 can be reached over port E1, and intelligently forwards the frame to that single port.

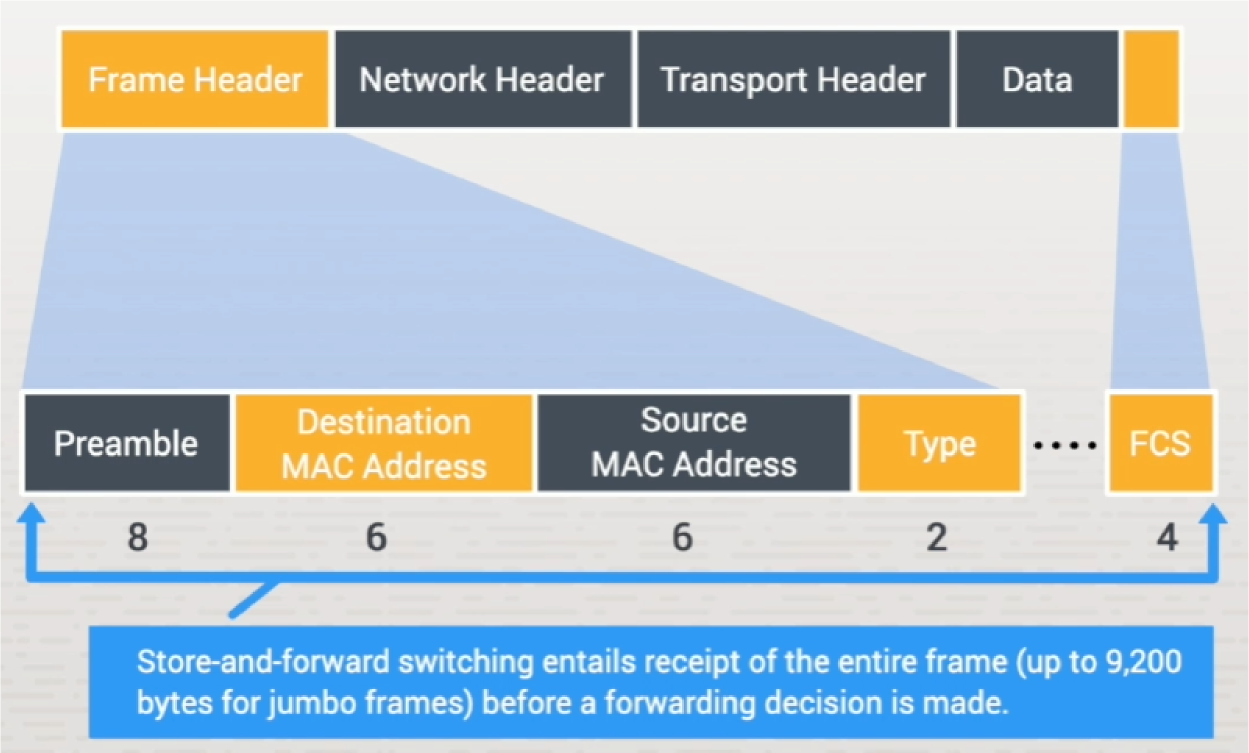
Application-specific Integrated Circuits (ASICs)

Switches use application-specific-integrated circuits, or ASICs, to make fast decisions. They make these decisions one of two ways: store-and-forward switching or cut-through switching.

Store-and-Forward Switching

Store-and-forward switching is Cisco's preferred method because the switch won't forward the frame until the entire frame is received and validated.

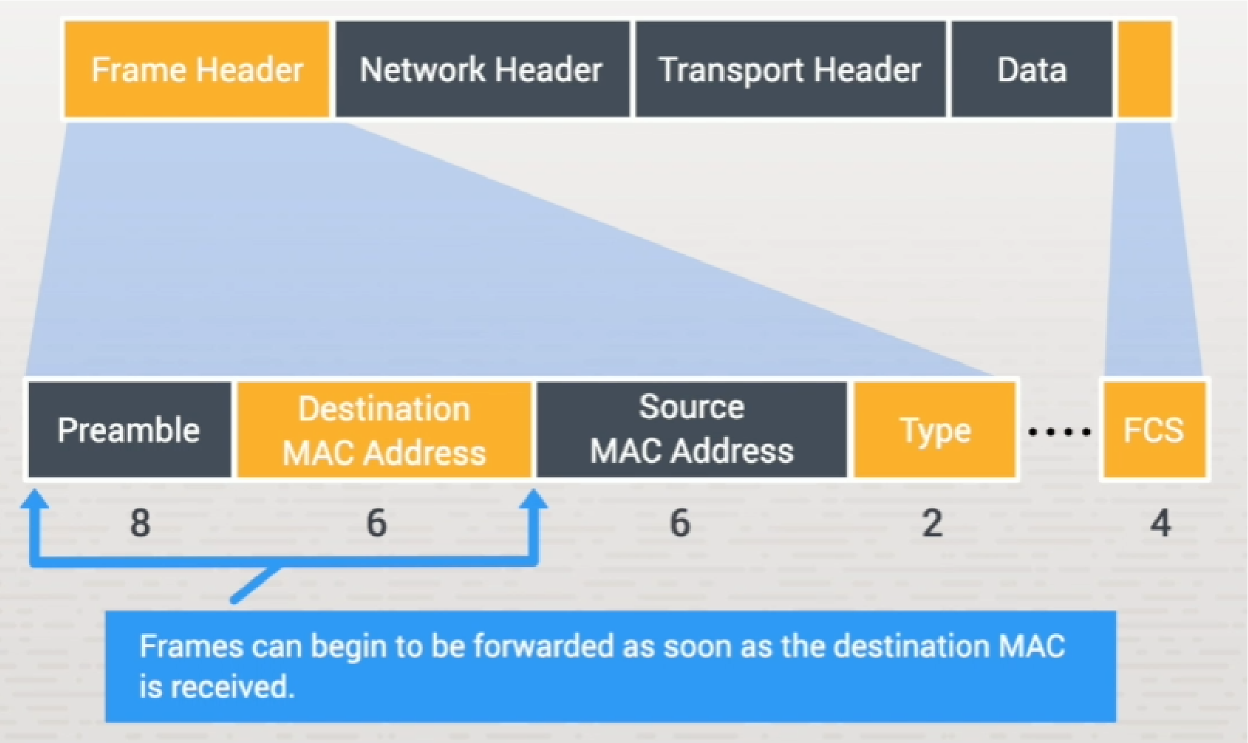
The switch checks the frame check sequence, the FCS, for errors and discards bad frames. The ingress interface buffers the frames as the FCS is checked. This gives the switch the opportunity to adjust speed differences between the ingress and egress ports.



Store-and-forward switching entails receipt of the entire frame (up to 9,200 bytes for jumbo frames) before a forwarding decision is made.

Cut-Through Switching

Cut-through switching forwards the frames as soon as it knows the destination MAC address and the egress port. This method ensures that the frame is at least 64 bytes and removes fragmented frames. Cut-through switching is good for switches that need extremely fast latency, under 10 microseconds. But it does not check the FCS-causing errors. If they build up, it'll cause bandwidth problems. In addition, this method does not support ports with varying ingress and egress speeds.



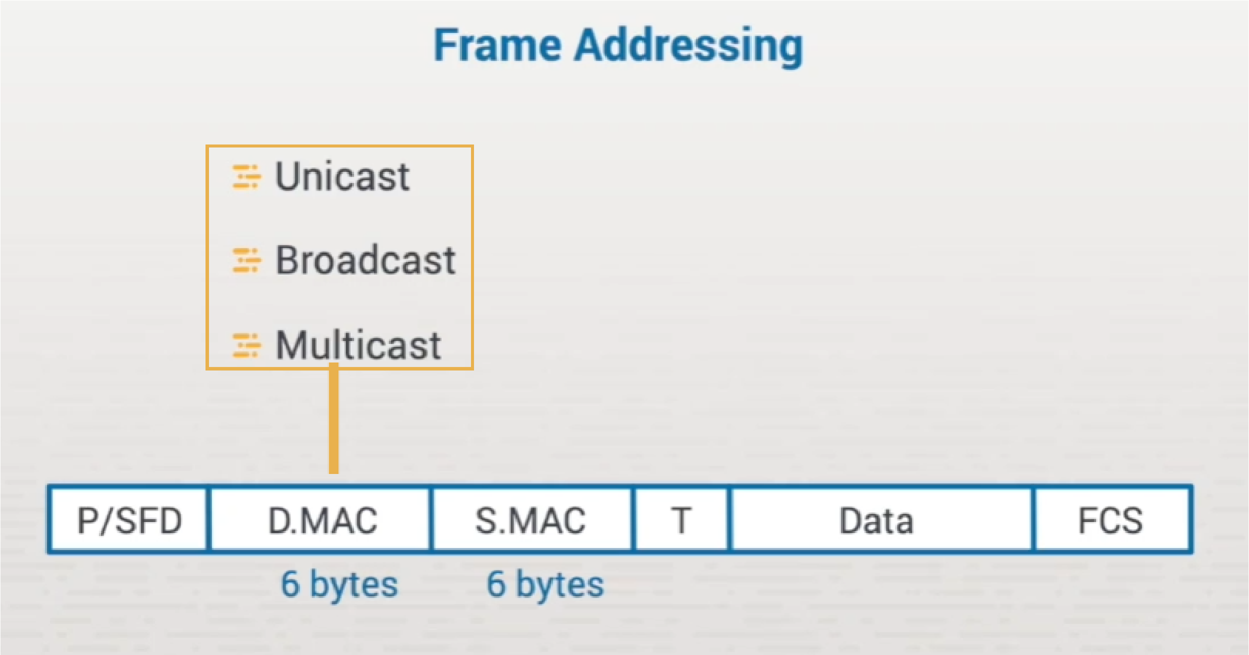
Default MAC Address Use

It's important to note that, by default, all switch ports allow any MAC address to be used. The switch doesn't care what MAC address is learned on a given port. If a device is disconnected and a new device is connected, the switch will purge the old entry, add the new entry, and start forwarding frames. You probably don't want just anybody plugging into these switch ports, so you'll want to lock down your switch so that you can control which MAC addresses are connected to each switch port.

Unicast, Broadcast, and Multicast Frames

Frame Addressing

In a frame, we have the data or payload of the frame, which is simply the packet being sent. On the end of the frame, we have something called a frame check sequence. We also have type fields and some headers called preambles and start frame delimiters. But, most importantly, we have a destination MAC field and a source MAC field, which are each six bytes in length. The source MAC field is the physical, or burnt-in, Ethernet address of the device sending this frame, and the destination MAC address is who you're trying to send the frame to.

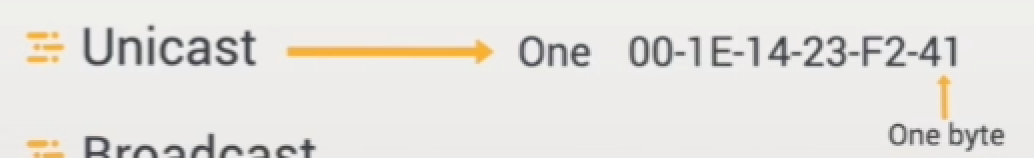


In this video, we're going to focus on the destination MAC address because there are three types of addresses that you can put into that field: unicast, broadcast, and multicast.

Unicast Address

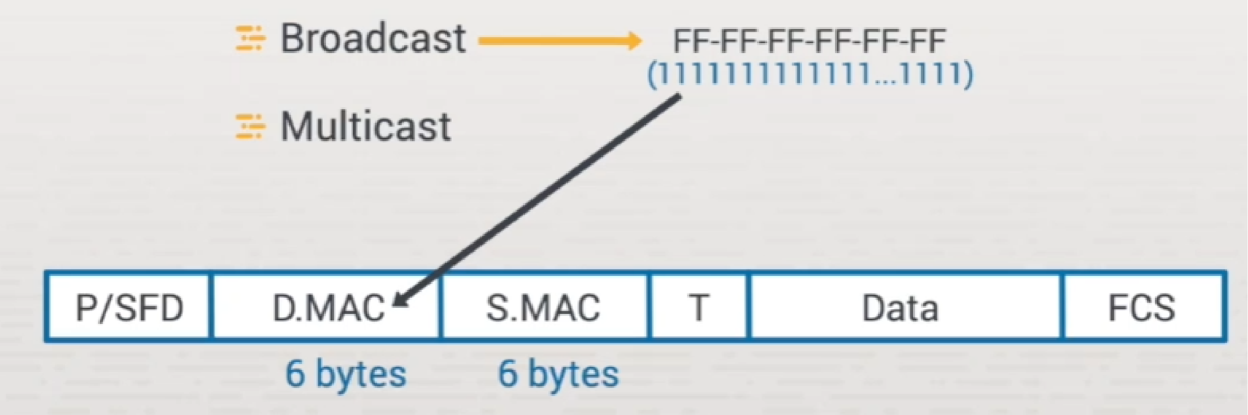
Let's say you want to send data to a single device--a local web server, a router, or local printer. That's called a unicast address.

"Uni" means one. You have one recipient. So, if you're trying to target one system, you need to put its 6-byte MAC address into that field. That MAC address might look something like this, 00-1E-14-23-F2-41. The number might look a little strange because it's expressed in hexadecimal, so you're going to have some letters and numbers. But each of the segments represents one byte, so that's what's put in the destination MAC field.



Broadcast Address

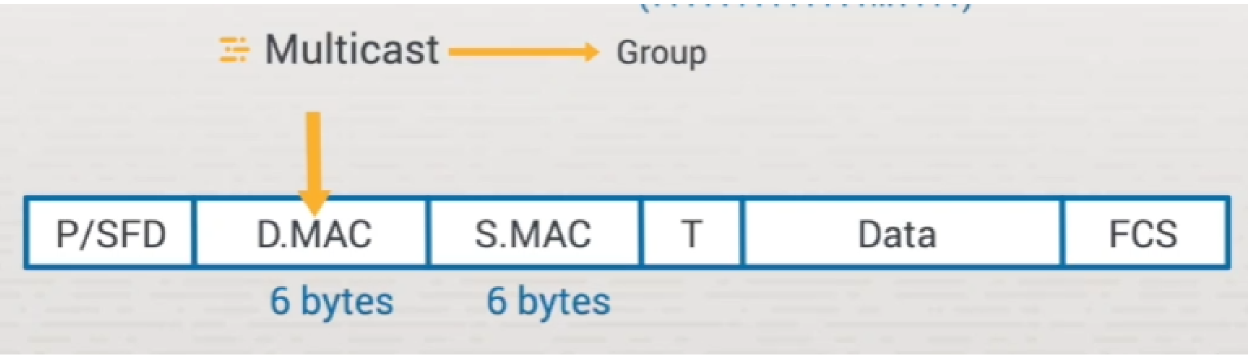
If you want to send data to all recipients--not just one, but all--now, you're looking at a broadcast frame. The broadcast address looks quite a bit different from a unicast address. A broadcast address in hexadecimal is going to be a series of Fs. Now, a series of Fs, if you actually convert it to binary, is going to be consecutive 1s--6 bytes (or 48 bits) of consecutive ones. So nothing but 1s are going to be inserted into this field, and that designates it as a broadcast address.



Multicast Address

A multicast address is intended for a group of recipients--maybe a certain number of endpoints that are wanting to watch a streaming video or a certain number of endpoints that you want to receive a message at the same time. It's for items that are sent to more than one person, but not everyone. The address that's going to go in this field for multicast really depends on the types of multicast groups and IP addressing you're using.

With all three types of frames, we're only talking about the destination MAC address, and it's based on how many devices you're trying to communicate with. And since you're the one sending the data, the source MAC address is the same across all three types.



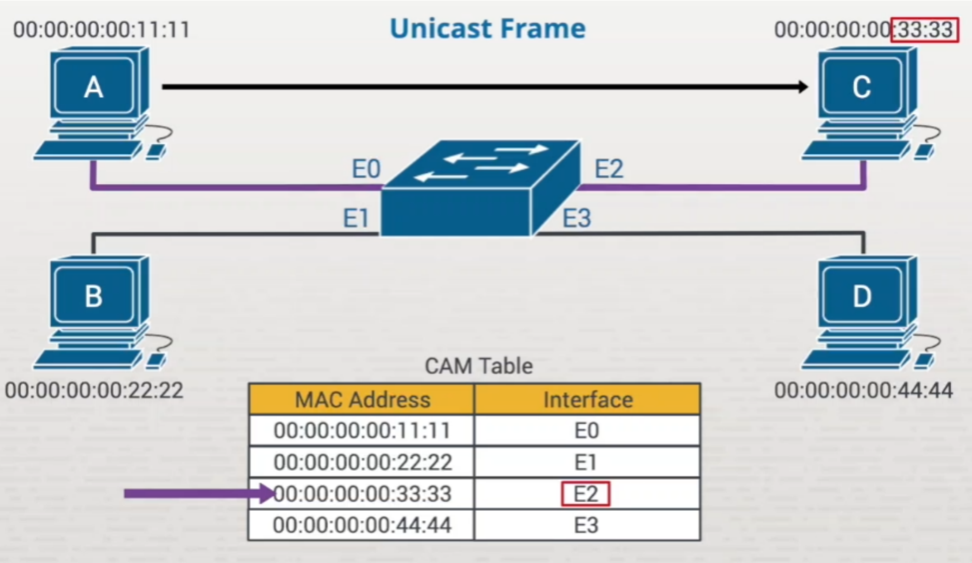
Switches

To demonstrate how a switch reacts to these different types of frames, we'll use this diagram of four connected computers. The switch has built its MAC address or CAM table, so it's had the chance to learn where all four MAC addresses are and the ports required to reach them.

UNICAST:

Computer A is trying to send a frame to computer C. In this case, the destination MAC address is going to be 3333.

1. This is a unicast frame, so the data is sent to the switch.
2. The switch looks at the destination MAC field. It sees 3333, finds it in the CAM table, and see that it needs to send that data out port E2 and forwards the unicast frame out to port E2, leading to computer C.



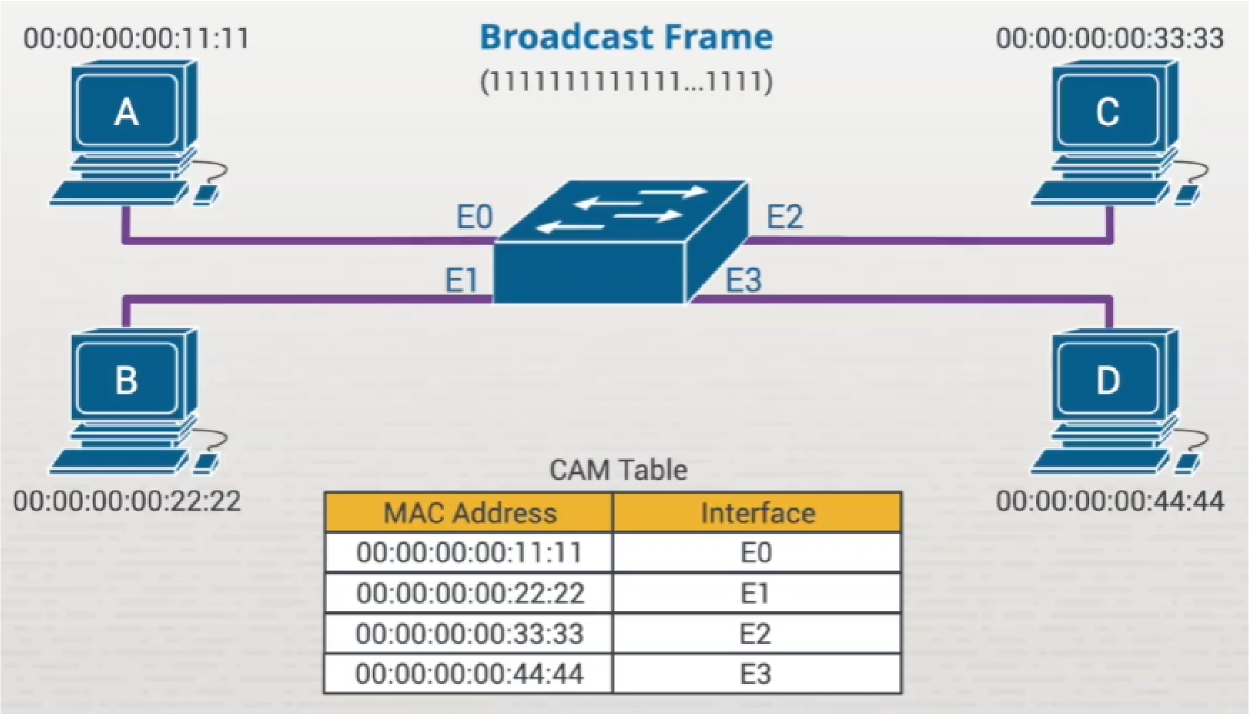
MAC Address	Interface
00:00:00:00:11:11	E0
00:00:00:00:22:22	E1
00:00:00:00:33:33	E2
00:00:00:00:44:44	E3

BROADCAST:

In the case of a broadcast frame, let's say that computer A is now sending out a broadcast frame for whatever reason. Maybe it's trying to ARP. ARP requests are broadcasts. Maybe it's trying to locate the MAC address of another node on the network. Maybe it's sending out a DHCP request to obtain an IP address configuration, or maybe it's a server trying to discover devices on the network using broadcast frames.

Computer A sends a broadcast-addressed frame into the switch. Remember that a broadcast address is simply a series of 48 consecutive 1s, or six bytes, or 48 bits.

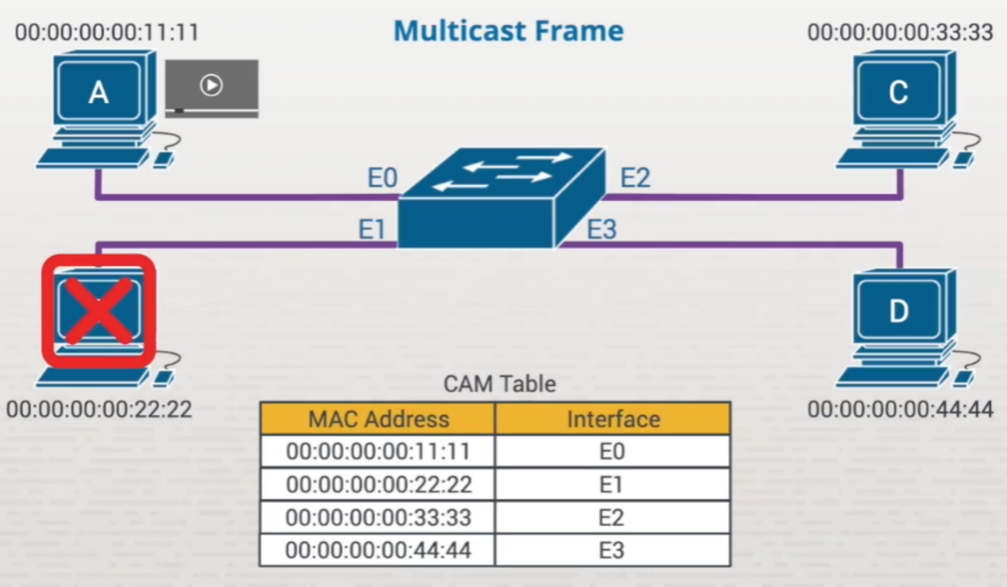
1. When a switch receives the frame with this string of 1s in the destination MAC field, it assumes that all ports on the switch should get a copy of this frame, so it takes a copy of it and sends it out E1, E2, and E3. Broadcast frames are propagated to all ports on the switch.



MAC Address	Interface
00:00:00:00:11:11	E0
00:00:00:00:22:22	E1
00:00:00:00:33:33	E2
00:00:00:00:44:44	E3

MULTICAST

A multicast frame is handled the same way as a broadcast frame. Let's say that station A is a streaming video server, and computers C and D are currently registered to the multicast group that video is playing toward. The switch receiving a multicast frame will still copy it everywhere, just like a broadcast. But in this case, the multicast frame will only be received by C and D and processed by their video player, while computer B will realize it's not part of that multicast group and drop the frame.



CAM Table

MAC Address	Interface
00:00:00:00:11:11	E0
00:00:00:00:22:22	E1
00:00:00:00:33:33	E2
00:00:00:00:44:44	E3

All right, let's review what we've learned before we end. The destination MAC address field of an Ethernet frame can have a unicast, broadcast, or multicast address. A unicast address is intended for a single target, while a broadcast is intended for all recipients--everyone. Multicast is intended for a sub-set or group of hosts. Switches can intelligently forward unicast traffic, assuming that their MAC address table (their CAM table) has learned where that target is, but switches will always flood broadcast and multicast traffic to all ports on that switch.

Switch Operations Facts

Switches learn their environment, where devices are, and can intelligently forward frames through the network to the intended target.

Content Addressable Memory (CAM) Table

Both bridges and switches build a forwarding database. The database is a list of MAC (Data Link) addresses and the port used to reach the device. Bridges and switches can automatically learn about devices to build the forwarding database. A network administrator can also program the device database manually.

Switches build a forwarding database in a manner similar to bridges. The steps for this process are:

1. When a frame arrives on a switch port, also called an interface, the switch examines the source and destination address in the Data Link header and uses the information to complete the next two tasks.
2. The switch examines the source MAC address of the frame and notes which switch port the frame arrived on. If the source MAC address is:

  - Not in the switch's Content Addressable Memory (CAM) table, a new entry is added to the table that maps the source device's MAC address to the port on which the frame was received. This is called *MAC learning* and allows the switch to build (over time) a map of the devices that are connected to specific switch ports.
  - Is already mapped to the port on which the frame was received, no changes are made to the switch's CAM table.
  - Is already in the switch's CAM table, but the frame was received on a different switch port, the switch updates the record in the CAM table with the new port.
    MAC addresses are kept in the CAM table for only a finite period of time. If no messages have been received from a particular MAC address for a period of time, the MAC address is removed from the CAM table. This process is known as *MAC aging*.
3. The switch examines the destination MAC address of the frame. If the destination MAC address of the frame is a:
    - Broadcast address, then the switch sends a copy of the frame to all connected devices on all ports. This is called *flooding* the frame.
    - Unicast address, but no mapping exists in the CAM table for the destination address, the switch floods the frame to all ports. The connected device that the frame is addressed to will accept and process the frame. All other devices will drop the frame.
    - Unicast address and mapping exists in the CAM table for the destination address, the switch sends the frame to the switch port specified in the CAM table. This is called *forwarding*.
    - Unicast address and mapping exists in the CAM table for the destination address, but the destination device is connected to the same port from which the frame was received, the switch ignores the frame and does not forward it. This is called *filtering*.

CAM tables are also referred to as MAC address tables. In most situations, the two terms can be used interchangeably.

Transparent bridges forward packets only if the following conditions are met:

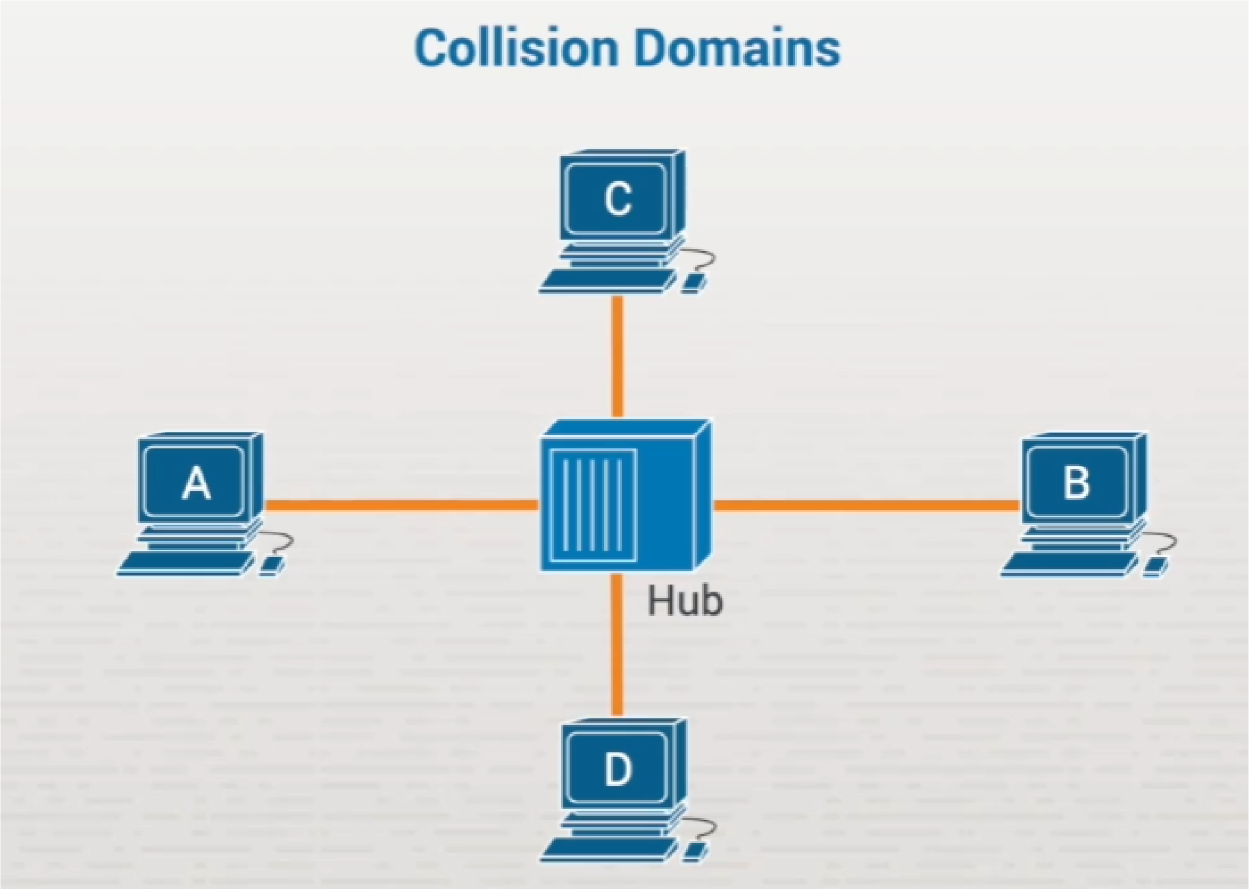
- The frame contains data from the layers above the Data Link layer.
- The frame's integrity has been verified through a valid Cyclic Redundancy Check (CRC).
- The frame is not addressed to the bridge.

Frame Types

Three types of frames can be created by network hosts and transmitted by network switches:

Frame Type	Characteristics
Unicast	A unicast frame is sent to a single recipient. The MAC address of the recipient's network interface is inserted in the Destination MAC Address field of the frame. When a switch receives a unicast frame, the switch checks its CAM table to determine which port the destination host is connected to and then forwards the frame to that port. The most significant bit of the most significant byte in the destination MAC determines whether the frame is a unicast frame or some other type of frame. The most significant byte is the leftmost byte in the address, and the most significant bit is the rightmost bit of that byte. If the most significant bit is set to **0**, the address is a unicast address. Switches can quickly check the most significant bit and determine if a frame is a unicast frame or not. For example, the most significant byte in the 00-50-56-C0-00-08 MAC address is 0000000**0** in binary. The most significant bit in this byte is **0**, so a frame addressed to this address would be a unicast frame.
Broadcast	A broadcast frame is sent to all interfaces on the same physical network segment. The destination MAC address of the frame is set to FF-FF-FF-FF-FF-FF (which is the binary equivalent of 48 consecutive ones). When a switch receives a broadcast frame, it floods the frame to all ports. Broadcast frames are commonly used by the ARP and DHCP protocols.
Multicast	A multicast frame is sent to multiple recipients. The destination MAC address of the frame is set to the unique multicast MAC address of the application, protocol, or data stream that created the frame. When a switch receives a multicast frame, it floods the frame to all ports. However, only network hosts that are members of the multicast group actually process the frame. All other hosts ignore it. If the most significant bit of the most significant byte in the destination MAC address in the frame is set to **1**, then the frame is a multicast frame.

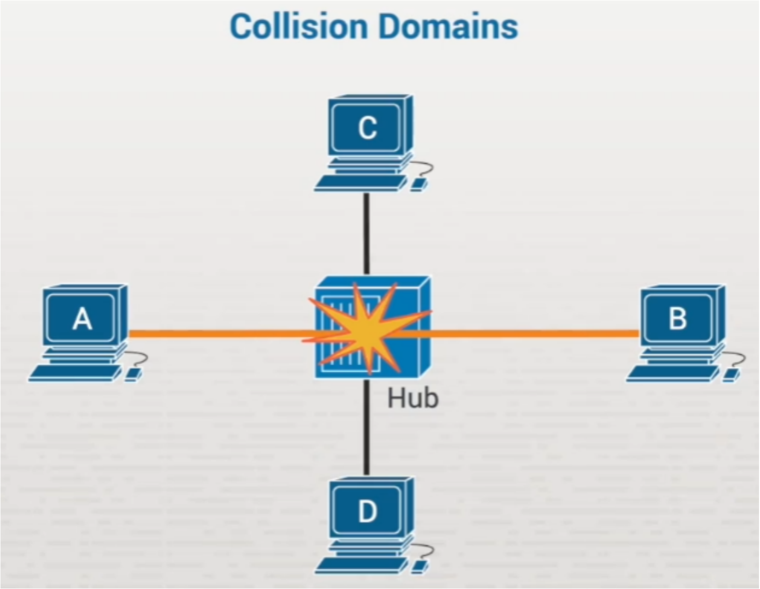
Collision and Broadcast Domains



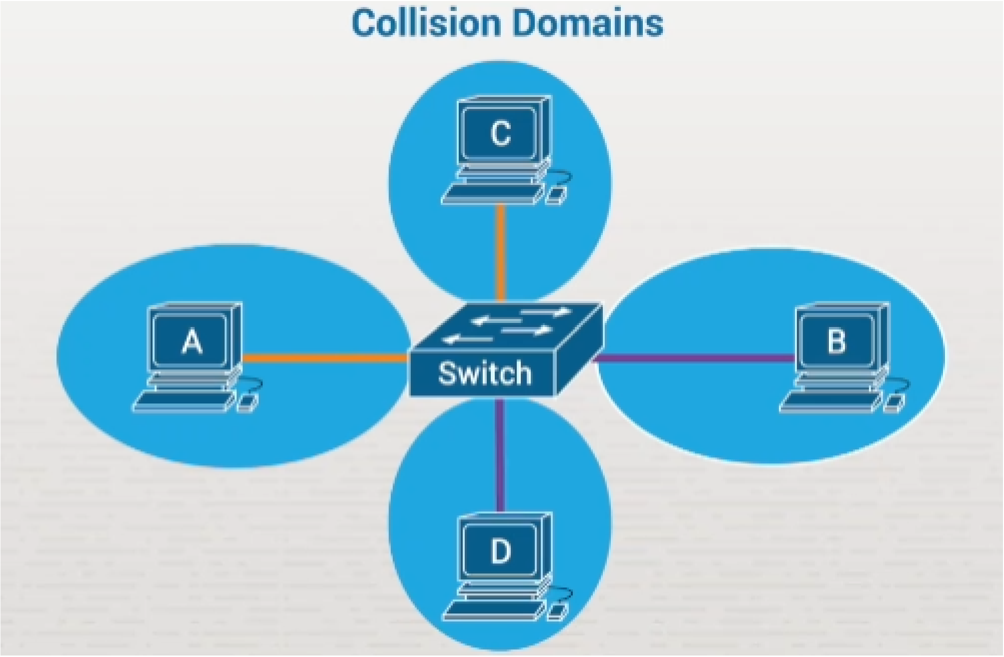
Collision Domains

We have a hub that's a multi-port repeater. Any signaling that's received by the hub is replicated to all active ports on the hub. For instance, if this workstation on the left sends a frame out to the hub, the hub doesn't really see an Ethernet frame. The hub only sees the sequence of bits coming in, which are then replicated to the other ports. All the active hosts on the hub see all the transmissions. Whether it's a unicast frame or a broadcast frame doesn't matter because the hub only sees bits. And all bits are replicated to all other ports.

***Essentially, it's like all four endpoints in the picture are connected to the same wired segment.**



All ports on a hub are in one collision domain because a hub can only process one piece of data at a time. A collision domain is defined by a region in your network where two systems within it try to communicate simultaneously. When they do, their signals collide, and they're destroyed. So only one workstation can send data through a hub at a given moment.

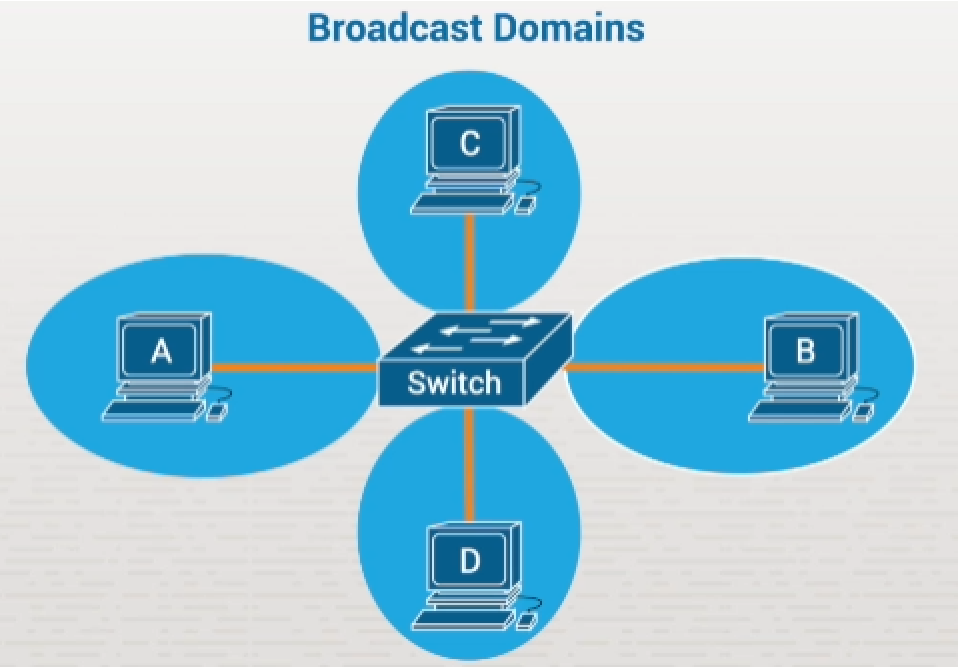


Let's consider what happens when we change that hub to a switch. In a switched environment, every port connected to the switch is its own separate collision domain, meaning that the switch is capable of supporting multiple concurrent conversations. The station on the left, here, could be talking to the station on top, and these two could also be talking at the same moment, because there's no contention between the involved ports. Switches are more intelligent than hubs, and every switched port has its own processor. So, in this case, every port being its own collision domain means the switch can support multiple concurrent conversations without fear of collision.

Remember that a hub is considered a Layer 1 device in the OSI model. It sees incoming voltage and replicates it.

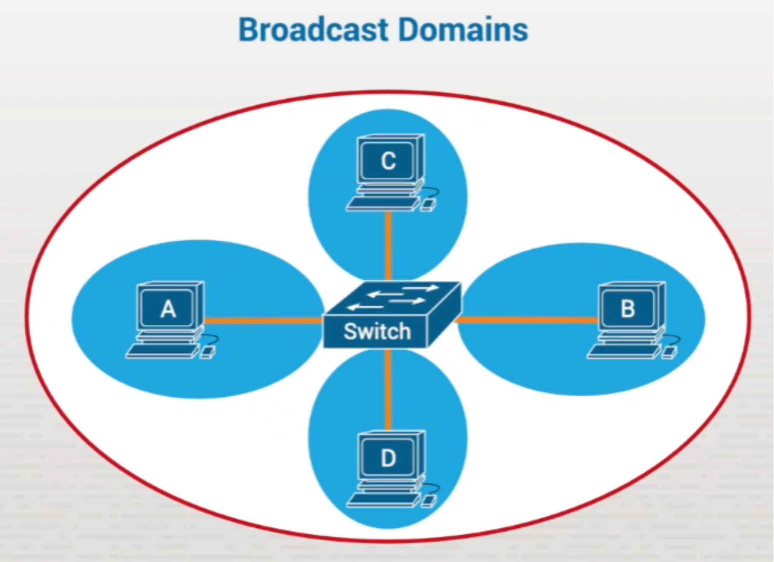
A switch is a Layer 2 device in the OSI model, so it has the ability to read frames and discern where they're supposed to go. Because it has the ability to see destination MAC addresses, it can take data and intelligently forward it out one destination port. This is how it avoids sending data out other ports, where it's not needed.

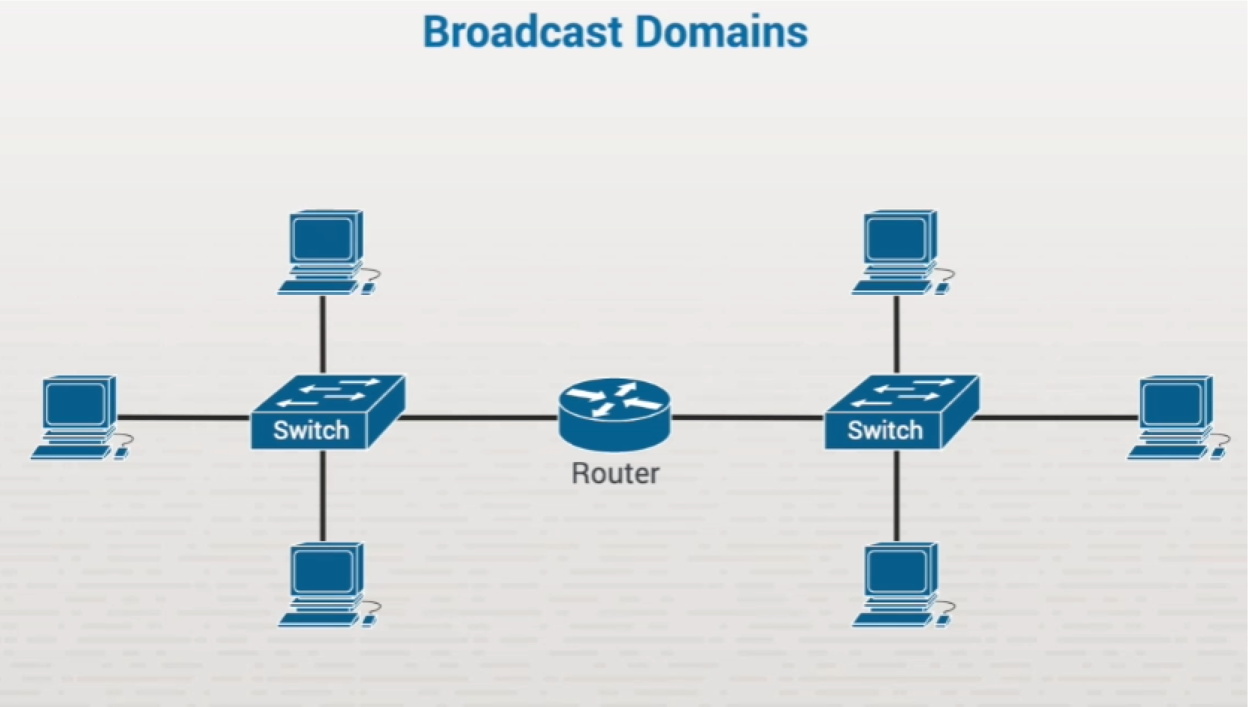
Broadcast Domains



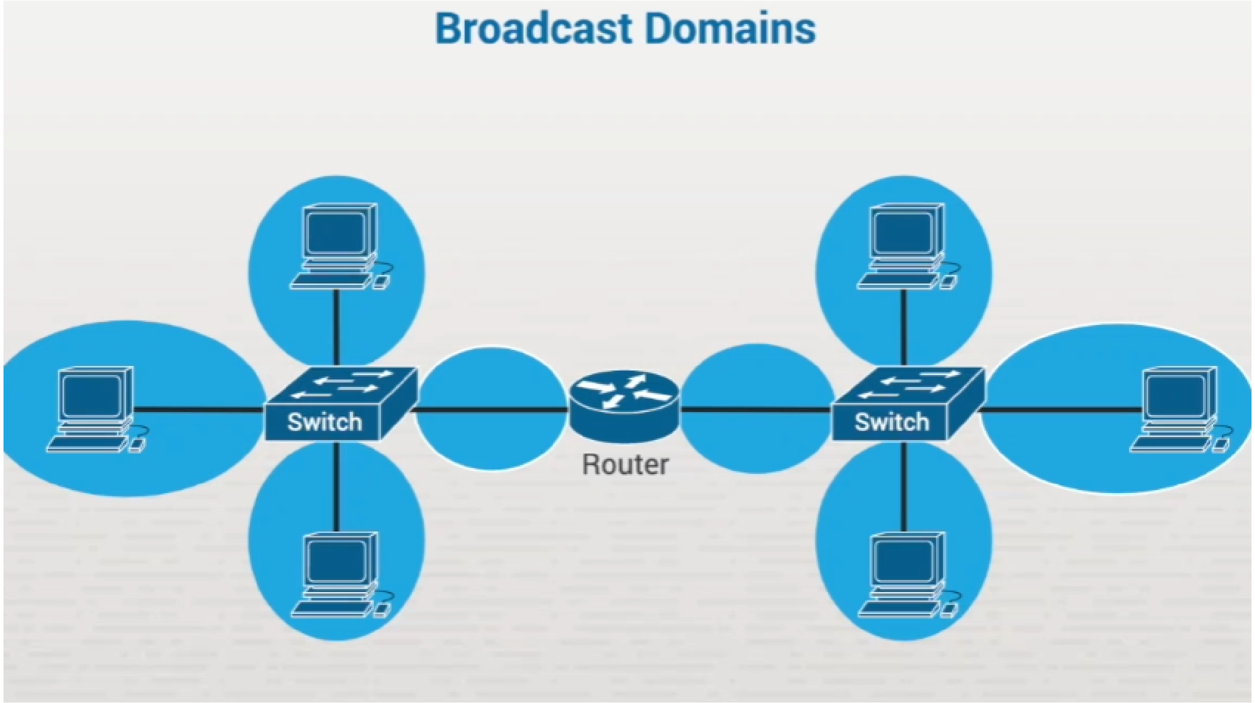
Now let's look at a broadcast domain. In this picture, when any of these hosts sends out a broadcast or multicast frame, the switch forwards the data across all the connected ports.

Unicast frames are intelligently forwarded, while broadcast frames are still treated like a hub and sent to all the connected ports. So, even though every port on the switch defines a separate collision domain, all the ports on the switch belong to a single broadcast domain.



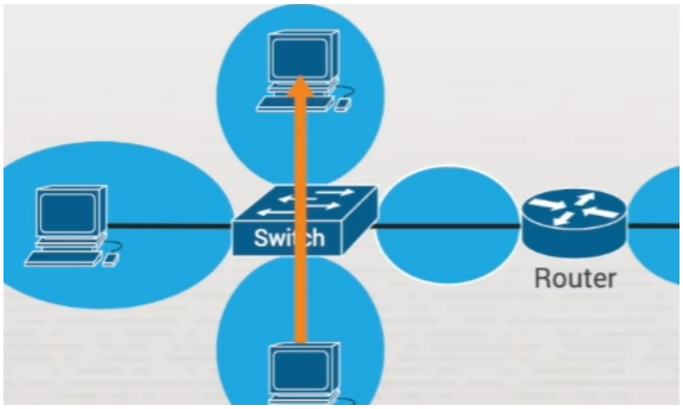


In a routed environment, things work a little bit differently. In this case, we have one router in the middle connected to two switches on either side, and the switches have a few nodes connected to them. We already know that switches create collision domains. We know that this is going to be a collision domain--all links connected to the switch, including the link that connects to the router. Same on the other side. We have one, two, three, four; a total of eight separate collision domains.

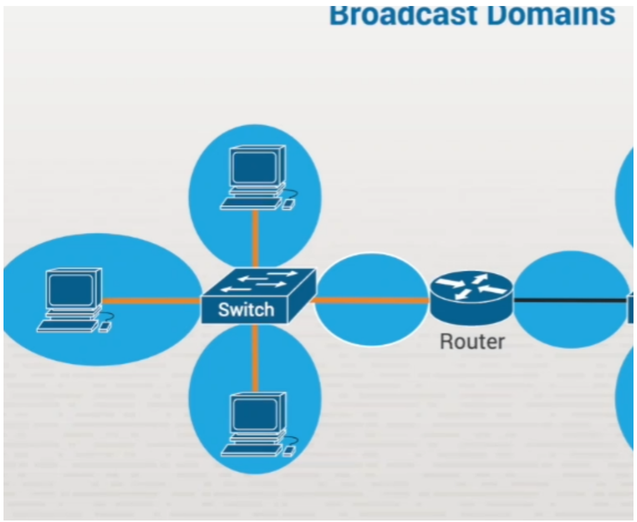


Broadcast Domains

Based on that, we know that if we're sending unicast frames, then any node trying to communicate to a different node through this switch will be sent through the switch intelligently, to only that single destination port.



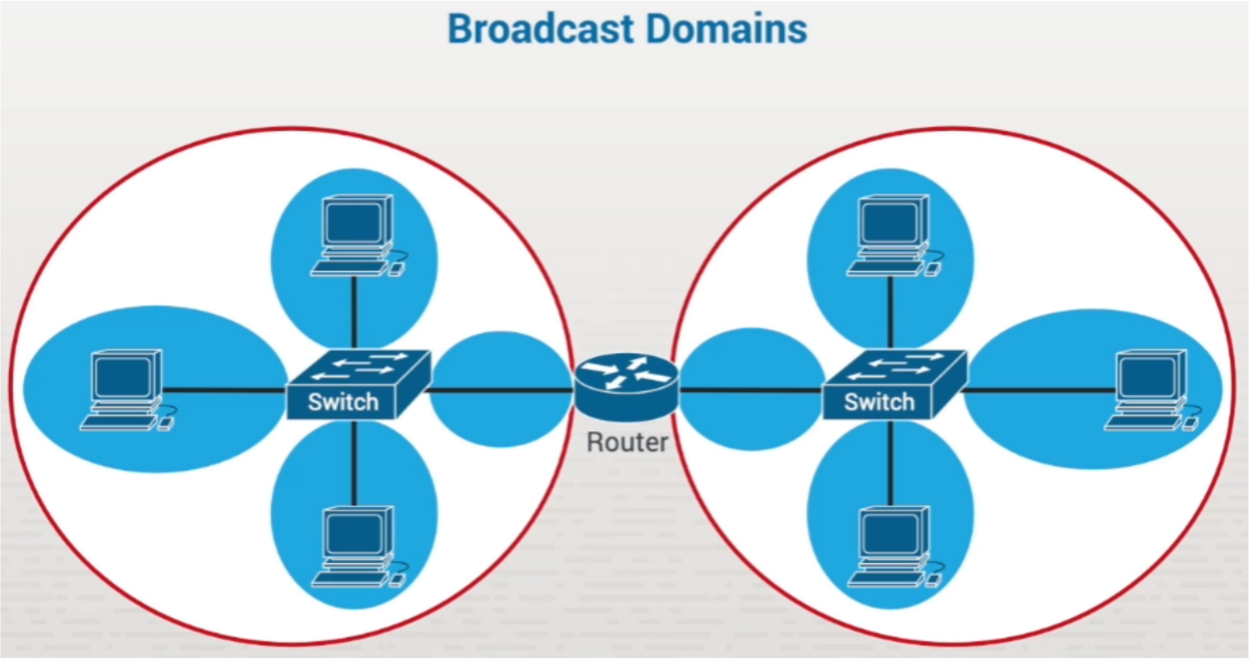
But we also know that a broadcast frame sent into that switch from, let's say, this guy on the left, will get sent to all connected ports, including the one connected to the router.



That's because all these ports belong to a single broadcast domain on the switch. But what happens when it hits the router?

Because a router defines both a collision domain and a broadcast domain, broadcast traffic won't go through the router. Every port on a router is both a collision domain and a broadcast domain.

In this diagram, you have two broadcast domains. And that's a good thing because without these broadcast boundaries, a workstation sending out an ARP broadcast would end up propagating throughout every switch port on your network and be received by every node throughout your organization. Routers provide broadcast boundaries to prevent that, but they also provide collision domain boundaries.



Broadcast and Collision Domain Facts

LAN segmentation is the process of dividing the network into segments to overcome problems such as excessive collisions, broadcast traffic, or heavy network traffic. By segmenting a LAN, you can increase network performance, maximize bandwidth, and reduce congestion.

This lesson covers the following topics:

- Collision and broadcast domains
- Device connectivity issues

Collision and Broadcast Domains

As you segment the network, consider the collision and broadcast domains on the network:

- A *collision domain* is any network or subnetwork where devices share the same transmission medium and where packets can collide. Collisions naturally increase as the number of devices in a collision domain increase.
- A *broadcast domain* is any network or subnetwork where computers can receive frame-level broadcasts from their neighbors. As you add devices to a network segment, the amount of broadcast traffic on a segment also increases.

A special condition called a *broadcast storm* happens when broadcast traffic is sent, regenerated, and responded to. In this condition, the amount of broadcast traffic consumes network bandwidth and prevents normal communications. Faulty devices or improper configuration conditions can lead to a broadcast storm.

Segmentation may increase the number of both collision and broadcast domains. The following table shows how membership within collision or broadcast domains differs depending on the connection device used.

Device	Collision Domain	Broadcast Domain
Hub	All devices connected to the hub are in the same collision domain.	All devices are in the same broadcast domain.
Bridge or switch	All devices connected to a single port are in the same collision domain. Each port is its own collision domain.	All devices connected to the bridge or the switch are in the same broadcast domain.

Router	All devices connected to a single interface are in the same collision domain.	All devices accessible through an interface (network) are in the same broadcast domain. Each interface represents its own broadcast domain if the router is configured not to forward broadcast packets.

Device Connectivity Issues

In considering a network expansion solution, it is important to identify the connectivity problems you need to resolve, and then identify the device that is best suited for that situation. The main differences between routers, switches, and bridges are the range of services each performs and the OSI layer they operate at.

Device	Characteristics
Router	Routers perform the following functions that are not performed by bridges or switches: ● Route packets between separate networks. ● Modify packet size through fragmentation and combination. ● Route packets based on service address. Choose a router if you need to: ● Connect your network to a WAN, such as the internet. ● Filter broadcast traffic to prevent broadcast storms. ● Connect two separate networks that use the same protocol. ● Improve performance in the event of a topology change. Routers recover faster than bridges or switches. ● Reduce the number of devices within a domain, effectively increasing the number of broadcast domains. ● Enforce network security. ● Dynamically select the best route through an internetwork. ● Connect two networks of different architectures (e.g., Ethernet to Token Ring).

Switch	Choose a switch if you need to: ● Provide guaranteed bandwidth between devices. ● Reduce collisions by decreasing the number of devices in a collision domain, effectively creating multiple collision domains. ● Implement full-duplex communication. ● Connect two network segments or devices using the same protocol. ● Provide improved performance over a current bridged network. ● Switch traffic without the cost or administration involved with routers.
Bridge	Choose a bridge if you need to: ● Isolate data traffic to one network segment. ● Route traffic from one segment to another with the same network ID. ● Link unlike physical media, e.g. twisted pair and coaxial Ethernet, of the same architecture type. ● Link segments that use the same protocol. ● Create segments without the expense and administration of routers. In most cases, a switch should be used in place of a bridge.

In general, follow these guidelines to make decisions about the appropriate connectivity device:

- Use a bridge to segment the network (divide network traffic) and to provide fault tolerance.
- Use a switch to reduce collisions and offer guaranteed bandwidth between devices.
- Use a router to filter broadcast messages, implement security, or connect different networks.

LAN segmentation and design may be affected by the types of applications and protocols running over the network. For instance, Voice over Internet Protocol (VoIP) requires a well-engineered, end-to-end network that provides little latency for data stream transmission. Fine-tuning the network to adequately support VoIP involves overcoming the following challenges:

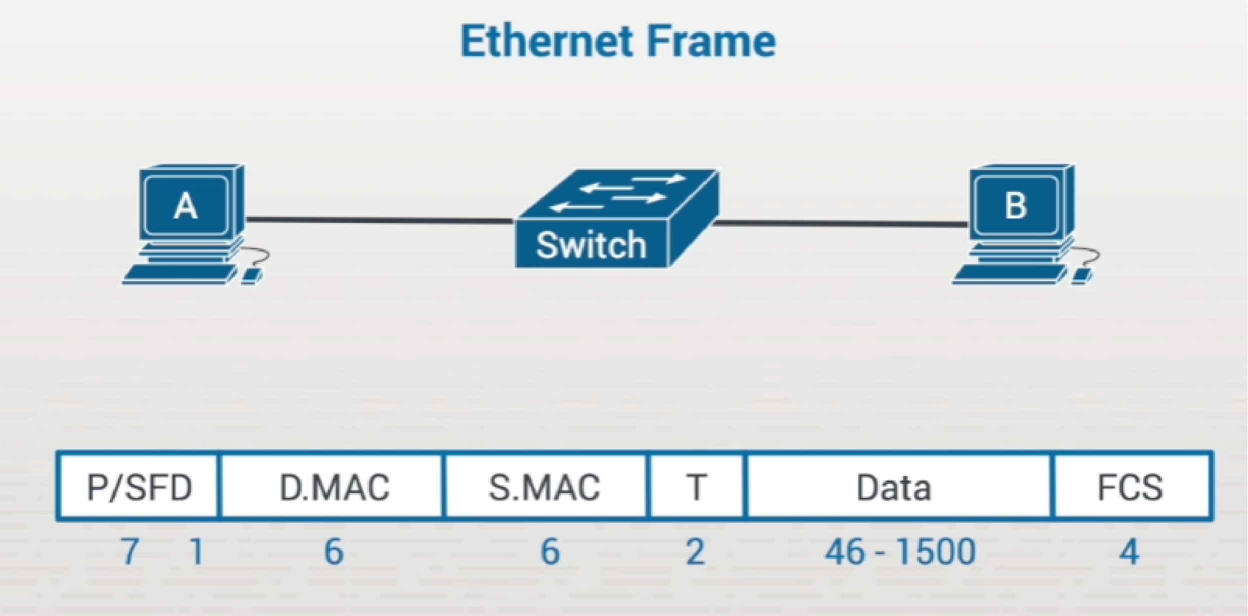
- VoIP requires a very low delay as data is transferred between the sending and receiving phones, e.g., less than 200 milliseconds (.2 seconds).
- During transfer, the *jitter* (variations in delay) must be low as well, e.g., less than 30 milliseconds (.03 seconds).

- When packets do not arrive at the destination it is known as *packet loss*. If a VoIP packet was lost in transit, there is no need to recover the packet. By the time the packet is recovered, it would sound like a break in the VoIP call.
- *Echo* is hearing your own voice in the telephone receiver while you are talking. When timed properly, echo is reassuring to the speaker. If the echo exceeds approximately 25 milliseconds, it can be distracting and cause breaks in the conversation. VoIP implementations use echo cancellers to regulate the echo.
- To secure VoIP data, the network should have a VoIP Virtual Private Network (VPN) solution. A VPN is a network that uses encryption to allow IP traffic to travel securely over the TCP/IP network. Without a VoIP VPN solution, it is relatively easy to eavesdrop on VoIP calls and even change their content.
- In some cases, IP telephones require Power over Ethernet (PoE). PoE is useful for powering IP telephones and other appliances where it would be inconvenient, expensive, or infeasible to supply power separately.

Switching Methods

Three switching methodologies that impact how much data a switch has to actually read off of a frame before it begins transmitting that frame, which directly impacts performance through the switch.

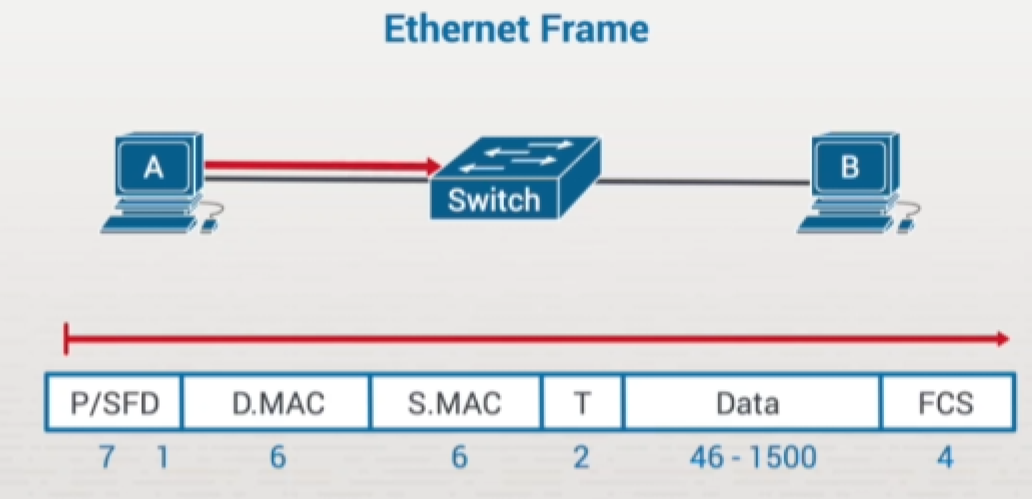
- Cut through
- Fragment free
- Store and forward



Ethernet Frame

P/SFD	D.MAC	S.MAC	T	Data	FCS
7 1	6	6	2	46 - 1500	4

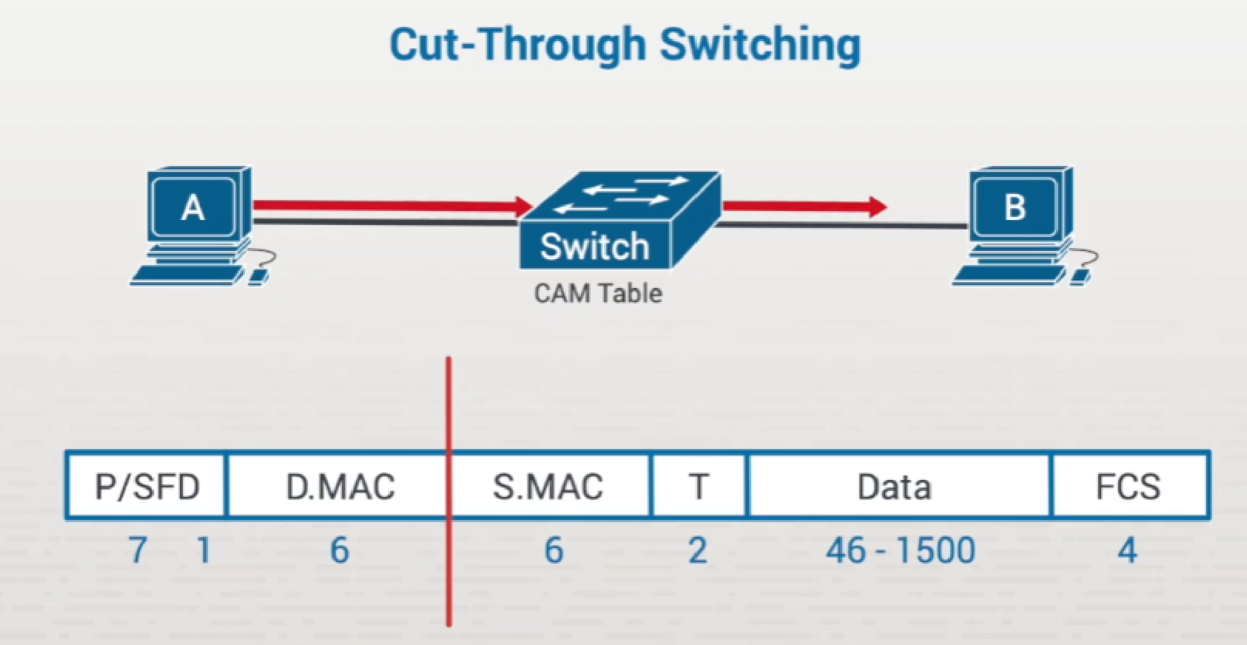
- **P/SFD:** In the front of an Ethernet frame, we've got the Preamble and the Start Frame Delimiter. These are the handshake of a receiving frame.
- **D.MAC** destination MAC (target)
- **S.MAC** Source MAC (sender)
- **T:** Type Field (2-byte)
- **Data:** variable-size Data field. The Data field can be 46 to 1500 bytes, depending on what the frame's carrying.
- **FCS:** Frame Check Sequence, or a Checksum field, which checks to see if any bad frames were received or if a frame was corrupted during transmission. (4 byte)



In this video, we're going to look at how much data has to be read by the receiving switch from front to back before it can begin forwarding the data over to Workstation B. Keep in mind that this frame can be 1500+ bytes in length, or it can be as small as 64 bytes. When workstation A is transmitting a frame for workstation B, the frame transmits the first field received by the switch, here, the Preamble, followed by the Start Frame Delimiter, followed by the destination MAC address. All the frames are transmitted in order.

Cut-Through Switching

If I begin switching the frame after receiving just the destination MAC address, I read this far into it, and I begin sending the frame, even as the rest of it is still coming in. That's called cut-through switching. That's the fastest switch methodology.

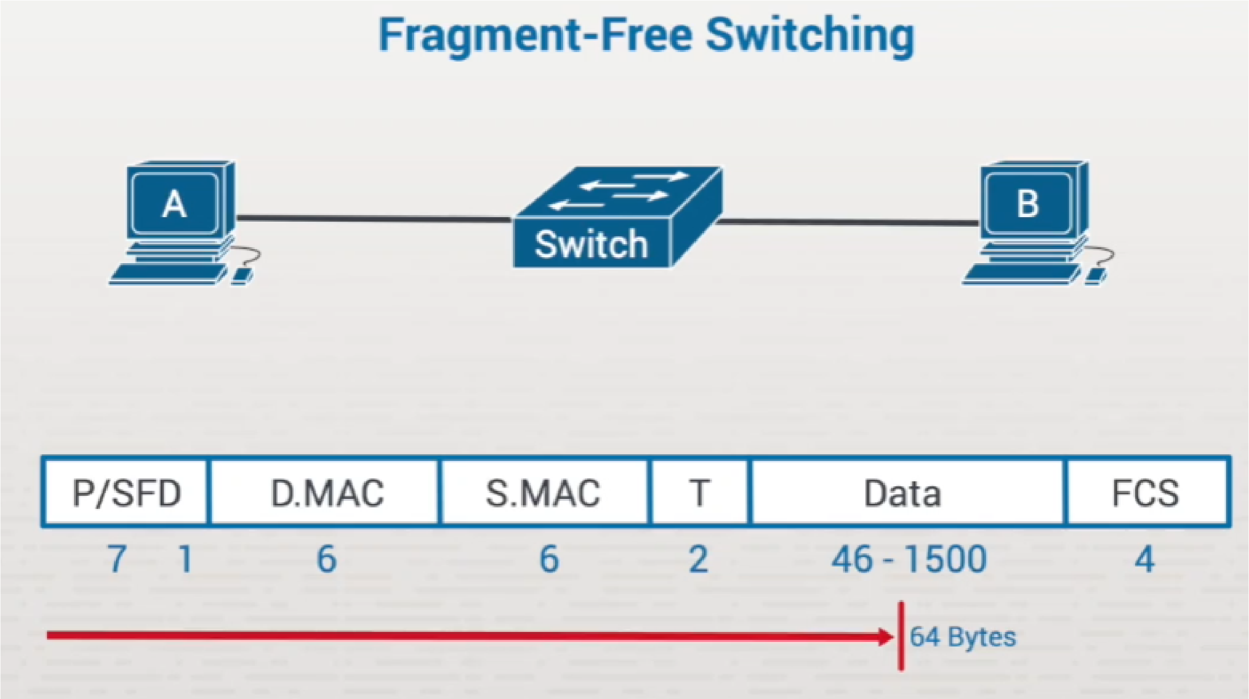


Issues: corrupted data still sent to switch. The primary risk is forwarding corrupt frames.

Fragment-Free Switching

Since the smallest possible frame size for an Ethernet frame is 64, fragment-free mode will read 64 bytes into the frame. After it reads 64 bytes into the data, fragment-free mode knows that the frame is at least a properly sized frame. It's not smaller than 64 bytes, which would make it a runt, which is the result of a collision occurring. If I can read 64 bytes into the data, then I know it's at least a properly sized Ethernet frame, and I'll start forwarding the data over to workstation B. So, it goes one step beyond cut-through.

Fragment-free will at least verify that the frame is a minimally-sized Ethernet frame before it starts that process.**Fragment-free is the second-fastest switching mode.**

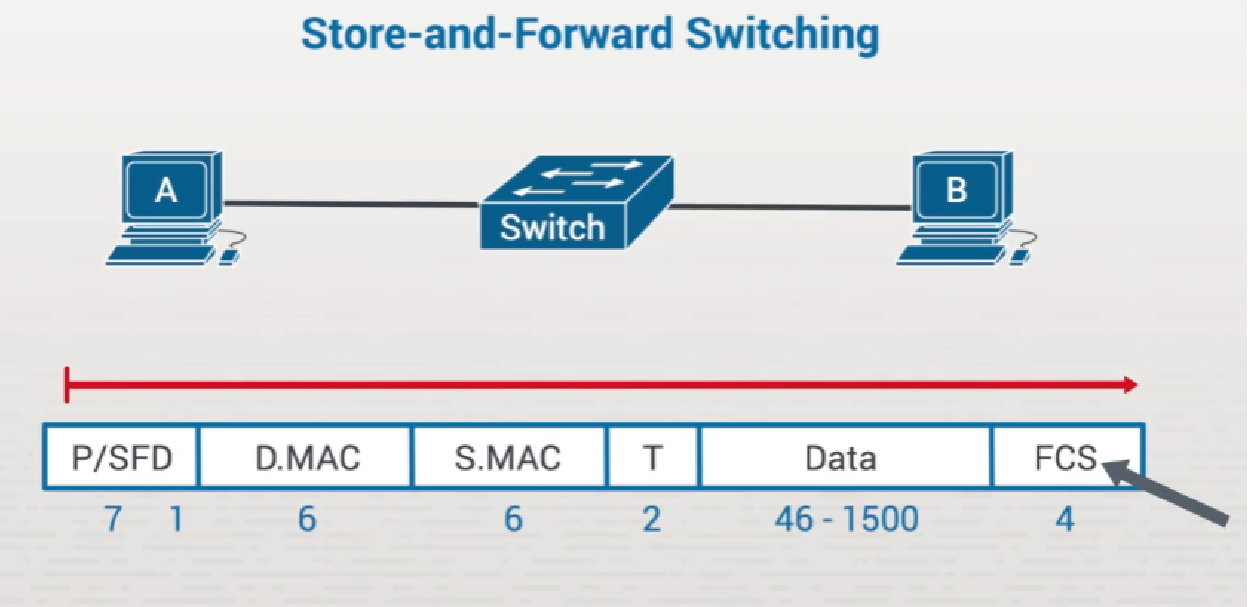


P/SFD	D.MAC	S.MAC	T	Data	FCS
7 1	6	6	2	46 - 1500	4

Issue: Because it hasn't gotten to the Frame Check Sequence yet, you still run the risk of passing a corrupt frame.

Store-and-Forward Switching

With store-and-forward, there's no risk of sending a corrupt frame because the switch doesn't begin forwarding the data until it's read the entire frame, including the frame check sequence. It'll compute its own frame check sequence and verify that they match. That way, it knows for sure that what it's received is a well-formed, properly-sized, and uncorrupted frame. After the error check, the switch forwards that frame out to workstation B.



Store-and-Forward Switching

P/SFD	D.MAC	S.MAC	T	Data	FCS
7 1	6	6	2	46 - 1500	4

Issues: store-and-forward is the slowest (or highest latency) delay switching mode.
Benefit: You for sure that the frames are not corrupt.

NOTE

In reality, it also turns out that switching hardware, even when running in store-and-forward mode, is so fast that you aren't going to see any latency associated with using it over cut-through or fragment-free switching.

Switching Method Facts

There are three internal processing methods that a switch can use when it is forwarding frames. The methods used by a particular switch vary depending on the make and model of the switch. These methods can be categorized as follows, based on how much data the switch reads from the frame before it begins to forward it.

Method	Characteristics
Cut-Through	With the cut-through method, the switch: • Reads the frame until it gets to the destination MAC address and copies it into its buffer. • Begins forwarding the packet without verifying frame integrity. Cut-through has the least latency, or delay, of the three methods, but it does forward packets that have been corrupted.
Fragment-Free	With the fragment-free method, the switch: • Reads the first 64 bytes of a frame. • Verifies that the packet is not fragmented by a collision. • Forwards non-fragmented frames. Fragment-free has more latency than cut-through but less than store-and-forward. This method still forwards some corrupted packets.
Store-and-Forward	With the store-and-forward method, the switch: • Reads the entire frame. • Verifies the frame's integrity with the frame check sequence (FCS). • Forwards the frame to the destination device. Store-and-forward has more latency than the other methods, but corrupt frames are never forwarded.

The original benefit of cut-through and fragment-free switching was decreased latency, but that has become negligible. With network links and modern switching hardware getting to be so much faster, store-and-forward has become the preferred internal processing method.