Basic switch configuration

Cisco IOS

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 an 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 no 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 student 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 your 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 the 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 your 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 it's 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.

VLAN Name	Status Ports	
1 default	active Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24	
10 LAN_A 1002 fddi-default Output omitted	Gig1/1, Gig1/2 active act/unsup	

Table 1: VLAN output

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 its 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 to 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 its 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.