Table of Contents

How LAN Switches Work

Document ID: 10607

This document contains Flash animation

Introduction

This document covers the general concept of how LAN switches work and the most common features that are available on a LAN switch. The document also covers the differences between bridging, switching, and routing. This document does not cover any of the Cisco Catalyst LAN switch products or configuration features on Cisco Catalyst switches. For Catalyst switch configuration information and Cisco switch product information, refer to:

- LAN Product Support Pages

Prerequisites

Requirements

There are no specific requirements for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

Switches and Networks

A typical network consists of:

- nodes, or computers
- a medium for connection, either wired or wireless
- special network equipment, such as routers or hubs

In the case of the Internet, these pieces work together to allow your computer to send information to another computer. The other computer can be on the other side of the world!

Switches are a fundamental part of most networks. Switches enable several users to send information over a network. Users can send the information at the same time and do not slow each other down. Just like routers allow different networks to communicate with each other, switches allow different **nodes** of a network to communicate directly with each other. A node is a network connection point, typically a computer. Switches allow the nodes to communicate in a smooth and efficient manner.

Illustration of a Cisco Catalyst switch.

There are many different types of switches and networks. Switches that provide a separate connection for each node in a company internal network have the name LAN switches. Essentially, a LAN switch creates a series of instant networks that contain only the two devices that communicate with each other at that particular moment. This document focuses on Ethernet networks that use LAN switches. The document describes what a LAN switch is and how transparent bridging works. The document also explains VLANs, trunking, and spanning trees.

The Addition of Switches

In the most basic type of network of today, nodes simply connect together with the use of hubs. As a network grows, there are some potential problems with this configuration:

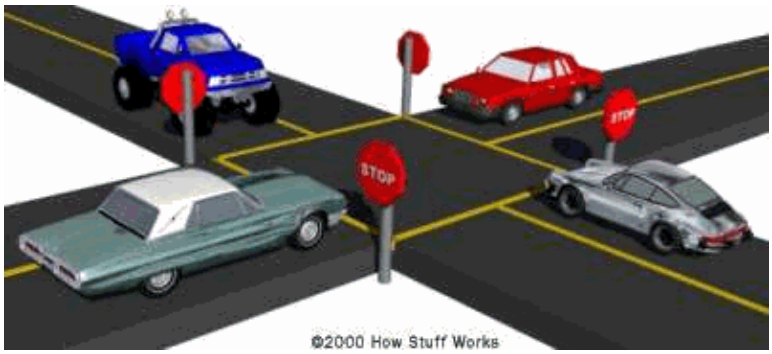
- **Scalability** In a hub network, there is a limit to the amount of bandwidth that users can share. Significant growth is difficult to accommodate without a sacrifice in performance. Applications today need more bandwidth than ever before. Quite often, the entire network must undergo a periodic redesign to accommodate growth.
- **Latency** Latency is the amount of time that a packet takes to get to the destination. Each node in a hub−based network has to wait for an opportunity to transmit in order to avoid **collisions**. The latency can increase significantly as you add more nodes. Or, if a user transmits a large file across the network, all the other nodes must wait for an opportunity to send packets. You have probably experienced this problem before at work. You try to access a server or the Internet, and suddenly everything slows down to a crawl.
- **Network Failure** In a typical network, one device on a hub can cause problems for other devices that attach to the hub. Incorrect speed settings or excessive broadcasts cause the problems. An

example of an incorrect speed setting is 100 Mbps on a 10 Mbps hub. You can configure switches to limit broadcast levels.

- **Collisions** Ethernet uses a process with the name carrier sense multiple access collision detect (CSMA/CD) to communicate across the network. Under CSMA/CD, a node does not send out a packet unless the network is clear of traffic. If two nodes send out packets at the same time, a collision occurs and the packets are lost. Then, both nodes wait for a random amount of time and retransmit the packets. Any part of the network where packets from two or more nodes can interfere with each other is a **collision domain**. A network with a large number of nodes on the same segment often has a lot of collisions and, therefore, a large collision domain.

Hubs provide an easy way to scale up and shorten the distance that the packets must travel to get from one node to another. But hubs do not break up the actual network into discrete segments. Switches handle this job.

Imagine that each vehicle is a packet of data that waits for an opportunity to continue the trip.

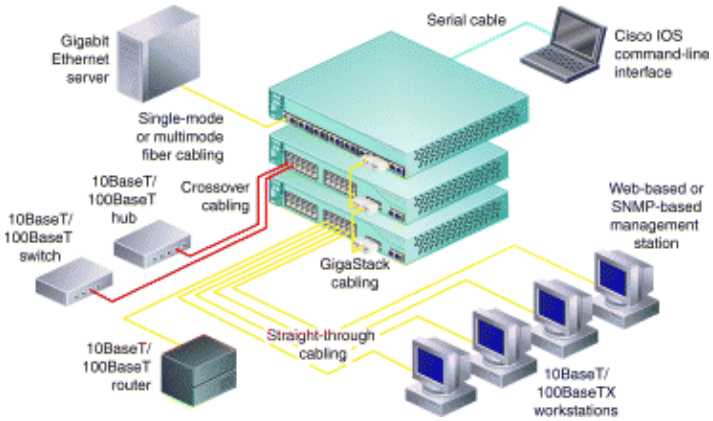


Think of a hub as a four−way intersection where all vehicles have to stop. If more than one car reaches the intersection at one time, the cars must wait for a turn to proceed. But a switch is like a cloverleaf intersection. Each car can take an exit ramp to get to the destination without the need to stop and wait for other traffic to pass. Now imagine this scenario with a dozen or even a hundred roads that intersect at a single point. The wait and the potential for a collision increases significantly if every car has to check all the other roads before the car proceeds. Imagine that you can take an exit ramp from any one of those roads to the road of your choice. This ability is what a switch provides for network traffic.

There is a vital difference between a hub and a switch; all the nodes that connect to a hub share the bandwidth, but a device that connects to a switch port has the full bandwidth alone. For example, consider 10 nodes that communicate with use of a hub on a 10 Mbps network. Each node can only get a portion of the 10 Mbps if other nodes on the hub want to communicate as well. But, with a switch, each node can possibly communicate at the full 10 Mbps. Consider the road analogy. If all the traffic comes to a common intersection, the traffic must share that intersection. But a cloverleaf allows all the traffic to continue at full speed from one road to the next.

In a fully switched network, switches replace all the hubs of an Ethernet network with a dedicated segment for every node. These segments connect to a switch, which supports multiple dedicated segments. Sometimes the number of segments reaches the hundreds. Since the only devices on each segment are the switch and the node, the switch picks up every transmission before the transmission reaches another node. The switch then forwards the frame over the appropriate segment. Since any segment contains only a single node, the frame only reaches the intended recipient. This arrangement allows many conversations to occur simultaneously on a network that uses a switch.

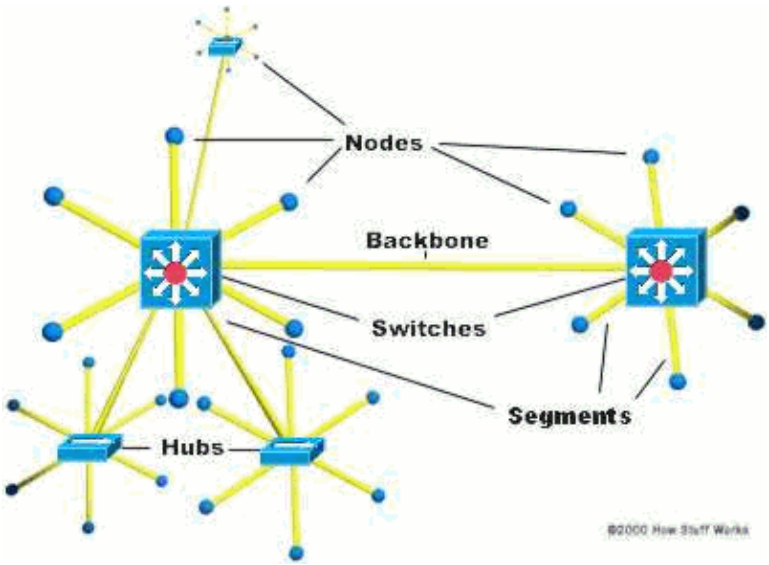
An example of a network that uses a switch.



Switching allows a network to maintain full−duplex Ethernet. Before switching existed, Ethernet was half duplex. Half duplex means that only one device on the network can transmit at any given time. In a fully switched network, nodes only communicate with the switch and never directly with each other. In the road analogy, half duplex is similar to the problem of a single lane, when road construction closes one lane of a two−lane road. Traffic attempts to use the same lane in both directions. Traffic that comes one way must wait until traffic from the other direction stops in order to avoid collision.

Fully switched networks employ either twisted pair or fiber−optic cable setups. Both twisted pair and fiber−optic cable systems use separate conductors to send and receive data. In this type of environment, Ethernet nodes can forgo the collision detection process and transmit at will; these nodes are the only devices with the potential to access the medium. In other words, the network dedicates a separate lane to traffic that flows in each direction. This dedication allows nodes to transmit to the switch at the same time that the switch transmits to the nodes. Thus, the environment is collision−free. Transmission in both directions also can effectively double the apparent speed of the network when two nodes exchange information. For example, if the speed of the network is 10 Mbps, each node can transmit at 10 Mbps at the same time.

A mixed network with two switches and three hubs.

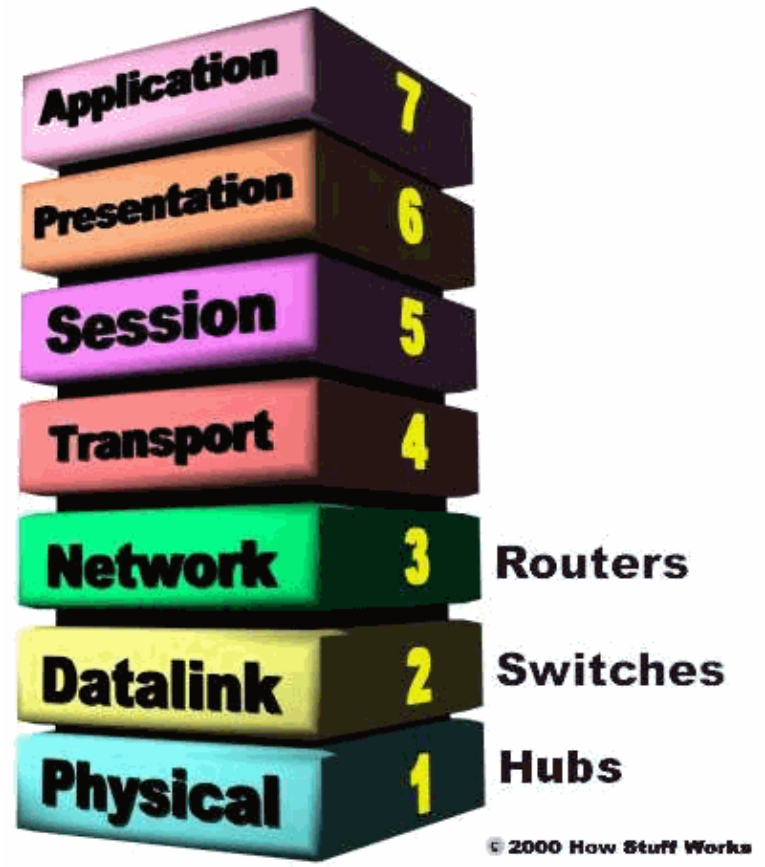


Most networks are not fully switched because replacement of all the hubs with switches is costly. Instead, a combination of switches and hubs create an efficient yet cost−effective network. For example, a company can have hubs that connect the computers in each department and a switch that connects all the department−level hubs together.

Switch Technologies

A switch has the potential to radically change the way that the nodes can communicate with each other. But what makes a switch different than a router? Switches usually work at **Layer 2 (Data or Datalink)** of the Open System Interconnection (OSI) reference model with use of MAC addresses. Routers work at **Layer 3 (Network)** with Layer 3 addresses. The routers use IP, Internetwork Packet Exchange (IPX), or Appletalk, which depends on the Layer 3 protocols that are in use. The algorithm that switches use to decide how to forward packets is different than the algorithms that routers use to forward packets. One difference in the algorithms is how the device handles **broadcasts**. On any network, the concept of a broadcast packet is vital to the operability of the network. Whenever a device needs to send out information but does not know to whom to send the information, the device sends out a broadcast. For example, every time a new computer or other device comes onto the network, the device sends out a broadcast packet to announce the entry. The other nodes, such as a domain server, can add the device to the **browser list**. The browser list is like an address directory. Then, the other nodes can communicate directly with that device. A device can use broadcasts to make an announcement to the rest of the network at any time.

The OSI reference model consists of seven layers that build from the wire (Physical) to the software (Application).



A hub or a switch passes along any broadcast packets that the device receives to all the other segments in the broadcast domain. But a router does not pass along broadcast packets. Think about the four−way intersection again. In the analogy, all the traffic passes through the intersection, despite the direction of travel. Now, imagine that this intersection is at an international border. In order to pass through the intersection, you must provide the border guard with the specific address to which you are going. If you do not have a specific destination, the guard does not let you pass. A router works in a similar way. If a data packet does not have the specific address of another device, the router does not let the data packet pass. This restriction keeps

networks separate from each other, which is good. But, when you want to talk between different parts of the same network, the restriction is not good. Switches can overcome this restriction.

LAN switches rely on **packet switching**. The switch establishes a connection between two segments and keeps the connection just long enough to send the current packet. Incoming packets, which are part of an Ethernet frame, save to a temporary memory area. The temporary memory area is a **buffer**. The switch reads the MAC address that is in the frame header and compares the address to a list of addresses in the switch **lookup table**. In a LAN with an Ethernet basis, an Ethernet frame contains a normal packet as the payload of the frame. The frame has a special header that includes the MAC address information for the source and destination of the packet.

Switches use one of three methods for routing traffic:

- Cut−through
- Store and forward
- Fragment−free

Cut−through switches read the MAC address as soon as a packet is detected by the switch. After storing the six bytes that make up the address information, the switches immediately begin to send the packet to the destination node, even though the rest of the packet is coming into the switch.

A switch that uses **store and forward** saves the entire packet to the buffer and checks the packet for Cyclic Redundancy Check (CRC) errors or other problems. If the packet has an error, the packet is discarded. Otherwise, the switch looks up the MAC address and sends the packet on to the destination node. Many switches combine the two methods by using cut−through until a certain error level is reached, then changing over to store and forward. Very few switches are strictly cut−through because this provides no error correction.

A less common method is **fragment−free**. Fragment−free works like cut−through, but stores the first 64 bytes of the packet before sending the packet on. The reason for this is that most errors and all collisions occur during the initial 64 bytes of a packet.

LAN switches vary in physical design. Currently, there are three popular configurations in use:

- **Shared−memory** The switch stores all incoming packets in a common memory buffer that all the switch **ports** (input/output connections) share. Then, the switch sends the packets out the correct port for the destination node.
- **Matrix** This type of switch has an internal grid with which the input ports and the output ports cross each other. When the switch detects a packet on an input port, the switch compares the MAC address to the lookup table to find the appropriate output port. The switch then makes a connection on the grid where these two ports intersect.
- **Bus−architecture** Instead of a grid, an internal transmission path (**common bus**) is shared by all the ports using time division multiplex access (TDMA). A switch with this configuration dedicates a memory buffer to each port. There is an application−specific integrated circuit (ASIC) to control the internal bus access.

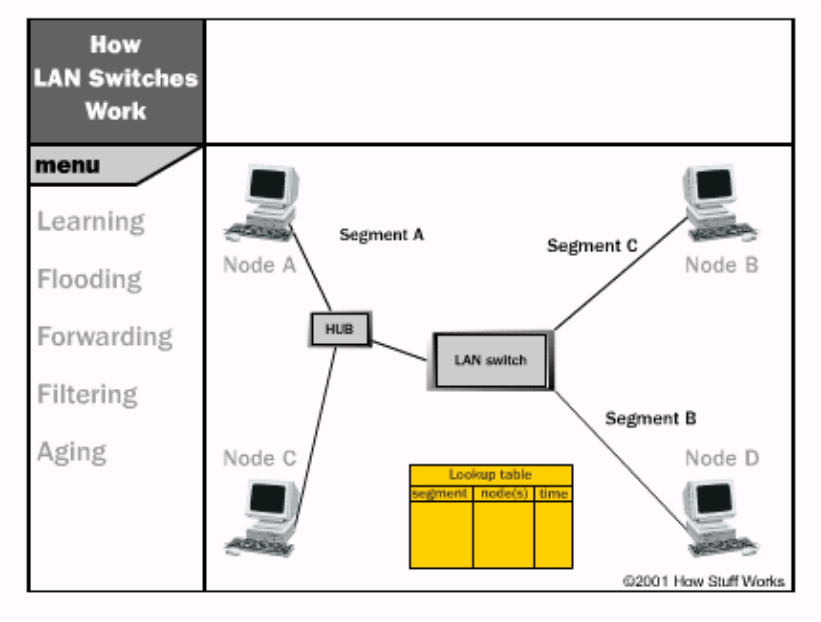
Transparent Bridging

Most Ethernet LAN switches use transparent bridging to create the address lookup tables. Transparent bridging technology allows a switch to learn everything that the switch needs to know about the location of nodes on the network without the need for the network administrator to do anything. Transparent bridging has five parts:

- Learning
- Flooding
- Filtering
- Forwarding
- Aging

Flash Animation: How Transparent Bridging Works

Click here to see Flash animation that teaches you more about how transparent bridging works. Click the Back button on your browser to return to this document.



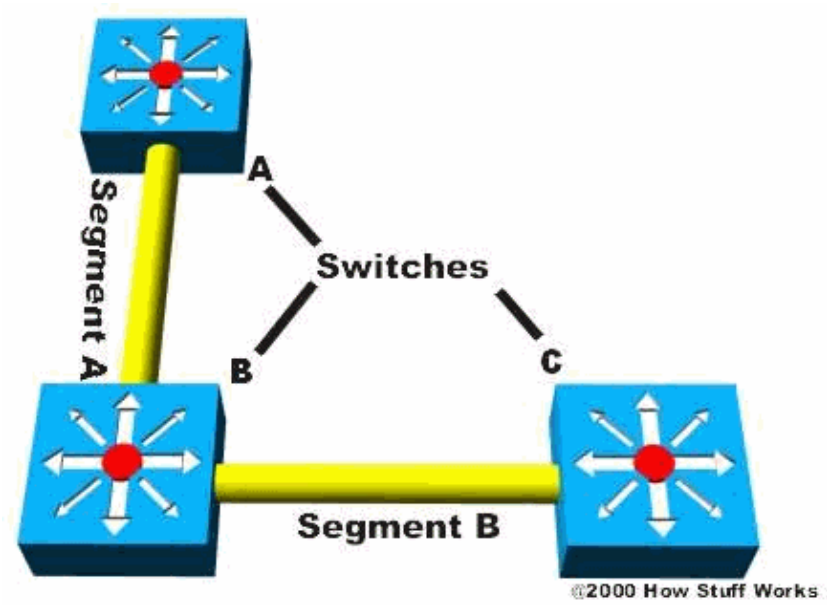
- The addition of the switch to the network occurs, and the various segments plug into the switch ports.
- The computer Node A on Segment A sends data to the computer Node B on another segment, Segment C.
- The switch gets the first packet of data from Node A. The switch reads the MAC address and saves the address to the lookup table for Segment A. The switch now knows where to find Node A whenever a packet with this address arrives. This process is **learning**.
- Since the switch does not know where Node B is, the switch sends the packet to all the segments. But the switch does not send the packet to the segment on which the packet arrived, Segment A. When a switch sends a packet out to all segments to find a specific node, this is **flooding**.
- Node B gets the packet and sends a packet back to Node A in acknowledgement.
- The packet from Node B arrives at the switch. Now the switch can add the MAC address of Node B to the lookup table for Segment C. Since the switch already knows the address of Node A, the switch sends the packet directly to the node. Because Node A is on a different segment than Node B, the switch must connect the two segments to send the packet. You call this action **forwarding**.
- The next packet from Node A to Node B arrives at the switch. The switch now has the address of Node B, too, so the switch forwards the packet directly to Node B.
- Node C sends information to the switch for Node A. The switch looks at the MAC address for Node C and adds the address to the lookup table for Segment A. The switch already has the address for Node A and determines that both nodes are on the same segment. The switch does not need to connect Segment A to another segment for the data to travel from Node C to Node A. Therefore, the switch ignores packets that travel between nodes on the same segment. This is **filtering**.

- The switch continues to learn and flood as it adds nodes to the lookup tables. Most switches have plenty of memory to maintain the lookup tables. But remove old information so that the switch does not waste time with a search through stale addresses. In order to optimize the use of this memory, switches use the **aging** technique. Basically, when a switch adds an entry to the lookup table for a node, the entry gets a time stamp. Each time the switch receives a packet from a node, the switch updates the time stamp. The switch has a user−configurable timer that erases the entry after a certain length of time of no activity from that node. The erasure frees up valuable memory resources for other entries. Transparent bridging is a good, essentially maintenance−free way to add all the information that a switch needs to operate.

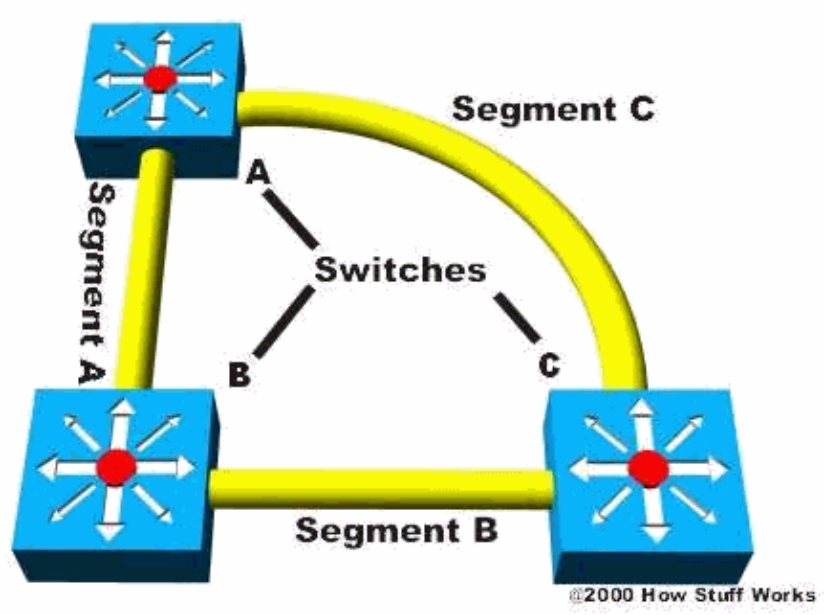
In the example, two nodes share each segment. In an ideal LAN−switched network, every node would have a separate segment. Separate segments would eliminate the possibility of collisions and the need for filtering. Notice that, while a node on Segment A talks to a node on Segment B at 10 Mbps, a node on Segment C can also communicate with a node on Segment B at 10 Mbps.

Redundancy and Broadcast Storms

The section The Addition of Switches mentions the possibility of a single point of failure in a network. In a star or starbus network, the point with the greatest potential to bring all or part of the network down is the switch or hub. See this example:

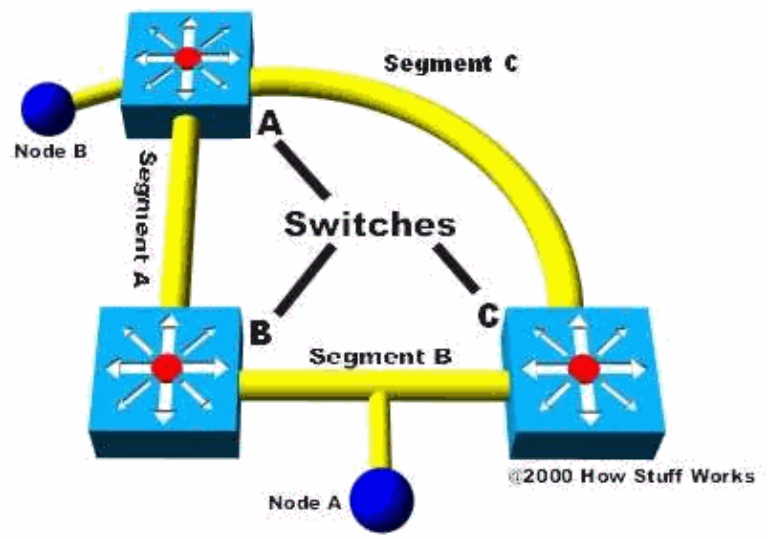


If either switch A or C fails, the failure affects the nodes that connect to that switch. But nodes at the other two switches can still communicate. If switch B fails, the failure brings down the entire network. What if you add another segment that connects switches A and C?



Even if one of the switches fails, the network continues. This setup provides redundancy and effectively eliminates the single point of failure.

Now there is a new problem. In the section Flash Animation: How Transparent Bridging Works, you discovered how switches learn the location of the nodes. Now all the switches connect in a loop, so a packet from a node can come to a switch from two different segments. For example, imagine that Node B connects to switch A and needs to communicate with Node A on Segment B. Switch A does not know Node A, so the switch floods the packet.



The packet travels via Segment A or Segment C to the other two switches, B and C. Switch B adds Node B to the lookup table that the switch maintains for Segment A. Switch C adds the node to the lookup table for Segment C. Suppose that neither switch has learned the address for Node A yet. The switches then flood Segment B to look for Node A. Each switch takes the packet that the other switch has sent and immediately floods the packet back out since the switch still does not know Node A. Switch A receives the packet from each segment and floods the packet back out on the other segment. A **broadcast storm** results at the broadcast, receipt, and rebroadcast of packets by each switch. A broadcast storm can cause potentially severe network congestion.

Spanning Tree

In order to prevent broadcast storms and other side effects of looping, **Digital Equipment Corporation** created the **Spanning Tree Protocol (STP)**. The IEEE has standardized STP as the 802.1D specification. Essentially, a spanning tree uses the spanning−tree algorithm (STA), which senses that the switch has more than one way to communicate with a node. The STA then determines which communication method is the best and blocks out the other paths. Also, STP keeps track of the other paths in case the primary path is unavailable.

Here is how STP works:

- Each switch is assigned a group of IDs, one for the switch itself and one for each port on the switch. The switch identifier is called the **Bridge ID (BID)**. The BID is 8 bytes long and contains a bridge priority, along with one of the switch MAC addresses. The bridge priority is 2 bytes, and the MAC address is 6 bytes. Each port ID is 16 bits long with two parts, a 6−bit priority and a 10−bit port number.
- A **path cost** value is given to each port. The cost is typically based on a guideline established as part of 802.1D. According to the original specification, cost is 1000 Mbps (1 Gbps) divided by the bandwidth of the segment that connects to the port. Therefore, a 10 Mbps connection has a cost of 100 (1000 divided by 10).

  The speed of networks has increased beyond the gigabit range, so there has been a slight modification of the standard cost. The new cost values are:

Bandwidth	STP Cost Value
4 Mbps	250
10 Mbps	100
16 Mbps	62
45 Mbps	39
100 Mbps	19
155 Mbps	14
622 Mbps	6
1 Gbps	4
10 Gbps	2

  Note: The path cost can be an arbitrary value that the network administrator assigns instead of one of the standard cost values.
- Each switch begins a discovery process to choose which network paths to use for each segment. Special network frames with the name **Bridge Protocol Data Units (BPDU)** share this information between all the switches. The parts of a BPDU are:

  - Root BID This is the BID of the current **root bridge**.
  - Path cost to root bridge Determines how far away the root bridge is. For example, if the data have to travel over three 100−Mbps segments to reach the root bridge, the cost is 38 (19 + 19 + 0). The segment that attaches to the root bridge normally has a path cost of 0.
  - Sender BID The BID of the switch that sends the BPDU.
  - Port ID The actual port on the switch from which this BPDU was sent.

All the switches constantly send BPDUs to each other in attempt to determine the best path between various segments. When a switch receives a BPDU from another switch that is better than the BPDU that the switch is broadcasting for the same segment, the switch stops broadcasting its BPDU out that segment. The switch instead stores the other switch's BPDU for reference and broadcasting out to **inferior segments**, such as segments that are farther away from the root bridge.

- A **Root Bridge** is chosen based on the results of the BPDU process between the switches. Initially, every switch considers itself the root bridge. When a switch first powers up on the network, the switch sends out a BPDU with its own BID as the root BID. When the other switches receive the BPDU, the switches compare the BID to the one they already have stored as the root BID. If the new root BID has a lower value, the switches replace the saved one. But if the saved root BID is lower, the switch sends a BPDU to the new switch with this BID as the root BID. When the new switch receives the BPDU, this switch realizes that it is not the root bridge. The switch replaces the root BID in the switch table with the new root BID. The result is that the switches elect as the root bridge the switch that has the lowest BID.
- Based on the location of the root bridge, the other switches determine which of their ports has the lowest path cost to the root bridge. These ports are called **root ports**. With the exception of the current root bridge, each switch must have one.
- The switches determine who will have **designated ports**. A designated port is the connection used to send and receive packets on a specific segment. The assignment of only one designated port per segment resolves all looping issues.

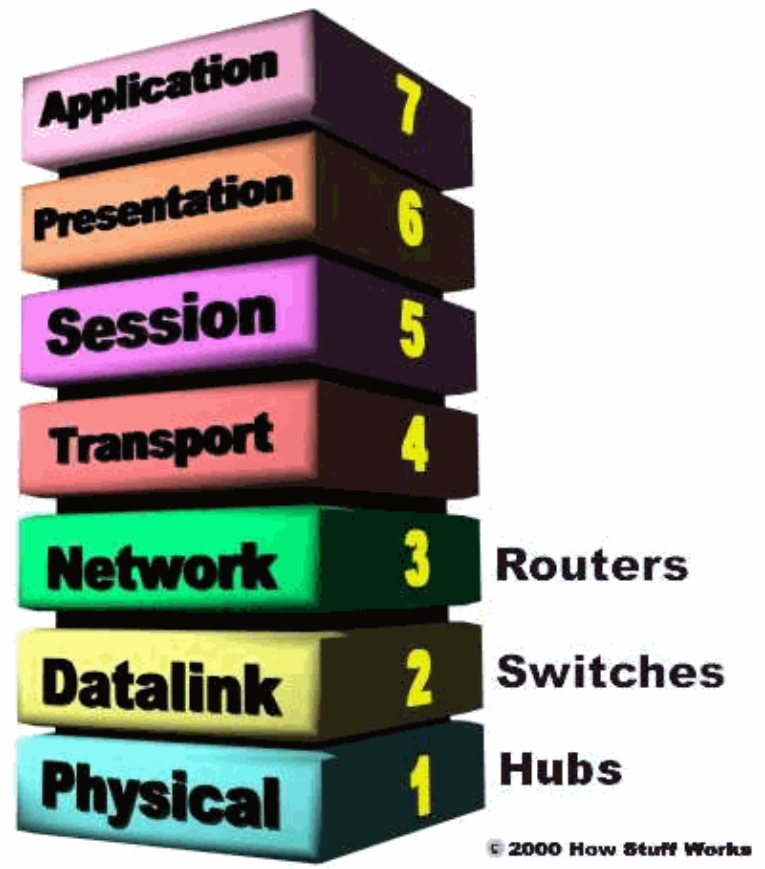
  Designated ports are selected based on the lowest path cost to the root bridge for a segment. Since the root bridge has a path cost of 0, any ports on the root bridge that connect to segments become designated ports. For the other switches, there is a path cost comparison for a specific segment. If one port has a lower path cost, that port becomes the designated port for that segment. If two or more ports have the same path cost, the choice is the switch with the lowest BID.
- After the choice of the designated port for a network segment, any other ports that connect to that segment become **nondesignated ports**. These ports block network traffic from that path so that the traffic can only access the segment through the designated port.

Each switch has a table of BPDUs that the switch continually updates. The network now has a single spanning tree configuration. The root bridge is the trunk and all the other switches are branches. Each switch communicates with the root bridge through the root ports and with each segment through the designated ports to maintain a loop−free network. In the event that the root bridge begins to fail or has network problems, STP allows the other switches to immediately reconfigure the network so that another switch acts as root bridge. This process gives a company the ability to have a complex network that is fault−tolerant yet fairly easy to maintain.

Routers and Layer 3 Switching

While most switches operate at the **Data layer (Layer 2)** of the OSI reference model, some incorporate features of a router and operate at the **Network layer (Layer 3)** also. In fact, a Layer 3 switch is incredibly similar to a router.

Like routers, Layer 3 switches actually work at the Network layer.



When a router receives a packet, the router looks at the Layer 3, or Network layer, source and destination addresses to determine the path for the packet to take. This activity is Layer 3 (Network) networking activity. A standard switch relies on the MAC addresses to determine the source and destination of a packet. This activity is Layer 2 (Data) networking. The fundamental difference between a router and a Layer 3 switch is that Layer 3 switches have optimized hardware to pass data as fast as Layer 2 switches. Yet Layer 3 switches make decisions on how to transmit traffic at Layer 3, just like a router. Within the LAN environment, a Layer 3 switch is usually faster than a router because the switch foundation is switching hardware. In fact, many Cisco Layer 3 switches are actually routers that Cisco built on "switching" hardware with customized chips inside the box.

The pattern matching and caching on Layer 3 switches is similar to the pattern matching and caching on a router. Both a Layer 3 switch and a router use a routing protocol and routing table to determine the best path. However, a Layer 3 switch can reprogram the hardware dynamically with the current Layer 3 routing information. This is what allows much faster packet processing. Current Layer 3 switches like the Cisco Catalyst 6500/6000 switches use the information from the routing protocols to update the hardware caching tables. The 6500/6000 is a great way to connect to the Internet because it has WAN cards; but simple routers of varying sizes are usually fine for connecting to the Internet based on traffic flow and budget.

Note: Routers are necessary for communication between two VLANs.

VLANs

As networks grow in size and complexity, many companies turn to **Virtual Local Area Networks (VLANs)** to provide some way to structure this growth logically. Basically, a VLAN is a collection of nodes that group together in a single **broadcast domain**. The basis for the group is something other than physical location. The section Switch Technologies describes broadcasts and how a router does not pass along broadcasts. A

broadcast domain is a network or portion of a network that receives a broadcast packet from any node within that network. In a typical network, everything on the same side of the router is part of the same broadcast domain. A switch on which you have implemented VLANs now has multiple broadcast domains, which is similar to a router. But you still need a router to route from one VLAN to another. The switch alone cannot perform this routing.

A company can choose to have VLANs for these common reasons:

- Security A separation of systems with sensitive data from the rest of the network provides security. This system separation decreases the chance that someone can gain access to information that the person does not have authorization to see.
- Projects/special applications You can simplify the management of a project or work with a special application with the use of a VLAN. Because the VLAN brings all the necessary nodes together, project management can be simpler.
- Performance/bandwidth The careful monitor of network use allows the network administrator to create VLANs that reduce the number of router hops. The VLANs can also increase the apparent bandwidth for network users.
- Broadcasts/traffic flow Since VLANs do not pass broadcast traffic to nodes that are not part of the VLAN, a VLAN automatically reduces broadcasts. **Access lists** provide the network administrator with a way to control who sees particular network traffic. An access list is a table that the network administrator creates. The table lists the addresses that have access to that network.
- Departments/specific job types Companies can set up VLANs for departments that are heavy network users, such as multimedia or engineering departments. A company can also set up a VLAN across departments and dedicate the VLAN to specific types of employees, such as managers or salespeople.

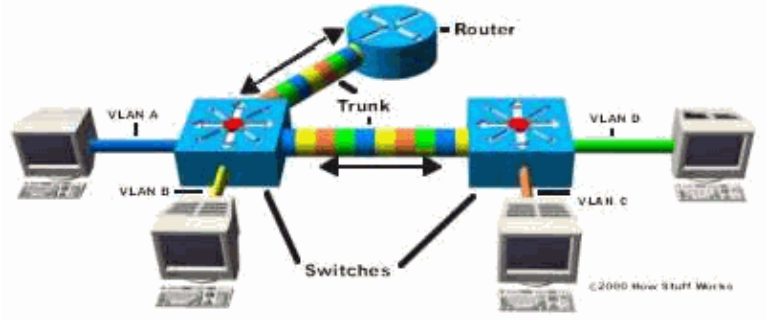
A managed switch is necessary for the creation or administration of VLANs. The term "managed" refers to the ability to manage and configure the switch properties, which includes VLAN creation. Unmanaged switches do not support VLANs. All Cisco Catalyst LAN switching products are managed switches.

You can create a VLAN with most switches. Simply log in to the switch via Telnet or the console port, and enter the parameters for the VLAN. These parameters are the name and the domain and port assignments. After you have created the VLAN, any network segments that connect to the ports that you have assigned become part of that VLAN.

While you can have more than one VLAN on a switch, the VLANs cannot communicate directly with each other. This VLAN communication defeats the purpose of a VLAN, which is to isolate a part of the network. Communication between VLANs requires the use of a router.

VLANs can span across multiple switches. Also, you can have more than one VLAN on each switch. For multiple VLANs on multiple switches to communicate via a single link between the switches, you must use **trunking**. Trunking is the technology that allows the carry of information from multiple VLANs over just one link between switches.

The **VLAN Trunk Protocol (VTP)** is the protocol that switches use to communicate between each other about VLAN configuration.



In this image, each switch has two VLANs. On the first switch, the send of VLAN A and VLAN B occurs through a single port, which is trunked. These VLANs go to both the router and, through another port, to the second switch. VLAN C and VLAN D are trunked from the second switch to the first switch and, through that switch, to the router. This trunk can carry traffic from all four VLANs. The trunk link from the first switch to the router can also carry all four VLANs. In fact, this one connection to the router actually allows the router to appear on all four VLANs. The appearance is that the router has four different physical ports with connection to the switch.

The VLANs can communicate with each other via the trunking connection between the two switches. This communication occurs with use of the router. For example, data from a computer on VLAN A that need to get to a computer on VLAN B must travel from the switch to the router and back again to the switch. Because of the transparent bridging algorithm and trunking, both PCs and the router think that they are on the same physical segment.

LAN switches can make a big difference in the speed and quality of your network.

NetPro Discussion Forums − Featured Conversations

Networking Professionals Connection is a forum for networking professionals to share questions, suggestions, and information about networking solutions, products, and technologies. The featured links are some of the most recent conversations available in this technology.

NetPro Discussion Forums − Featured Conversations for LAN
Network Infrastructure: LAN Routing and Switching
Network Infrastructure: Getting Started with LANs

Related Information

- **Internetworking Technology Overview: Ethernet Technologies**
- **whatis.com: switch**
- **University of New Hampshire InterOperability Laboratory Ethernet Tutorials**
- **LAN Product Support Pages**
- **LAN Switching Support Page**
- **Tools & Resources − Technical Support**
- **Technical Support − Cisco Systems**

Updated: Dec 15, 2005 Document ID: 10607