CCNA Lab – Switch Exploration

Topology Overview

PC-1 and PC-2 are connected to SW1 which has replaced the hub. PC-3 will be connected later in the lab.

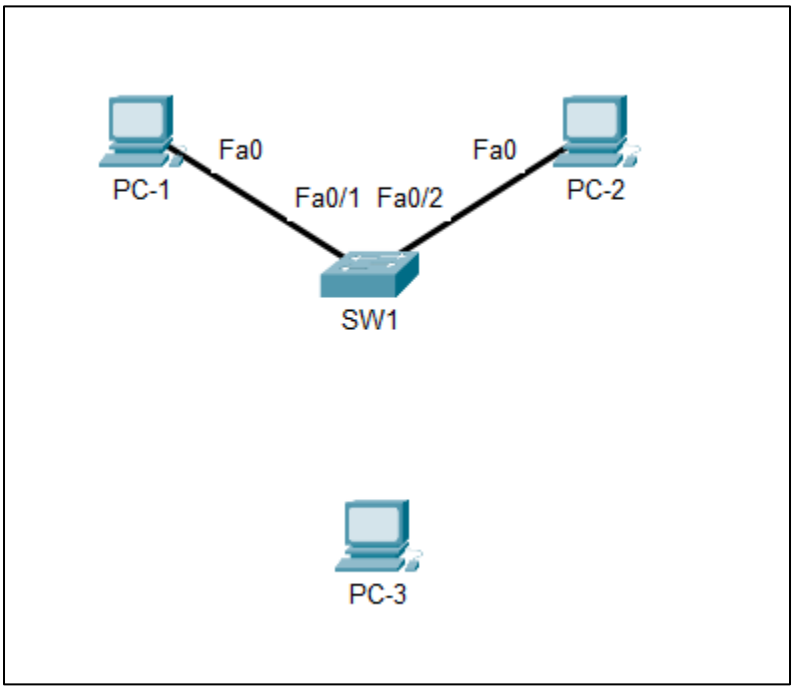


Figure 1 – Initial Lab Topology

In a previous lab, several problems relating to hub behavior were discovered. One problem was the hub's behavior of forwarding an incoming frame out all other ports, an action called **flooding**. Every connected device receives the frame regardless of whether it is the intended recipient. Each device must determine whether the destination MAC address belongs to it and discards the frame if it does not.

Imagine every time someone wanted to speak to your coworker, the conversation was also sent to you. You have no immediate way of knowing it isn't meant for you, so you have to stop and check before realizing, "Nope, that's not for me.

Another problem is that all ports on a hub are part of **one collision domain**. If two devices transmit at the same time, their signals can collide and the frames become corrupted. The affected data may then need to be transmitted again.

Imagine having to repeat yourself because someone else started talking at the same time and neither conversation could be understood.

The solution to these problems is a device called a switch.

a. Access SW1 and hit enter.

Observe

You are now accessing the switch's **Command Line Interface (CLI)**, where you type instructions to the switch. This is similar to using a **Graphical User Interface (GUI)** on a Windows computer, except instead of clicking buttons, icons, and menus, you interact with the switch by typing commands.

This area of the CLI is called **User EXEC mode**. The commands and privileges available in this mode are limited.

```
Press RETURN to get started!


Switch>
```

Figure 2 – User EXEC mode

b. On SW1, type enable.

```
Switch>enable
```

Observe

You have now entered **Privileged EXEC mode**, and the prompt has changed from > to #. A much larger set of commands is available in this mode, though most are used to view and manage the operational status of the switch.

You do not need to commit the modes to memory just yet. We will be working in the CLI throughout the labs, so you will become familiar with them through practice. For now, I just wanted to provide some foundational knowledge.

```
Switch>enable
Switch#
```

Figure 3 – Privilege EXEC mode

Engineer Insight

Switches, along with other Cisco networking devices, have several different modes.

c. View the status of SW1's interfaces.

```
Switch# show interface status
```

Observe

The command's output displays all the switch's interfaces along with valuable information about their current state. For now, the focus is the **Status** column.

```
Switch#show interface status
Port      Name               Status       Vlan       Duplex  Speed Type
Fa0/1                        connected    1          a-full  a-100 10/100BaseTX
Fa0/2                        connected    1          a-full  a-100 10/100BaseTX
Fa0/3                        notconnect   1          auto    auto  10/100BaseTX
Fa0/4                        notconnect   1          auto    auto  10/100BaseTX
Fa0/5                        notconnect   1          auto    auto  10/100BaseTX
```

Figure 4 – SW1 show interfaces status

Engineer Insight

A status of **connected** signifies that the switch senses a device connected to that port and the link is operational. A status of **notconnect** means the switch does not sense an active connection on that port.

d. Connect PC-3 to SW1's Fa0/3 with a straight through cable then check the interface statuses.

Observe

The status of that port has changed from **notconnect** to **connected**. SW1 now senses an active connection to PC-3 on Fa0/3.

```
Switch#show interfaces status
Port      Name               Status       Vlan       Duplex  Speed Type
Fa0/1                        connected    1          a-full  a-100 10/100BaseTX
Fa0/2                        connected    1          a-full  a-100 10/100BaseTX
Fa0/3                        connected    1          a-full  a-100 10/100BaseTX
Fa0/4                        notconnect   1          auto    auto  10/100BaseTX
```

Figure 5 – SW1 show interfaces status output

e. On SW1, add interface descriptions so you can easily identify which device is connected to each port, then view the interface statuses. Use **PC-X** as the description, where **X** is the device's number.

```
Switch# configure terminal
Switch(config)# interface fa0/X
Switch(config-if)# description PC-X
```

Observe

The interfaces connected to PC-1, PC-2, and PC-3 now have descriptions.

```
Switch#show interfaces status
Port      Name               Status       Vlan       Duplex  Speed Type
Fa0/1     PC-1               connected    1          a-full  a-100 10/100BaseTX
Fa0/2     PC-2               connected    1          a-full  a-100 10/100BaseTX
Fa0/3     PC-3               connected    1          a-full  a-100 10/100BaseTX
Fa0/4                        notconnect   1          auto    auto  10/100BaseTX
```

Figure 6 – SW1 show interfaces status output

Engineer Insight

The command **configure terminal** is used to enter **Global Configuration mode**, where changes to the device's configuration are made. Note that the prompt changed to **(config)#**, then to **(config-if)#** after entering **Interface Configuration mode**.

f. If you are still in config mode, enter exit then view SW1's MAC table.

```
Switch# show mac address-table
```

Observe

The MAC table is empty.

```
SW1#show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

SW1#
```

Figure 7 – SW1show mac address-table (Empty)

g. Turn on **Simulation Mode**, then ping PC-3 from PC-1. Progress the simulation until the frame arrives at SW1, then view the MAC table.

Observe

There is now one entry in SW1's MAC table.

```
SW1#show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    5254.0046.ac7b    DYNAMIC     Fa0/1
SW1#
```

Figure 8 – SW1 show mac address-table (One MAC)

Question

- Which system's MAC do you see?
- What can you assume about the way a switch learns based on the MAC in the table?

Engineer Insight

A switch learns the **source MAC address** of frames arriving on a port. The logic is that if a device sends a frame and that frame arrives on a specific port, that device must be reachable through that port. The switch places this **MAC-to-port mapping** in its MAC address table for future reference when determining where to forward frames.

h. Progress the simulation until the frame from PC-1 arrives at PC-3. View the MAC table.

Observe

The frame from PC-1 was forwarded out the ports toward PC-2 and PC-3. PC-3's MAC is now in SW1's MAC table.

```
SW1#show mac address-table
          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----

   1    5254.0046.ac7b    DYNAMIC     Fa0/1
   1    5254.0062.65fb    DYNAMIC     Fa0/3
SW1#
```

Figure 9 – Initial Lab Topology

There's A Problem

The switch can deliver frames to a specific device's port, but only after it has learned the device's MAC address and location. PC-3 has not sourced any frames yet, so the switch has not learned where PC-3 is located.

A frame destined for a MAC address that the switch has not learned is called an **unknown unicast**. When an unknown unicast arrives, the switch **floods** the frame out every port except the port on which it arrived.

The Solution

When a switch needs to forward an **unknown unicast**, it sends the frame out all other ports in the same way a hub would. This action is called **flooding**.

Engineer Insight

This resolves one of the problems with a hub. A hub must flood every frame, while a switch can use its MAC address table to forward a frame only out the specific port through which the intended destination can be reached.

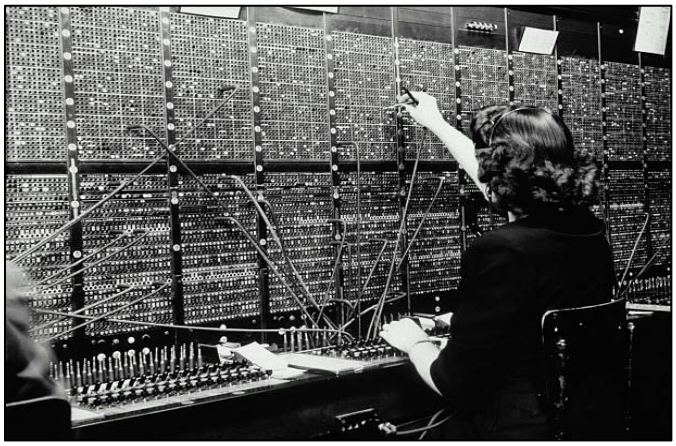
The logic behind flooding when the destination port is unknown is that the switch assumes the destination is reachable through one of its other ports. Flooding gives the frame the best chance of reaching its intended destination.

i. Turn on Simulation Mode. Ping the broadcast IP of 192.168.1.255 from PC-1 and PC-2 Allow the simulation to play out.

Observe

The frames are not colliding.

Engineer Insight



An easy mental model is to think of the switch as a **switchboard operator**, quickly connecting one port to another as needed. Unlike with a hub, frames from devices on different ports do not collide because every switchport is a separate **collision domain**.

Whereas all ports on a hub behave like they are part of **one shared logical cable** and one **collision domain**, a switch treats each port and its connection as an individual segment. Each switchport represents a separate **LAN segment** and **collision domain**.

You may be wondering what happens if frames are traveling in opposite directions on the same segment at the same time. That's a good question, and we'll answer it later. For now, assume that frames traveling in opposite directions cannot collide.