6
Switches and Switching - Forwarding Traffic on a Local Network

In the introduction to `Chapter 2`, *Understanding Local Area Networks*, I mentioned how most network engineers *cut their teeth* supporting an organization's internal infrastructure. This will ultimately mean gaining exposure to switches in some form or another. While it isn't likely that you'll be fully configuring switches at the entry-level, it is important that you understand how they function. The abundance of multi-user environments that are used both at home and at work makes it important for traffic to be forwarded quickly and efficiently.

This chapter introduces the concept of switching and provides an explanation of how switching decisions are made. We will discuss the different types of switch, their use cases, and the key attributes of these essential devices.

The following topics will be covered in this chapter:

- The purpose of switches and switching
- Understanding frame forwarding
- Understanding switch characteristics

Technical requirements

There are no technical requirements for this chapter.

The purpose of switches and switching

In `Chapter 2`, *Understanding Local Area Networks*, we introduced the use of switches within a LAN. Recall that a switch is a device that's used to forward traffic from one device to the next within a local network. By introducing a switch to our network, we are providing a means of segmenting the network into smaller, more manageable, and more efficient areas.

At a basic level, this segmentation leads to a reduction in collisions since each port on a switch is classed as its own collision domain. In the following diagram, each computer is connected to a port on the switch in the center and has formed its own collision domain:

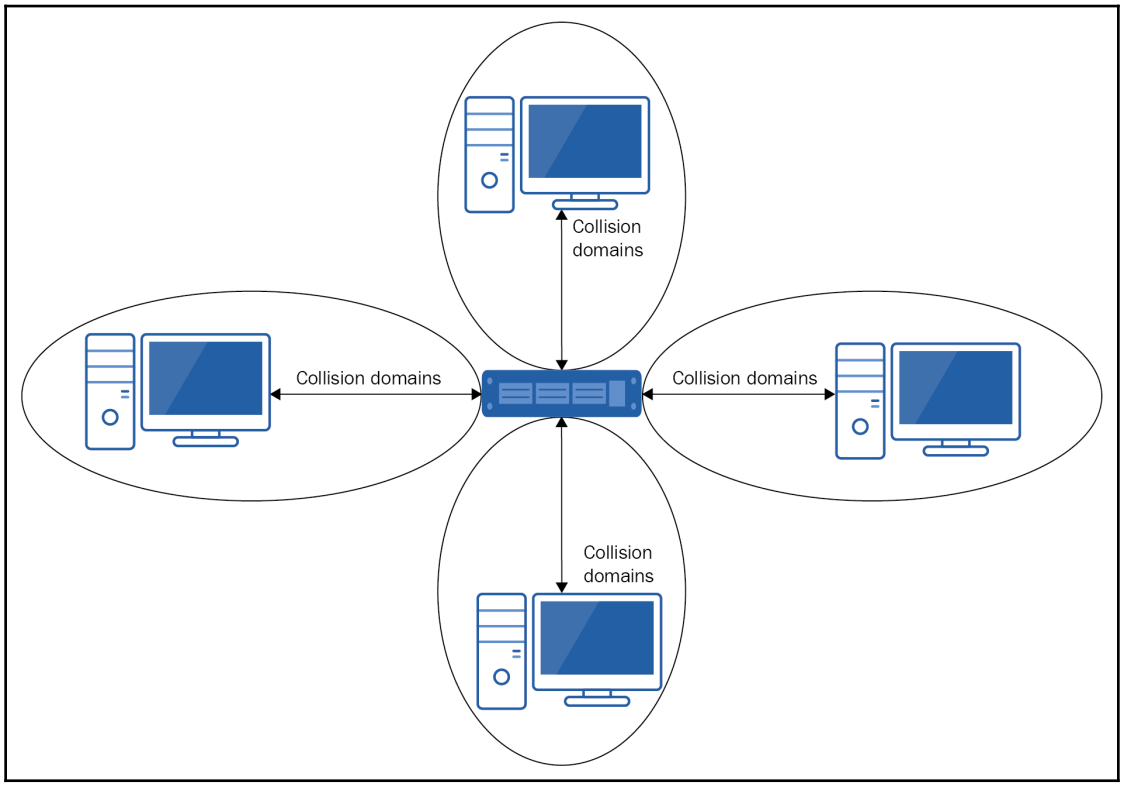

Figure 6.1: Collision domains

Notice that this segmentation reduces collisions but doesn't remove them. The reason for this is that, in most modern switched networks, a switch port is only connected to one device, thus providing zero collisions in a properly configured network. However, if a hub is connected to the switch port or has been incorrectly configured, this would lead to collisions still being detected. Further segmentation can be provided through the use of VLANs.

Now that we know why we should provide our network with a switch, let's look at how frames are forwarded.

Understanding frame forwarding

Recall that, unlike hubs, switches have intelligence built into them to prevent devices from receiving all the data being sent on the network, even if it isn't destined for them. In this section, I will walk you through the frame forwarding process that a switch undertakes to move data from A to B.

Methods of frame forwarding

Before I go any further, I would like to briefly mention the two methods of forwarding data. They are as follows:

- Cut-through switching
- Store and forward

Cut-through switching

In **cut-through switching**, the switch forwards the data almost immediately. It just needs to know the source and destination MAC addresses. As soon as it has this information, it will forward the data, even if the whole frame hasn't been received by the switch. A simplified visualization of this can be seen in the following diagram:

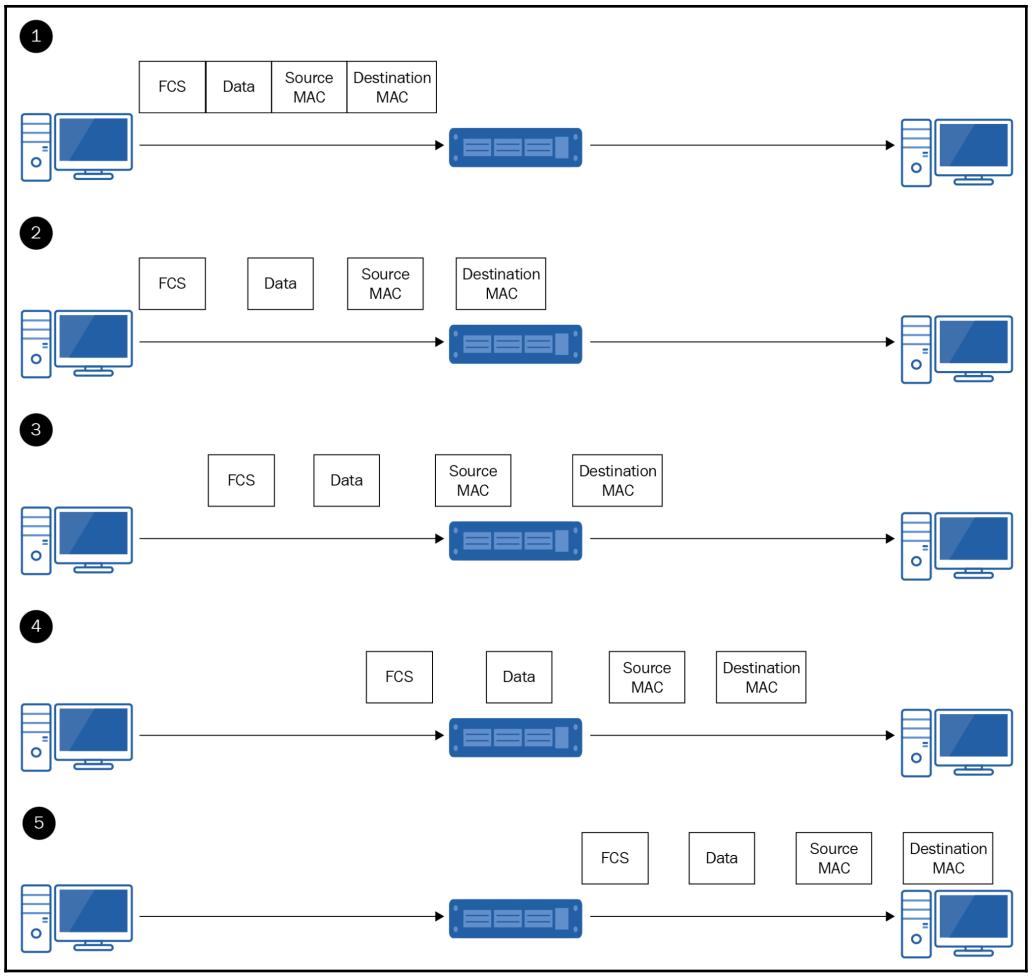

Figure 6.2: Cut-through switching

Let's go through the numbered steps in the preceding diagram so that we can understand this process better:

- In *step 1*, the frame is being sent.
- In *step 2*, the switch receives the destination MAC address.
- In *step 3*, the switch starts to forward the frame, even though it's only received the destination MAC address.
- In *steps 4* and *5*, the switch continues to send the remainder of the frame.

While this is a fast method, it means that frames that containing errors are forwarded. To prevent this, most modern switches will use a process called **store-and-forward** instead of cut-through switching.

Store and forward

In this process, the switch will store the frame data in its memory buffer until the complete frame has been received. Once the frame has been completely received, the switch will perform error checking on the data before forwarding the frame on. Any corrupt frames are discarded. Store and forward also allows data to be prioritized through **Quality of Service** (**QoS**). We can see a simplified visualization of this in the following diagram:

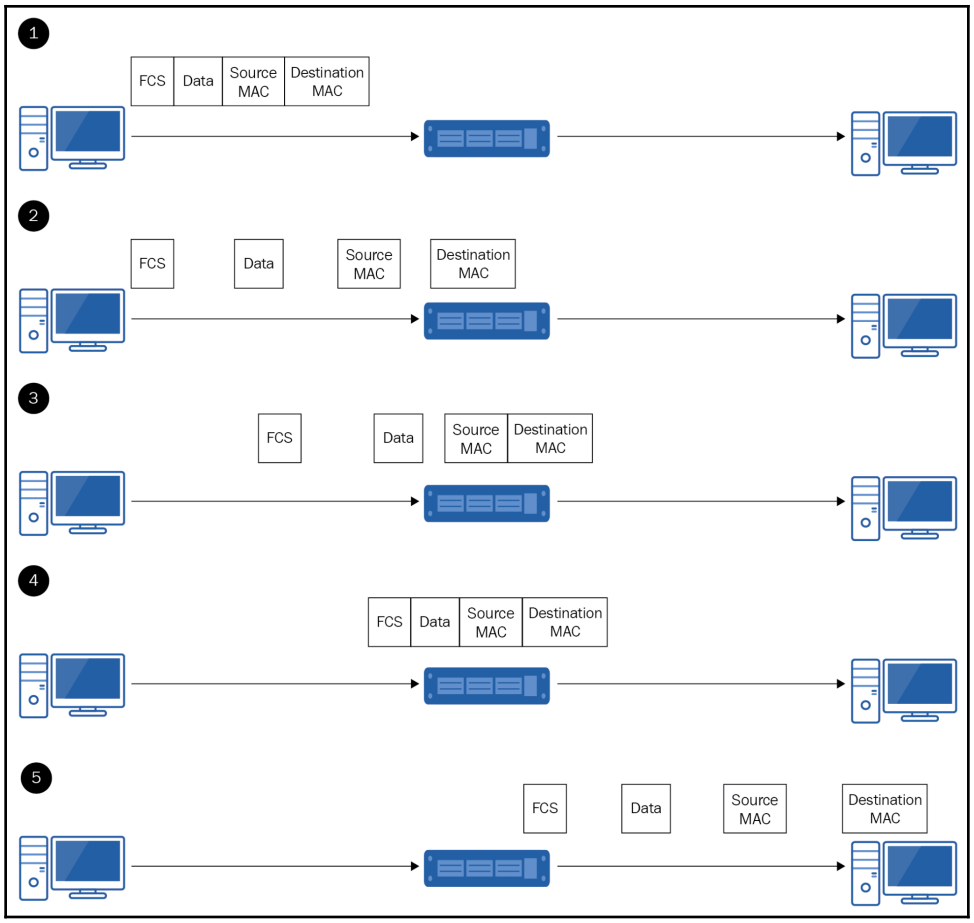

Figure 6.3: Store and forward switching

Let's go through the numbered steps in the preceding diagram so that we can understand this process better:

- In *step 1*, the frame is being sent.
- In *step 2*, the switch receives the destination MAC address.
- In *step 3*, the switch holds on to (stores) the frame.
- In *step 4*, the switch now has all of the frames it will carry out the error check on.
- In *step 5*, the switch forwards the frame(s) on if they passed the error check.

Regardless of which of these two methods you use, the process of frame forwarding will be the same. We'll learn more about this in the next section.

Frame forwarding process

I will summarize this process here and then walk through an example. When a switch receives data from a device, it follows this process:

1. It checks to see if the source MAC address is in its MAC table (also known as the **Content Addressable Memory** (**CAM**) table):
    - If it isn't, the switch updates the table by specifying which interface the sender's MAC address is on
2. Then, it looks to see if there is an entry for the destination MAC address:
    - If there is, the frame is forwarded internally to the interface that's listed, which and then transmits it to the destination device.
    - If there is no entry for the destination MAC address, the switch will flood or broadcast the frame from all its interfaces, except the interface the frame came in on.
3. Hopefully, the intended recipient will receive the frame and respond.
4. The switch will repeat this process for the response and should, of course, know the interface that the destination (the original source device) is on.

You may be wondering what happens if the device isn't directly connected to the switch that has received the frame – perhaps the frame has been forwarded from another switch. In this situation, the switch receiving the data doesn't really care. It just cares about what interface the data has been received on. The MAC table will purge entries on a regular basis if they haven't been used for a period of time.

In this walkthrough, we are going to use the topology shown in the following diagram. As an aside, can you recall which topology is shown here?

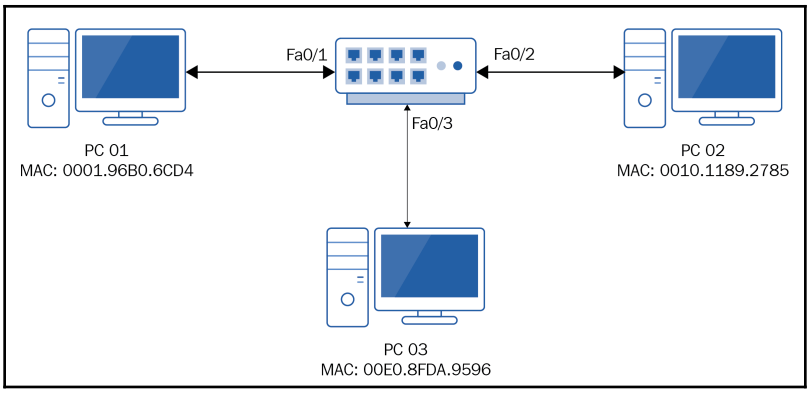

Figure 6.4: Topology walkthrough

In this walkthrough, **PC 01** sends some data to **PC 02**, who will respond:

1. **PC 01** sends data to the switch.
2. The switch receives the data and checks its MAC/CAM table to see if there is an entry for **PC 01**'s MAC address. There is no entry there at the moment:

```
Switch>show mac-address-table
          Mac Address Table
-------------------------------------------

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

Switch>
```

Figure 6.5: No entry for MAC address

3. Since there is no entry, the switch updates the table with the relevant information:

```
Switch>show mac-address-table
          Mac Address Table
-------------------------------------------

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

  10    0001.96b0.6cd4    DYNAMIC     Fa0/1
Switch>
```

Figure 6.6: Updated MAC/CAM table

4. Then, the switch looks for the destination MAC address (**PC 02**) in the table.
5. Since there is no entry, the frame is forwarded out of all the interfaces except for the one it came in on (`Fa0/1`).
6. Both **PC 02** and **PC 03** will receive the frame and check the destination MAC address contained within it. **PC 03** will ignore the frame as it's not intended for it.
7. **PC 02** will process the frame accordingly and, in this case, send a response.
8. On receipt of **PC 02**'s response at the switch, the latter will check the MAC table to see if there is an entry for **PC 02**'s MAC address.
9. Since there is no entry, it will update the MAC table with the relevant information:

```
Switch>show mac-address-table
          Mac Address Table
-------------------------------------------

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

  10    0001.96b0.6cd4    DYNAMIC     Fa0/1
  10    0010.1189.2785    DYNAMIC     Fa0/2
Switch>
```

Figure 6.7: Updated MAC/CAM table

10. Then, it will look at the destination MAC address (**PC 01**'s) and check whether it's in the MAC table.
11. On this occasion, there is an entry for it, so the switch forwards the frame through `Fa0/1`. There is no need to flood it out of all the interfaces.

You may have noticed that, in the output shown in the walkthrough screenshots, there is a column for `Type`, and each of the entries shows `DYNAMIC`. This indicates that the MAC address/interface mapping has been learned automatically. An administrator may configure static mapping, but this is unlikely.

Here, the switch flooded the frame out of every interface as it didn't know which interface the unicast MAC address was on. It will do the (same for any frame addresses) to broadcast MAC addresses or multicast MAC addresses. When you only have one switch in your network, this will not cause any issues. However, if you have switches that are connected, there are some other things you need to take into considerations. In the following subsection, I will explain one of these main issues and briefly cover a means of preventing it from happening.

Spanning Tree Protocol

I'd like to start this subsection by talking about the **Spanning Tree Protocol** (**STP**). Look at the simple topology shown in the following diagram. In this topology, I have two switches connected to each other with two links:

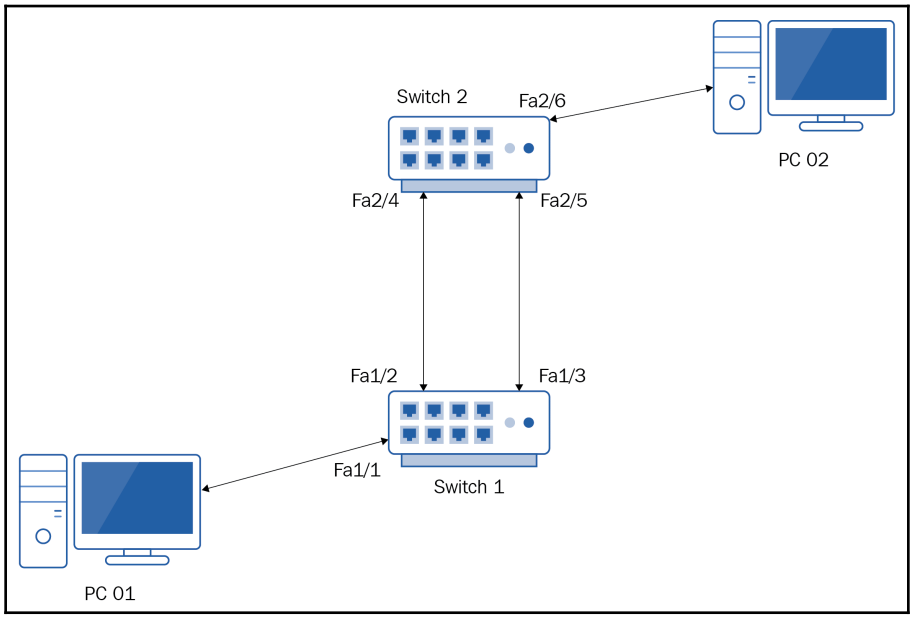

Figure 6.8: Switches connected with redundancy

This is a common practice and offers redundancy. Should one of the links fail, the other is there to allow communication to take place between the two switches. I'd like you to think for a moment about what happens if **PC 01** sends some data to **Switch 1** and **Switch 1** either doesn't know the outbound interface or it's addressed to a broadcast/multicast MAC address. What interface will **Switch 1** send it out of? What will **Switch 2** do when it receives this data and doesn't know the outbound interface, or it's addressed to a broadcast/multicast MAC address? Can you start to visualize what issue might arise?

Let me walk through what's going to happen in this scenario. For simplicity, I have labeled the interfaces so that **Fa1/x** is an interface on **Switch 1** and **Fa2/x** is an interface on **Switch 2**:

1. **Switch 1** receives the data from **PC 01** on **Fa1/1**.
2. It doesn't know the outbound interface, so it sends it out of both **Fa1/2** and **Fa1/3**.

3. **Switch 2** receives the data from **Fa1/2** on its interface: **Fa2/4**. Then, **Switch 2** sends it out of all the interfaces except the one it was received on: **Fa2/5** and **Fa2/6**.
4. Switch 2 also receives the data from **Fa1/3** on its interface, that is, **Fa2/5**, and sends it out of all the interfaces except the one it was received on: **Fa2/4** and **Fa2/6**. **Switch 2** has no way of knowing that the same frame has been received twice.
5. **Switch 1** will receive the data from interface **Fa2/4** on **Fa1/2** and has no way of knowing whether it was the data it forwarded on previously. It sends the data out of all the interfaces except **Fa1/2**.
6. **Switch 1** also receives the data from **Fa2/5** on its interface, **Fa1/3**, and sends it out of all the interfaces except **Fa1/3**.
7. **Switch 2** receives the two lots of data from **Switch 1** and this *looping* continues, with the data going back and forth between the two switches.

This was just one piece of data looping around – imagine what would happen if we had more and more data impacted by this. In the end, this will result in the network failing totally. In this situation, you will have created a **broadcast storm**, and usually the only way of stopping this is by powering off all the switches or unplugging the cables and then taking steps to prevent this from happening again through configuration. On top of this, the end devices will have to process the same piece of data over and over again, possibly resulting in a denial of service. We can't disable broadcasts or multicasts as they are a required function of all networks, so we have to use a protocol such as STP to prevent this.

Fortunately, for the exam, you don't need to know how the STP operates – you only need to know its purpose and the very basics of what it does. You can breathe a sigh of relief knowing that both of these are relatively simple to understand.

The purpose of STP is to prevent looping between switches when redundant links are implemented. To do this, an election takes place between the switches to decide which switch will be classed as the *root bridge*. Once identified, the switches identify which of their interfaces are the closest to the route bridge and dub them **root ports** (**RP**). These ports will always be available. The other ports are referred to as **designated ports** (**DPs**) and **non-designated ports** (**NDPs**). Non-designated ports are prevented from sending data unless something fails on the network that requires them to assume the role of sending data:

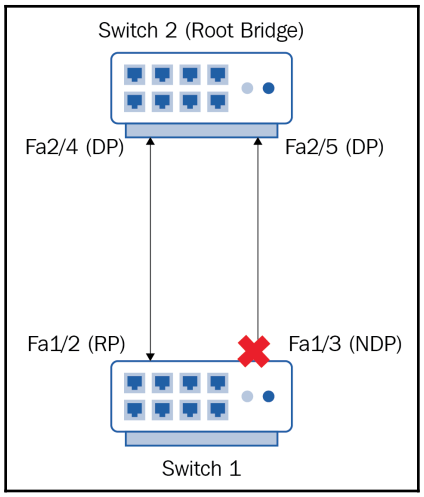

Figure 6.9: STP in operation

In the preceding diagram, we can see that **Switch 2** has been elected as the root bridge, while **Switch 1** has identified **Fa1/2** as the RP. This means it has identified **Fa1/3** as the non-designated port, so no traffic will flow between the two switches through that interface. Behind the scenes, there is a lot of management data going back and forth between the two switches; should the link between **Fa1/2** and **Fa2/4** fail, then this will be identified and **Fa1/3** will be promoted to RP and be allowed to send data.

Exam tip: STP can be used to refer to Spanning Tree Protocol and Shielded Twisted Pair, which is a type of cable. Remember this for the exam. People have been known not to choose it as an answer because they forgot it has two meanings.

Now that we have looked at how a switch functions, let's discuss some of the main characteristics of switches.

Understanding switch characteristics

While all the switches provide the basic functionality of forwarding frames, not all of them will be equal. In this section, I will provide an overview of some of the main characteristics to be aware of when building a switched network.

Managed and unmanaged switches

A switch can be classed as either managed or unmanaged. In a business environment, you are more likely to see a managed switch than an unmanaged switch due to the extra functionality the former offers. However, you may see small unmanaged switches being used in offices where there's a shortage of wall ports.

An unmanaged switch is a network device that ships with a preloaded configuration that cannot be changed and is used purely to allow endpoint devices to communicate with each other. Physically, the unmanaged switch will usually be a small box with a small number of ports. The following picture is an example of a NETGEAR unmanaged switch with five ports. Notice the labeling above the port with the red cable. We can see that this switch supports speeds of 10 Mbps and 100 Mbps. The fact that both LEDs below the labels are illuminated indicates that this connection is running at 100 Mbps:

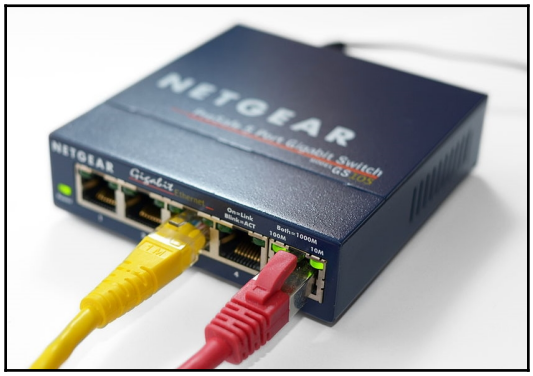

Figure 6.10: Unmanaged switch

The preceding picture can be found at `https://de.wikipedia.org/wiki/Switch_(Netzwerktechnik)#/media/Datei:Netgear_Gigabit_Switch_5-port.jpg`. Figure by Simon A. Eugster. Licensed under Creative Commons CC BY-SA 3.0: `https://creativecommons.org/licenses/by-sa/3.0/`.

In contrast to an unmanaged switch, a managed switch will allow you to configure VLANs, port speeds, security, duplex settings, and so on. To manage the switch, you would either physically connect to the switch's console port or remotely manage the switch using some form of terminal emulator. To be able to connect to the switch remotely, the switch needs to be configured with an IP address. This can only be initially configured through a console port, similar to the one shown in the following picture:

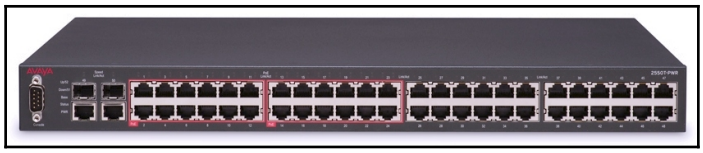

Figure 6.8: Managed switch with a console port

The preceding picture can be found at `https://upload.wikimedia.org/wikipedia/commons/b/b9/2550T-PWR-Front.jpg`. Licensed under Creative Commons CC BY-SA 3.0.

A laptop or similar device is physically connected to the console port and a connection is made from the operating system using a terminal emulator, such as the popular application PuTTY. This is shown in the following screenshot:

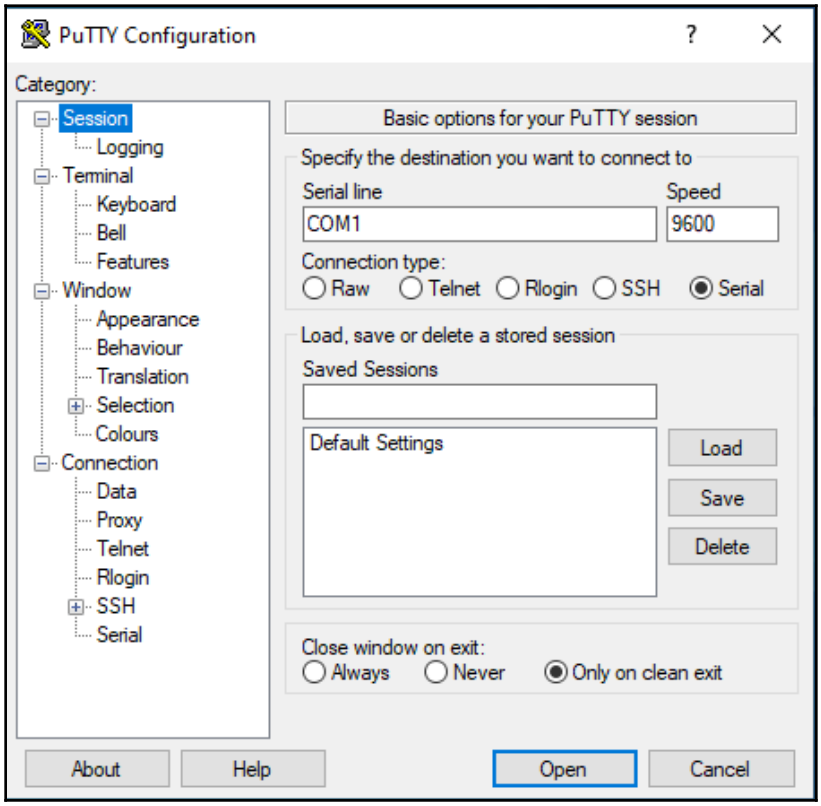

Figure 6.9: PuTTY emulator

As well as selecting **Serial** on the general screen, you also need to ensure that you configure advanced settings, as shown in the following screenshot:

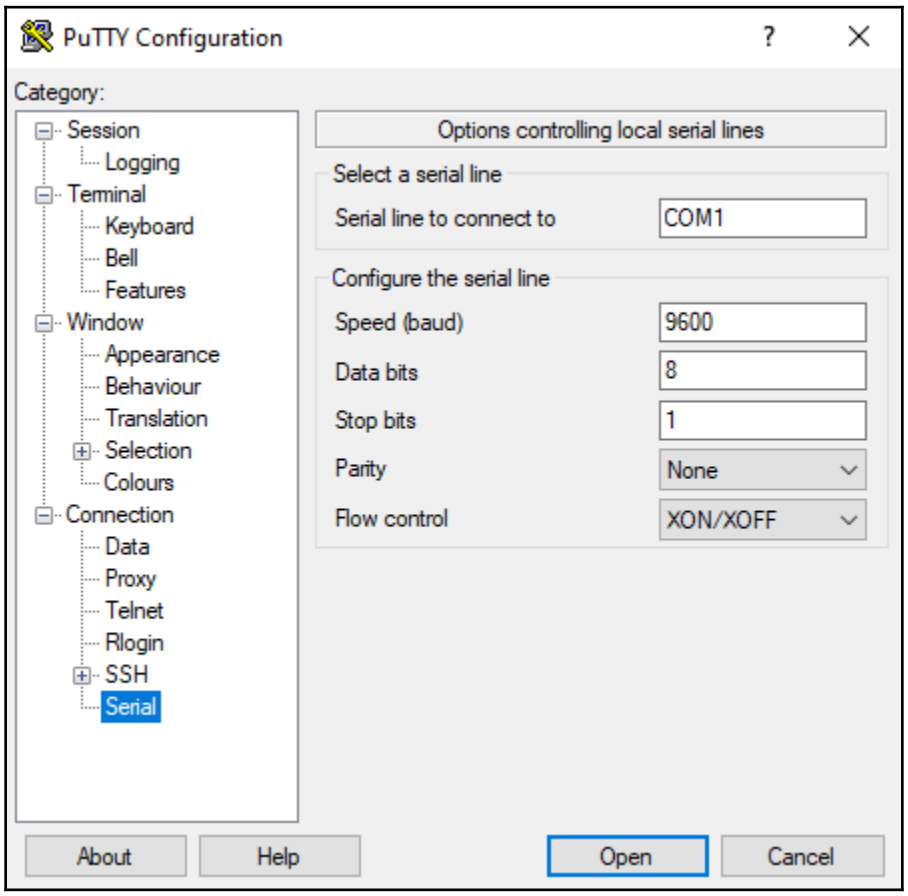

Figure 6.10: PuTTY's additional settings

It is imperative that the settings you select here match those of the device you are trying to configure. If they don't match, you will fail to connect or, if you can connect, you will likely experience issues.

Connectivity to the network

When using an unmanaged switch, the switch itself will be plugged into an available wall port and the devices will be connected to the switch.

In an enterprise environment, the end devices will be connected to an available wall port. The wall port will be connected to the back of a patch panel, into what is known as a **punch-down block**. The front of the patch panel will be connected to the switch. The following diagram shows how this is laid out. The purpose of using a patch panel is to ease cable management:

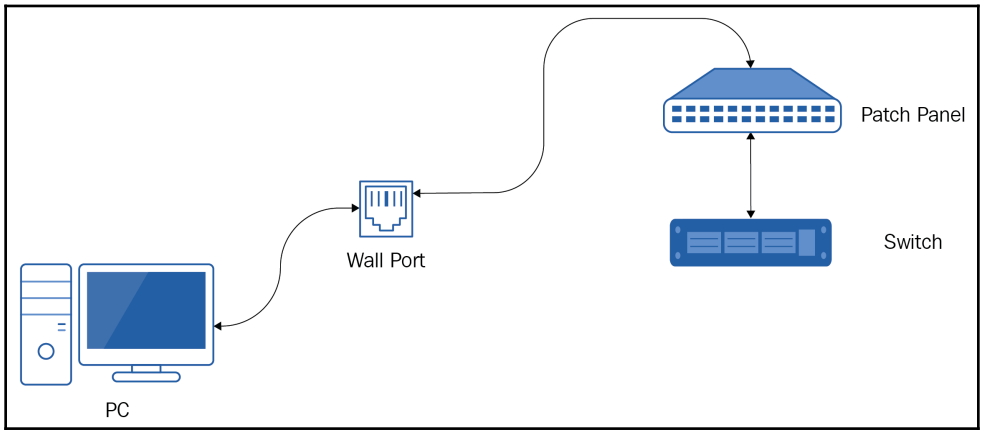

Figure 6.11: Enterprise connectivity

The wall port is wired differently compared to an RJ-45 connector. The connections you can see at the front of the wall port in the following picture are wired directly to the prongs at the rear. The wires from a cable are then connected to these prongs and the cable will be run back to a patch panel:

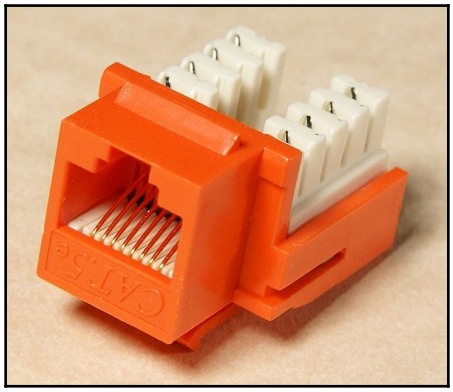

Figure 6.12: Wall port

The preceding image can be found at `https://en.wikipedia.org/wiki/Keystone_module#/media/File:Keystone_module_CAT5_orange.jpg`.

The following image shows the back of a patch panel. To connect the cables, you would use a punch-down tool. If you look at the bottom right of this image, you can see that each grouping is broken down into four (there are six sets of four colors: green, orange, brown, and blue). These are color-coded to match the colored pairs of cables being connected:

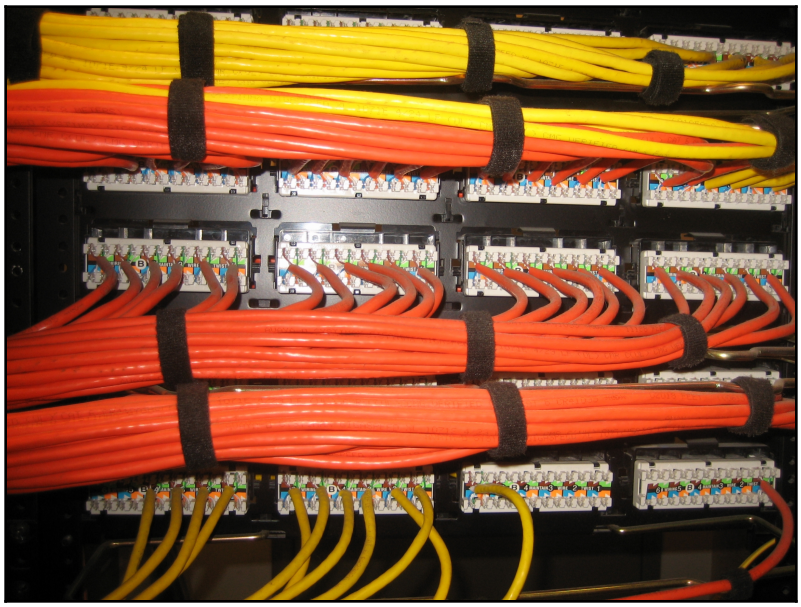

Figure 6.13: Rear of the patch panel

The preceding image can be found at `https://www.flickr.com/photos/dbarsky/2261404319`. Figure by Dmitry Barsky. Licensed under Creative Commons CC BY 2.0.

In the following image, you can see how the cables are connected to the front. The patch panels are at the top and the two silver-fronted boxes are the switches:

Figure 6.14: Front of the patch panel

The preceding image can be found at `https://upload.wikimedia.org/wikipedia/commons/b/bc/19-inch_rackmount_Ethernet_switches_and_patch_panels.jpg`.

The diagram in this section shows all of the ports that are available. We will discuss these now.

Ports

Once you have decided on whether you want a managed or unmanaged switch, you will likely focus on the number of interfaces or ports the switch can support and the type of port they are. Usually, the options will be 12, 24, or 48 ports. The switch we saw in *Figure 6.8* shows a 48-port switch, where the uplink ports are the four ports to the left of the main interfaces.

In `Chapter 8`, *Media Types – Connecting Everything Together*, we will discuss the various types of cable in detail. For now, though, your port choice will most likely be down to whether you want the switch to support twisted pair cables, fiber optic cables, or both.

Besides the port type, you will also need to focus on the speeds the interfaces can support. Switches can support various speeds, such as Fast Ethernet (10/100 Mbps), Gigabit Ethernet (10/100/1,000 Mbps), Ten Gigabit (10/100/1,000/10,000 Mbps), and even 40/100 Gbps speeds. Obviously, the faster the switch, the more expensive it will be. It's unlikely that the ports that are connected to the end user's computers will need the same connection speed as a port connected to a media streaming server, for example. Another form of speed that you may hear mentioned is backplane speed or fabric bandwidth. These refer to the internal speed of data being moved from one switch port to another.

Since ports can support multiple speeds and duplex settings, they may need to be configured accordingly. Most switches can auto-negotiate these, but it may be more appropriate to configure these manually to avoid any issues that may be caused by failure to successfully negotiate. As a fail-safe, a lot of switches will fall back to the slowest speed and half-duplex. Obviously, this slows down the network, and we would prefer to have the fastest possible speed and, of course, full-duplex to allow for simultaneous two-way communication.

Another characteristic that should be taken into consideration when purchasing an enterprise-level switch is modularity. Modern enterprise switches, such as Cisco's Catalyst range, provide us with the ability to expand the capabilities of a switch through the use of modules that you physically add to the switch. These modules can include support for different cable types, different speeds, and wireless functionality. The use of modules allows an organization to buy a switch at a much lower cost when first starting, but upgrade with minimal disruption as the company grows.

Activity: We have just talked about the modularity of some switches. It is worthwhile having an awareness of what modules are available to organizations so that you understand the device's extensibility. Conduct an online search for Cisco Ethernet switching network modules and investigate the various modules that are available.

In `Chapter 2`, *Understanding Local Area Networks*, we discussed the use of a tiered approach when it comes to using switches with an access layer, distribution layer, and core layer. To provide suitable speeds when sending data between each of the layers, enterprise-level switches usually include faster uplink ports offering speeds of 10-40 Gbps.

Layer 3 switches and VLANs

Besides classifying switches as managed or unmanaged, we can also classify them as being a layer 2 or a layer 3 switch. A layer 2 switch only supports the use of forwarding data based on MAC addresses, whereas a layer 3 switch also offers additional support in terms of being able to recognize the IP address of an interface, and also provides traffic routing using an IP address. This is essential when routing traffic between VLANs. In `Chapter 2`, *Understanding Local Area Networks*, we learned how VLANs segment a network into subnets and that no direct communication can take place between them. For them to be able to communicate with each other, you need to implement a router or a layer 3 switch.

When creating a VLAN, you need to configure the relevant interfaces on the switch so that they're a part of that VLAN. By default, all the interfaces on a switch are part of a single default VLAN. That way, the interfaces can communicate with each other. If you want to allocate an interface to a different VLAN, you simply reconfigure it as part of the new VLAN. The following screenshot shows the VLAN configuration of a Cisco switch, with all the interfaces on the default VLAN:

```
Switch>show 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
                                                Gig0/1, Gig0/2
```

Figure 6.15: All the interfaces on the default VLAN

The following screenshot shows a Cisco switch that has been configured so it has multiple VLANs:

```
Switch>show vlan

VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
1    default                          active    Fa0/3, 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, Gig0/1
                                                Gig0/2
10   VLAN10                           active    Fa0/1, Fa0/2
20   VLAN20                           active    Fa0/4, Fa0/5, Fa0/6
```

Figure 6.16: Additional VLANs configured

Each interface that's connected to an end device will need to be configured as an access port and either left on the default VLAN or allocated to any VLAN that has been configured. An access port will carry traffic for one VLAN only.

Let's revisit the hospital scenario from `Chapter 2`, *Understanding Local Area Networks*, as shown in the following diagram:

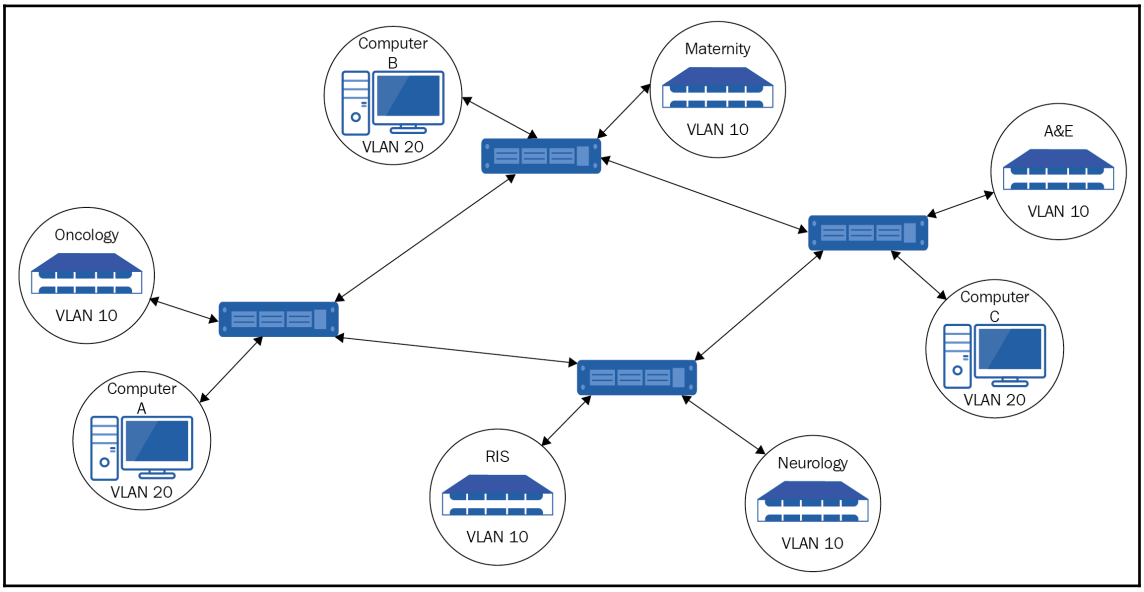

Figure 6.17: Example VLAN configuration

Looking at **Computer A**, **Computer B**, and **Computer C**, the interfaces that each of these is connected to will be classed as access ports. But what about the connections between the switches? Each of those connections will need to carry data from any of the VLANs. Therefore, interfaces connecting a switch to another switch will be configured as *trunk ports*. A trunk port carries data for multiple VLANs and has to be configured to allow the VLANs to transmit data over it.