**BASIC OF SWITCHING**

[](https://en.wikipedia.org/wiki/File:2550T-PWR-Front.jpg)

50-port Ethernet switch

A network switch is also called switching hub, bridging hub, officially MAC bridgeis a computer networking device that connects devices together on a [computer network](https://en.wikipedia.org/wiki/Computer_network) by using [packet switching](https://en.wikipedia.org/wiki/Packet_switching) to receive, process, and forward data to the destination device.

A network switch is a multiport [network bridge](https://en.wikipedia.org/wiki/Network_bridge) that uses [hardware addresses](https://en.wikipedia.org/wiki/Hardware_address) to process and forward data at the [data link layer](https://en.wikipedia.org/wiki/Data_link_layer) (layer 2) of the [OSI model](https://en.wikipedia.org/wiki/OSI_model). Some switches can also process data at the [network layer](https://en.wikipedia.org/wiki/Network_layer) (layer 3) by additionally incorporating [routing](https://en.wikipedia.org/wiki/Routing) functionality. Such switches are commonly known as layer-3 switches or [multilayer switches](https://en.wikipedia.org/wiki/Multilayer_switch).[[2]](https://en.wikipedia.org/wiki/Network_switch#cite_note-layer3-2)

Switches for [Ethernet](https://en.wikipedia.org/wiki/Ethernet) are the most common form of network switch. The first Ethernet switch was introduced by [Kalpana](https://en.wikipedia.org/wiki/Kalpana_(company)" \o "Kalpana (company)) in 1990. Switches also exist for other types of networks including [Fibred Channel](https://en.wikipedia.org/wiki/Fibre_Channel), [Asynchronous Transfer Mode](https://en.wikipedia.org/wiki/Asynchronous_Transfer_Mode), and [InfiniBand](https://en.wikipedia.org/wiki/InfiniBand" \o "InfiniBand).

Unlike less advanced [repeater hubs](https://en.wikipedia.org/wiki/Repeater_hub), which broadcast the same data out of each of its ports and let the devices decide what data they need, a network switch forwards data only to the devices that need to receive it.[[4]](https://en.wikipedia.org/wiki/Network_switch#cite_note-4)

A switch is a device in a [computer network](https://en.wikipedia.org/wiki/Computer_network) that connects together other devices. Multiple data cables are plugged into a switch to enable communication between different networked devices. Switches manage the flow of data across a network by transmitting a received [network packet](https://en.wikipedia.org/wiki/Network_packet) only to the one or more devices for which the packet is intended. Each networked device connected to a switch can be identified by its [network address](https://en.wikipedia.org/wiki/Network_address), allowing the switch to direct the flow of traffic maximizing the security and efficiency of the network.

A switch is more intelligent than an [Ethernet hub](https://en.wikipedia.org/wiki/Ethernet_hub), which simply retransmits packets out of every port of the hub except the port on which the packet was received, unable to distinguish different recipients, and achieving an overall lower network efficiency.

An Ethernet switch operates at the [data link layer](https://en.wikipedia.org/wiki/Data_link_layer) (layer 2) of the OSI model to create a separate [collision domain](https://en.wikipedia.org/wiki/Collision_domain) for each switch port. Each device connected to a switch port can transfer data to any of the other ports at any time and the transmissions will not interfere.[[a]](https://en.wikipedia.org/wiki/Network_switch#cite_note-6) Because [broadcasts](https://en.wikipedia.org/wiki/Broadcasting_(networking)) are still being forwarded to all connected devices by the switch, the newly formed network segment continues to be a [broadcast domain](https://en.wikipedia.org/wiki/Broadcast_domain). Switches may also operate at higher layers of the OSI model, including the network layer and above. A device that also operates at these higher layers is known as a *multilayer switch*.

Segmentation involves the use of a switch to split a larger collision domain into smaller ones in order to reduce collision probability, and to improve overall network throughput. In the extreme case (i.e. micro-segmentation), each device is located on a dedicated switch port. In contrast to an Ethernet hub, there is a separate collision domain on each of the switch ports. This allows computers to have dedicated bandwidth on point-to-point connections to the network and also to run in full-duplex mode. Full-duplex mode has only one transmitter and one receiver per collision domain, making collisions impossible.

The network switch plays an integral role in most modern Ethernet [local area networks](https://en.wikipedia.org/wiki/Local_area_network) (LANs). Mid-to-large sized LANs contain a number of linked managed switches. [Small office/home office](https://en.wikipedia.org/wiki/Small_office/home_office) (SOHO) applications typically use a single switch, or an all-purpose device such as a [residential gateway](https://en.wikipedia.org/wiki/Residential_gateway) to access small office/home [broadband](https://en.wikipedia.org/wiki/Broadband) services such as [DSL](https://en.wikipedia.org/wiki/Digital_Subscriber_Line) or [cable Internet](https://en.wikipedia.org/wiki/Cable_Internet). In most of these cases, the end-user device contains a [router](https://en.wikipedia.org/wiki/Router_(computing)) and components that interface to the particular physical broadband technology. User devices may also include a telephone interface for [Voice over IP](https://en.wikipedia.org/wiki/Voice_over_IP) (VoIP).

**Role of switches in a network:**

Switches are most commonly used as the network connection point for hosts at the edge of a network. In the [hierarchical internetworking model](https://en.wikipedia.org/wiki/Hierarchical_internetworking_model) and similar network architectures, switches are also used deeper in the network to provide connections between the switches at the edge.

In switches intended for commercial use, built-in or modular interfaces make it possible to connect different types of networks, including [Ethernet](https://en.wikipedia.org/wiki/Ethernet), [Fibre Channel](https://en.wikipedia.org/wiki/Fibre_Channel" \o "Fibre Channel), [RapidIO](https://en.wikipedia.org/wiki/RapidIO" \o "RapidIO), [ATM](https://en.wikipedia.org/wiki/Asynchronous_Transfer_Mode), [ITU-T](https://en.wikipedia.org/wiki/ITU-T) [G.hn](https://en.wikipedia.org/wiki/G.hn) and [802.11](https://en.wikipedia.org/wiki/802.11). This connectivity can be at any of the layers mentioned. While the layer-2 functionality is adequate for bandwidth-shifting within one technology, interconnecting technologies such as [Ethernet](https://en.wikipedia.org/wiki/Ethernet) and [token ring](https://en.wikipedia.org/wiki/Token_ring) is performed more easily at layer 3 or via routing. Devices that interconnect at the layer 3 are traditionally called [routers](https://en.wikipedia.org/wiki/Router_(computing)), so layer 3 switches can also be regarded as relatively primitive and specialized routers.

Where there is a need for a great deal of analysis of network performance and security, switches may be connected between WAN routers as places for analytic modules. Some vendors provide [firewall](https://en.wikipedia.org/wiki/Firewall_(computing)), network [intrusion detection](https://en.wikipedia.org/wiki/Intrusion_detection), and performance analysis modules that can plug into switch ports. Some of these functions may be on combined modules.

Through [port mirroring](https://en.wikipedia.org/wiki/Port_mirroring), a switch can create a mirror image of data that can go to an external device such as [intrusion detection systems](https://en.wikipedia.org/wiki/Intrusion_detection_system) and [packet sniffers](https://en.wikipedia.org/wiki/Packet_sniffer).

A modern switch may implement [power over Ethernet](https://en.wikipedia.org/wiki/Power_over_Ethernet) (PoE), which avoids the need for attached devices, such as a [VoIP phone](https://en.wikipedia.org/wiki/VoIP_phone) or [wireless access point](https://en.wikipedia.org/wiki/Wireless_access_point), to have a separate power supply. Since switches can have redundant power circuits connected to [uninterruptible power supplies](https://en.wikipedia.org/wiki/Uninterruptible_power_supply), the connected device can continue operating even when regular office power fails.

**Types of switches:**

Switches are available in many form factors, including stand-alone, desktop units which are typically intended to be used in a home or office environment outside a [wiring closet](https://en.wikipedia.org/wiki/Wiring_closet); rack-mounted switches for use in an [equipment rack](https://en.wikipedia.org/wiki/19_inch_rack) or an [enclosure](https://en.wikipedia.org/wiki/Enclosure_(electrical)), also as large chassis units with swappable module cards; [DIN rail](https://en.wikipedia.org/wiki/DIN_rail) mounted for use in [industrial environments](https://en.wikipedia.org/wiki/Industrial_Ethernet); and small installation switches, mounted into a cable duct, floor box or communications tower, as found, for example, in [FTTO Infrastructures](https://en.wikipedia.org/wiki/Fibre_to_the_Office).

**Configuration options:-**

* Unmanaged switches – these switches have no configuration interface or options. They are [plug and play](https://en.wikipedia.org/wiki/Plug_and_play). They are typically the least expensive switches, and therefore often used in a [small office/home office](https://en.wikipedia.org/wiki/SOHO_network) environment. Unmanaged switches can be desktop or rack mounted.
* Managed switches – these switches have one or more methods to modify the operation of the switch. Common management methods include: a [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface) (CLI) accessed via [serial console](https://en.wikipedia.org/wiki/Serial_console), [telnet](https://en.wikipedia.org/wiki/Telnet) or [Secure Shell](https://en.wikipedia.org/wiki/Secure_Shell), an embedded [Simple Network Management Protocol](https://en.wikipedia.org/wiki/Simple_Network_Management_Protocol) (SNMP) agent allowing management from a remote console or management station, or a web interface for management from a [web browser](https://en.wikipedia.org/wiki/Web_browser). Examples of configuration changes that one can do from a managed switch include: enabling features such as [Spanning Tree Protocol](https://en.wikipedia.org/wiki/Spanning_Tree_Protocol) or port mirroring, setting [port bandwidth](https://en.wikipedia.org/wiki/Bit_rate), creating or modifying [virtual LANs](https://en.wikipedia.org/wiki/Virtual_LAN) (VLANs), etc.

**Two sub-classes of managed switches are marketed today:**

1. **Smart (or intelligent) switches** – these are managed switches with a limited set of management features. Likewise "web-managed" switches are switches which fall into a market niche between unmanaged and managed. For a price much lower than a fully managed switch they provide a web interface (and usually no CLI access) and allow configuration of basic settings, such as VLANs, port-bandwidth and duplex.[[18]](https://en.wikipedia.org/wiki/Network_switch#cite_note-19)
2. **Enterprise managed (or fully managed) switches** – these have a full set of management features, including CLI, SNMP agent, and web interface. They may have additional features to manipulate configurations, such as the ability to display, modify, backup and restore configurations. Compared with smart switches, enterprise switches have more features that can be customized or optimized, and are generally more expensive than smart switches. Enterprise switches are typically found in networks with larger number of switches and connections, where centralized management is a significant savings in administrative time and effort. A [stackable switch](https://en.wikipedia.org/wiki/Stackable_switch) is a version of enterprise-managed switch.

**Basic switch configuration:**

**Basic switch functions, names and passwords:**

The switch name is tool to let us see what device we are connected to. The prompt will display the name of the switch so

SW1>

Tells us that we are connected to a switch named 'SW1'. The prompt also tells us another thing, “where” in the different hierarchical modes of the switch we are. The switch has three **basic modes, unprivileged, privilege (or enable) and configuration mode**. The prompts are, in the same order:

SW1>

SW1#

SW1(config)#

The configuration mode actually has a few sub-modes like interface configuration and line configuration:

SW1(config-if)#

SW1(config-line)#

Some features, like the configuration VLAN, have their own sub-modes.

**Moving between modes**

Move between modes is done by calling the “name” of the mode if you want to move up in the hierarchy and exit or end if you want to move down:

SW1>enable

SW1#configure terminal

SW1(config)#interface FastEthernet 0/1

SW1(config-if)#exit

SW1(config)#interface line console 0

SW1(config-line)#end

SW1#

Notice how the move from line configuration to privilege mode differs from the move from interface configuration to configuration mode? The command exit will move you down one step while end will take you all the way back to privilege mode no matter where you start.

**Configuring a name**

The configuration mode is mainly used for configuration that will affect the “whole” switch (in contrast to interface configuration mode that will only affect the specified interface or interfaces). To change the name, move to configuration mode and execute the following command:

SW1(config)#hostname newHostname

newHostname(config)#

**Command interpretation**

When the switch interprets the commands entered, it compares the command to the possible commands in that mode and if there is a single match with the characters given the switch executes the command.

An example might make it clear. Let's say we want to move from unprivileged to privilege mode. The command is enable.

SW1>enable

SW1#exit

SW1>en

SW1#

The same thing can be done with every command. As long as there's no other command sharing the characters given, the switch will accept the command as the one it can translate to. The hostname can therefore be set with the command: SW1(config)#host newHostname

newHostname(config)#

**Disabling DNS look-up**

Apart from the command interpretation and shortening, the switch will interpret any unknown single command in unprivileged or privileged mode as an attempt to make a telnet connection. This can be quite annoying since a spelling error for enable (let's say enable) could turn into a waiting period while the DNS times out the switch realizes that it can't find an IP for 'enable'. The lookup will be done even if the switch don't have an IP enable interface.

To disable the DNS look-up (the telnet feature will still be there but the switch will know that it won't be able to make the translation to an IP address and therefore abort immediately) just issue the following command:

SW1(config)#no ip domain-lookup

Depending on the software, the domain-lookup part might be split into two (domain lookup).

**The 'no' keyword**

As we see in the command to disable DNS look-ups the keyword 'no' is set before the command. To enable look-up, just issue the command without the no. This is the standard way to turn off function in Cisco IOS. For example if we want to enable an interface, we issue the no shutdown command and if we want to disable it, we just issue shutdown.

**Passwords**

Passwords can be configured to control how can access what on the switch. Different passwords can be used to limit access to:

• The switch via the console (unprivileged mode)

• The switch via the network (unprivileged mode)

• Privilege mode

Since the equipment in the lab is shared between multiple students groups there are only three allowed passwords in the lab: class, cisco and password. For clarity, only the passwords class and cisco will be used in this document.

**Privilege mode password (enable password)**

There are actually two ways to configure the ”enable” password for the switch, one in plain text (keyword password) and one that's encrypted using a special form of the MD5 hash (keyword secret). This example will only show how to set the encrypted password:

SW1(config)#enable secret cisco

Whenever a user tries to move from the unprivileged to the privilege modes, a prompt will appear asking for the password.

**Password for the console**

To limit the access via the console port (used for initial and on site configuration)

use the following string of commands:

SW1(config)#line console 0

SW1(config-line)#password class

SW1(config-line)#login

This will set the password 'class' for the console line. Whenever someone connect to the port, they will be prompted for a password.

**Password for network access**

Since it's not always possible to make a physical connection to the switch, it might be a good idea to make it accessible via the network. The simplest way is to allow access via telnet, only prompting for a password:

SW1(config)#line vty 0 4

SW1(config-line)#password class

SW1(config-line)#login

The login command is default for the VTY and the command can be used to disable access via the network (no login).

**Configuring a banner**

A banner can be used to give information to some that connects to the system. A good idea is to explain that the system is private, that you have to be authorized to access the system and that any attempt to connect is logged. The banner text is started and ended with an escape character – make sure to select one that you won't use in you banner text.

SW1(config)#banner login \*

Enter TEXT message. End with the characters '\*'.

PRIVATE SYSTEM!

\*

SW1(config)#

**VLAN, access and trunk ports:**

The switch can be used to create one big happy LAN. Just connect whatever needs network access and let them communicate. Need more ports? Just add another switch. As long as you want all your users connected to the same link you’re done.

If you on the other hand want to separate them from one another, maybe to gain some lever of control on the IP level or just to make sure that two nodes can't communicate with one another, you might want to create some sort of logical wall between them, forcing them to go through your router in order to communicate. Enter the 'virtual local area network' (VLAN).

The process is fairly simple: number every VLAN with a unique ID (VID), tell the switch what VID should be associates with a specific port and you’re done. Nodes connected to ports sharing VID can communicate, and nodes connected to ports associated with different VID can't (at least not at the link layer). But how about connecting switches together have more ports (or connecting to locations)? Either use one link per VID (expensive) or use what Cisco calls a 'trunk' (other vendors might talk about 'tagging', from the act of inserting a VID in the frame when it passes between the switches).

The 'trunk' is one type of port, used between switches (or between a switch and a device that is VLAN aware and has the need to send traffic on more than one VLAN). We'll start by looking at VLAN and then move over to the trunk port and its cousin, the access port.

**Configuring VLAN**

You can create, name and delete VLAN in the following way:

SW1(config)#vlan 10

SW1(config-vlan)#name LAN\_A

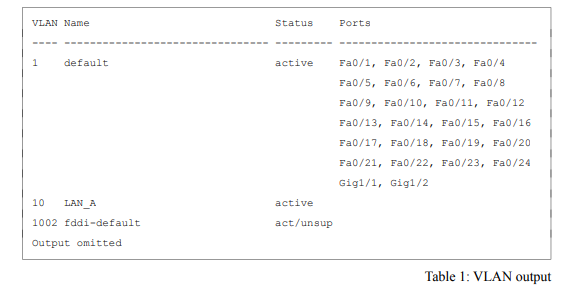
As you can see from the example, we can both create and name the VLAN. The see the VLAN configuration of the switch:

SW1#show vlan brief

The output from the command can be found in Table 1: VLAN output. As can be seen in the output, all port are associated with VLAN 1, but there is a VLAN with VID 10 named LAN\_A.

**Configuring access ports**

Ports used by non VLAN aware nodes, such as hosts or some routers are called access ports. The access port is normally used to connect one node to the network and the port is associated with a VLAN.



The following command can be used to tell a switch that a interface FastEthernet 0/4 is an access port: SW1(config)#interface FastEthernet 0/4

SW1(config-if)#switchport mode access

In order to make sure that there are no loops in the network, the switches run a protocol called the spanning-tree protocol (STP). This protocol cycles every interface to a series of states when it’s activated to make sure that the interface won't form a loop through other switches back to itself. Since the access port only should connect to end-nodes (that is, nodes that are the source or destination of traffic but never a transit node) – you can tell the switch to skip the stages and move directly to a forwarding state using the command:

SW1(config-if)#spanning-tree portfast

If you think that this is too much configuration, search the Web for information about the switchport host command.

**Associating the port with a VLAN**

In order to tell the switch what VLAN an access port should be associated with, use the following command (in this case to associate it with VLAN 10):

SW1(config-if)#switchport access vlan 10

The association can be changed at any time by giving it the same command with a different VLAN.

**Configuring trunk ports**

Links connecting two switches are normally configured as trunks (if there are more than one VLAN). The trunks use the IEEE 802.1Q standard for tagging frames in order for the receiving switch to know what VLAN a frame is associated with. In order for a port to be a trunk port configure it with the following command:

SW1(config-if)#switchport mode trunk

On switches with support for the Cisco trunking protocol ISL, you have to tell the switch how the VLAN information is to be inserted into the frame. To use 802.1Q, issue this command:

SW1(config-if)#switchport trunk encapsulation dot1q

**The 'native' VLAN**

The norm on the trunk is that the VLAN is 'tagged', that is, the frame has a marker that tells the receiver what VLAN the frame belongs to. One VLAN can be 'untagged', that is, frames sent on that VLAN have no tagging inserted and any frame that is untagged will be associated with that VLAN. The untagged, or native VLAN as Cisco calls it, can be used by clients connected to a link that is configured as a trunk between two switches (let's say, for example, via a hub). The default native VLAN is the same as the default VLAN, that is, VLAN 1. To change the native VLAN, use the following command (in this case the native VLAN is set to VID 50):

SW1(config-if)#switchport trunk native vlan 50

Make sure to use the same native VLAN on both sides of the trunk.

**SVI, IP address and telnet:**

The switch will need an interface with IP configuration in order to be accessible via the network. This interface is a virtual interface associated with a specific VID. Some switches are able to have more than one active virtual interface (or SVI) at a time but we'll be satisfied with using just one.

**Configuring the SVI and IP address**

Let's configure the IP address 10.0.0.10 with 24-bit subnetmask on a virtual interface associated with VID 10:

SW1(config)#interface vlan 10

SW1(config-if)#ip address 10.0.0.10 255.255.255.0

SW1(config-if)#no shutdown

The last command, the no shutdown, might not be needed as the SVI probably activates when created – but it's good practice to always make sure that the interface is not in the 'shutdown' state since we don't like troubleshoot that kind of simple mistakes.

**Revisiting telnet**

If we've set a password on the VTY, we can now access the switch via telnet, pointing our client application at the IP address we just configured. But a VTY password is not enough, we also need the 'enable' password in order to get the access we want.

**Working with the configuration**

If we want to view the fruit of our labor, we can print the active configuration to our console using the show **running-config** command. Use 'space' or 'enter' to move forward in the configuration and 'q' to not print any more. If you want to save your configuration so it’s not lost when you reboot the switch, issue the command **copy running-config startup-config** and to view the saved configuration, just type **show startup-config.**