Router and switch interfaces

This chapter covers

- How to configure interfaces and verify their status
- Interface speed and duplex settings
- Using auto negotiation to automatically determine an interface's speed and duplex
- Errors that can occur when sending and receiving messages over a network

In chapter 7, we looked at how to configure IP addresses on and enable router interfaces. In this chapter, we will dig deeper into the topic of interfaces and how they operate. Whereas the previous chapter covered how to configure IP addresses on interfaces (a Layer 3 concept), this chapter will focus primarily on Layer 1 concepts, such as how to configure the speed at which an interface can send and receive data. Specifically, we will cover the following CCNA exam topics:

- 1.3.b Connections (Ethernet shared media and point-to-point)
- 1.4 Identify interface and cable issues (collisions, errors, mismatch duplex, and/or speed)

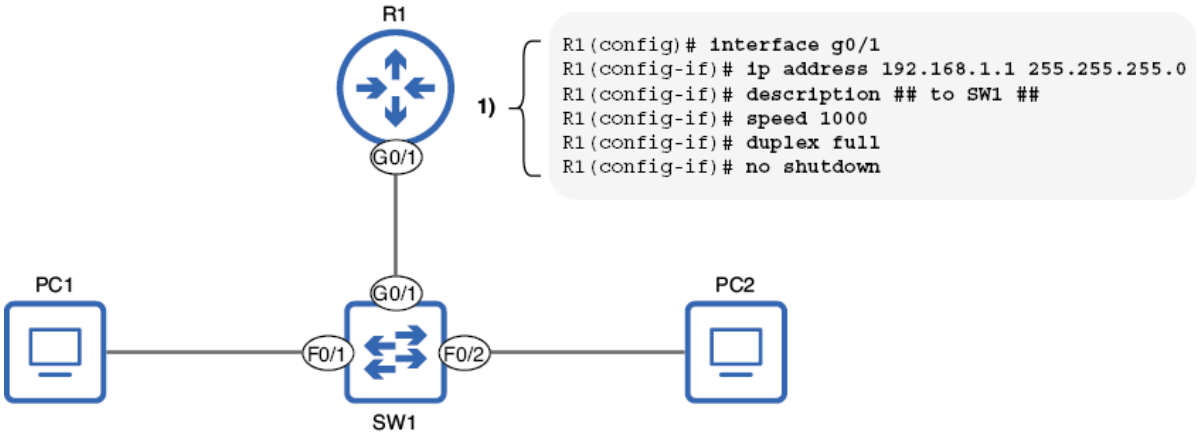
In previous chapters, I have used the terms *port* and *interface*. The exact definitions of these terms depend on who you ask; some say a port is a Layer 2 entity that forwards frames within a LAN (switches have ports), and an interface is a Layer 3 entity that forwards packets between LANs (routers have interfaces). Another definition is that a port is the physical connector on a device that you plug a cable into, and an interface is the representation of that port within the software (hence, most Cisco IOS commands use the term **interface** rather than **port**).

In reality, these terms are often used interchangeably - even Cisco's documentation isn't consistent regarding these terms. In this book I will generally use the term "port" to refer to a physical connector on a device and "interface" when talking about configurations, except in situations where one term is generally accepted as the standard over the other (in which case I will point that out).

8.1 Configuring interfaces

In this section we will look at how to configure three aspects of an interface: description, speed, and duplex. Figure 8.1 shows how to configure these settings on Cisco routers and switches running Cisco IOS.

Figure 8.1 Interface configurations on R1 and SW1: (1) configuring R1's G0/1 interface with an IP address, description, and manual speed and duplex settings; (2) configuring SW1's G0/1 interface with a description, and manual speed and duplex settings; (3) configuring SW1's connections to end-user devices (PCs) using autonegotiation; (4) disabling SW1's unused interfaces.




```
1)  R1(config)# interface g0/1
    R1(config-if)# ip address 192.168.1.1 255.255.255.0
    R1(config-if)# description ## to SW1 ##
    R1(config-if)# speed 1000
    R1(config-if)# duplex full
    R1(config-if)# no shutdown

2)  SW1(config)# interface g0/1
    SW1(config-if)# description ## to R1 ##
    SW1(config-if)# speed 1000
    SW1(config-if)# duplex full
3)  SW1(config-if)# interface range f0/1-2
    SW1(config-if)# description ## end users ##
    SW1(config-if)# speed auto
    SW1(config-if)# duplex auto
4)  SW1(config-if)# interface range f0/3-8, g0/2
    SW1(config-if)# description ## not in use ##
    SW1(config-if)# shutdown
```

Before we examine the above configurations in detail, there are a couple things worth mentioning that illustrate some differences between routers and switches. First, notice that I configured an IP address on R1's G0/1 interface, but not on SW1's interfaces; this is because switch interfaces don't need IP addresses to perform their role of forwarding frames within a LAN. Switches are not layer 3-aware; they only use layer-2 information (MAC addresses) to decide how to forward frames.

The second point is that I used no shutdown on R1's G0/1 interface to enable it, but not on SW1's G0/1, F0/1, or F0/2 interfaces. That is because, unlike router interfaces (which are disabled by default), switch interfaces are enabled by default.

Exam Tip

In figure 8.1 I used shutdown to disable SW1's unused ports. This is considered a security best practice and may come up in an exam question.

The reason switch interfaces are enabled by default is to allow switches to operate in a *plug-and-play*; this means that, to use the switch, you simply need to connect devices to it - no configuration required. However, although a switch does not require configuration to perform its most basic function of forwarding frames, most enterprise networks will require configurations on switches to use more advanced features

Note

A switch that is designed to be used in a plug-and-play manner is called an *unmanaged switch*. Unmanaged switches are inexpensive and are sometimes used in very small networks. The CCNA focuses on *managed switches*, which allow you to configure more advanced features.

8.1.1 Interface descriptions

An interface description is a simple string of text that you can configure to describe, or name an interface. A common use is to indicate what device is connected to the interface. The command to configure an interface's description is description *description*, where *description* is a string of text such as "connected to R1's G0/1 interface". The below example shows how I configured the descriptions of SW1's F0/1 and F0/2 interfaces. F0/1 and F0/2 are connected to end user devices (PCs), so I configured their descriptions as "## end users ##".

1 SW1# **configure terminal**

2 SW1(config)# **interface range f0/1-2**

3 SW1(config-if)# **description ## end users ##**

copy

Note

The two hash symbols at the beginning and end of the description are not necessary. I use them in my interface descriptions to help them stand out when viewing them in the CLI.

Notice that I used the interface range f0/1-2 command to configure SW1's F0/1 and F0/2 interfaces at the same time; interface range can be a big time saver when configuring multiple interfaces! The command to configure a range of interfaces is interface range *type slot/port-port*. Let me explain each of those arguments in the command:

- type means Ethernet, FastEthernet, GigabitEthernet, etc.
- slot is the first number in the interface name.
- port is the second number in the interface name.

Note

Another example from figure 8.1 is interface range f0/3-8, g0/2. This configures F0/3, F0/4, F0/5, F0/6, F0/7, F0/8, and G0/2. To include interfaces of a different type in the same interface range command, you must separate the interface names with a comma.

Interface descriptions are optional, but I highly recommend configuring them; interface descriptions that are consistently configured and updated as needed make it much easier to identify the purpose of each interface when viewing the configurations at a later date.

To view interface descriptions on a router or switch, you can use the show interfaces description command. The following example shows the output of that command after configuring SW1's interface descriptions (some output is omitted for sake of space). Note that the command also lists the layer-1 status (Status) and layer-2 status (Protocol) of each interface

1 SW1# **show interfaces description**

2 Interface Status Protocol Description

3 Fa0/1 up up ## end users ##

4 Fa0/2 up up ## end users ##

5 Fa0/3 down down ## not in use ##

6 . . .

7 Gi0/1 up up ## to R1 ##

8 Gi0/2 down down ## not in use ##

copy

8.1.2 Interface speed

1 SW1# **show interfaces status**

2 Port Name Status Vlan Duplex Speed Type

3 Fa0/1 ## end users ## connected 1 a-full a-100 10/100BaseTX

4 Fa0/2 ## end users ## connected 1 a-full a-100 10/100BaseTX

5 Fa0/3 ## not in use ## notconnect 1 auto auto 10/100BaseTX

6 . . .

7 Gi0/1 ## to R1 ## connected 1 a-full a-1000 10/100/1000BaseTX

8 Gi0/2 ## not in use ## notconnect 1 auto auto 10/100/1000BaseTX

copy

Note

When writing the syntax of a command, options in curly brackets are a mandatory choice. In the speed command, you must either specify the *speed* value, or auto to enable autonegotiation.

Note

SW1(config)# interface g0/1

SW1(config-if)# speed ?

10 Force 10 Mbps operation

100 Force 100 Mbps operation

1000 Force 1000 Mbps operation auto Enable AUTO speed configuration

SW1(config-if)# speed 1000

SW1(config-if)# do show running-config interface g0/1

. . .

interface GigabitEthernet0/1

description ## to R1 ##

speed 1000

end

Note

1 SW1(config)# **interface g0/1**
2 SW1(config-if)# **speed ?**
3 10 Force 10 Mbps operation
4 100 Force 100 Mbps operation
5 1000 Force 1000 Mbps operation
6 auto Enable AUTO speed configuration
7 SW1(config-if)# **speed 1000**
8 SW1(config-if)# **do show running-config interface g0/1**
9 . . .
10 interface GigabitEthernet0/1
11 description ## to R1 ##
12 speed 1000
13 End

NOTE

SW1(config)# interface range f0/1-2
SW1(config-if)# speed auto
SW1(config-if)# do show running-config interface f0/1
. . .
interface FastEthernet0/1
description ## end users ##
End

1 SW1(config)# **interface range f0/1-2**
2 SW1(config-if)# **speed auto**
3 SW1(config-if)# **do show running-config interface f0/1**
4 . . .
5 interface FastEthernet0/1
6 description ## end users ##
7 End

NOTE

An interface's *duplex* setting refers to whether it is able to send and receive data at the same time or not. There are two types of duplex:

8.1.3 Interface duplex

Full duplex: The interface can send and receive data at the same time.

- **Note**
- The opposite of duplex is *simplex*, which is one-way communication. The communication from a keyboard to a computer is an example of simplex communication; the keyboard sends data to the computer, but the computer does not send data to the keyboard.

Note

An example of full duplex communication is a wired Ethernet LAN using a switch (or switches). Devices connected to a switch are able to send and receive traffic at the same time, which allows much greater performance compared to half duplex. However, devices connected to a wired LAN weren't always able to operate in full duplex. Before switches, devices called *hubs* were used to connect devices together in a LAN, and devices connected to a hub had to operate in half-duplex mode (these days, hubs are almost never used).

Ethernet hubs

To understand duplex, let's examine one of the precursors to the Ethernet switch: the Ethernet hub. The basic role of a hub is the same as a switch: to connect hosts together in a LAN. Switches use layer-2 information (MAC addresses) to forward frames to the appropriate destination (or flood them as necessary). Hubs, on the other hand, aren't layer 2-aware; when bits of data are received on one port, they simply repeat those bits out of all other ports. This means that all devices in the LAN receive every frame sent in the LAN; each device then examines the destination MAC address of the frame to determine if it should keep or discard the frame. Hubs are considered layer-1 devices; they receive and repeat electrical signals, but don't examine those signals to make forwarding decisions.

Ethernet hubs

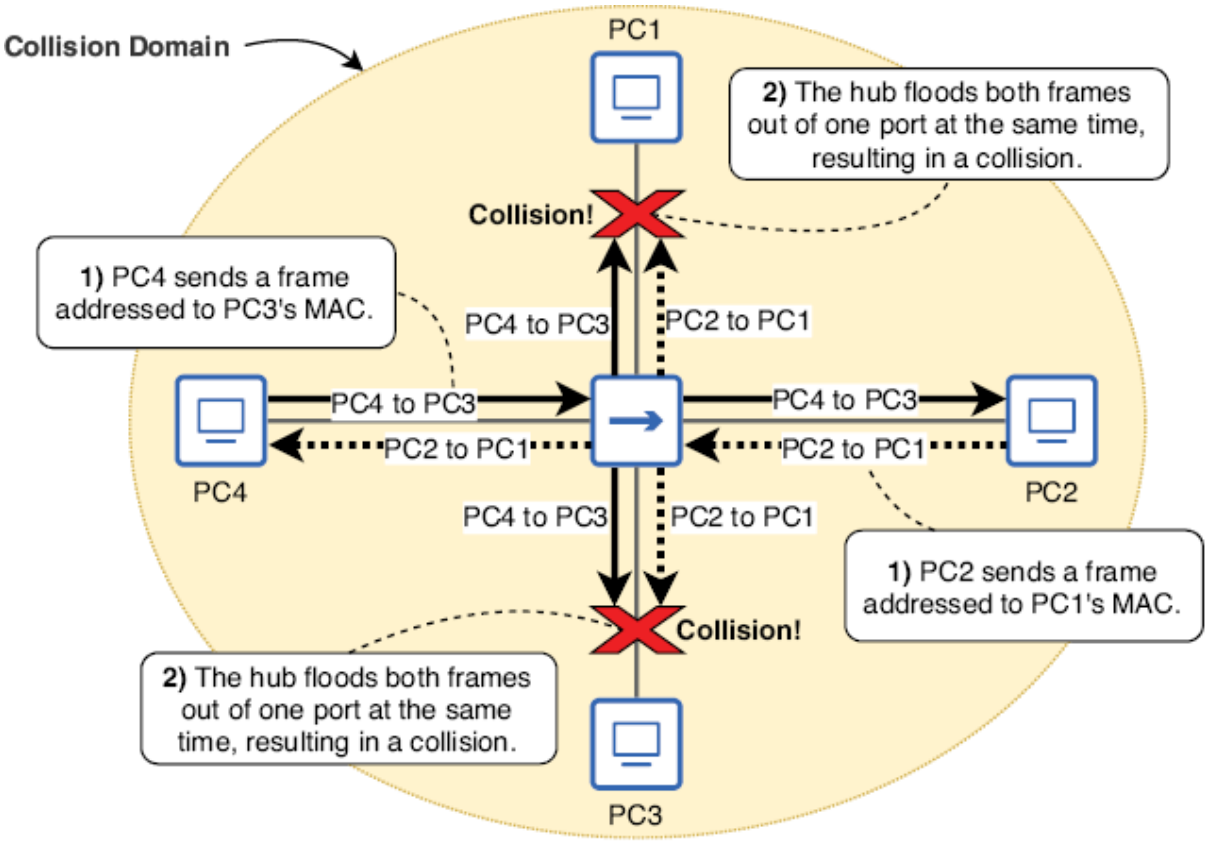
Collision domains

A *collision* occurs when two messages are sent simultaneously over a shared medium and then collide, resulting in an incoherent signal - like if two people talk at the same time,

making you unable to understand either of them. A *collision domain* is a network segment in which simultaneous transmission will result in collisions. As mentioned previously, all hosts connected to a hub are in the same collision domain; this means that only one can transmit at a time. While one host is transmitting, the others can only receive data - they have to wait their turn to transmit. Figure 8.2 shows what happens when two hosts connected to a hub attempt to transmit at the same time.

Unlike hosts connected to a hub, each host connected to a switch is in its own collision domain; switches are able to store frames in memory and forward (or flood) them one after the other, avoiding collisions. This means that hosts connected to a switch can operate in full-duplex mode; all devices in the LAN can send and receive traffic at the same time, with no worry of messages colliding

Figure 8.2 Four PCs are connected to a hub, and two attempt to transmit at the same time, resulting in collisions. All four PCs are in the same collision domain. (1) PC2 sends a frame addressed to PC1's MAC address, and PC4 sends a frame addressed to PC3's MAC address. (2) The hub attempts to flood both frames at the same time, resulting in collisions. Neither PC1 nor PC3 receive their respective frames intact.



Exam topic 1.3.b states: “Connections (Ethernet shared media and point-to-point)”. All devices connected to a hub are connected to a *shared medium*; each device has to share the medium and wait its turn to transmit. Connections to a switch are considered Ethernet *point-to-point* connections - connections between only two devices: the switch and its connected device. Devices connected to a switch do not have to share the medium - they do not have to wait their turn to transmit.

Figure 8.3 Four PCs are connected to a switch, each in its own collision domain. 1) PC2 and PC4 each send a broadcast frame at the same time. 2) The switch floods the frames, but it does not flood PC2’s frame to PC1 or PC3; it buffers the frame in memory. 3) The switch floods PC2’s frame to PC1 and PC3 after it has finished flooding PC4’s frame.

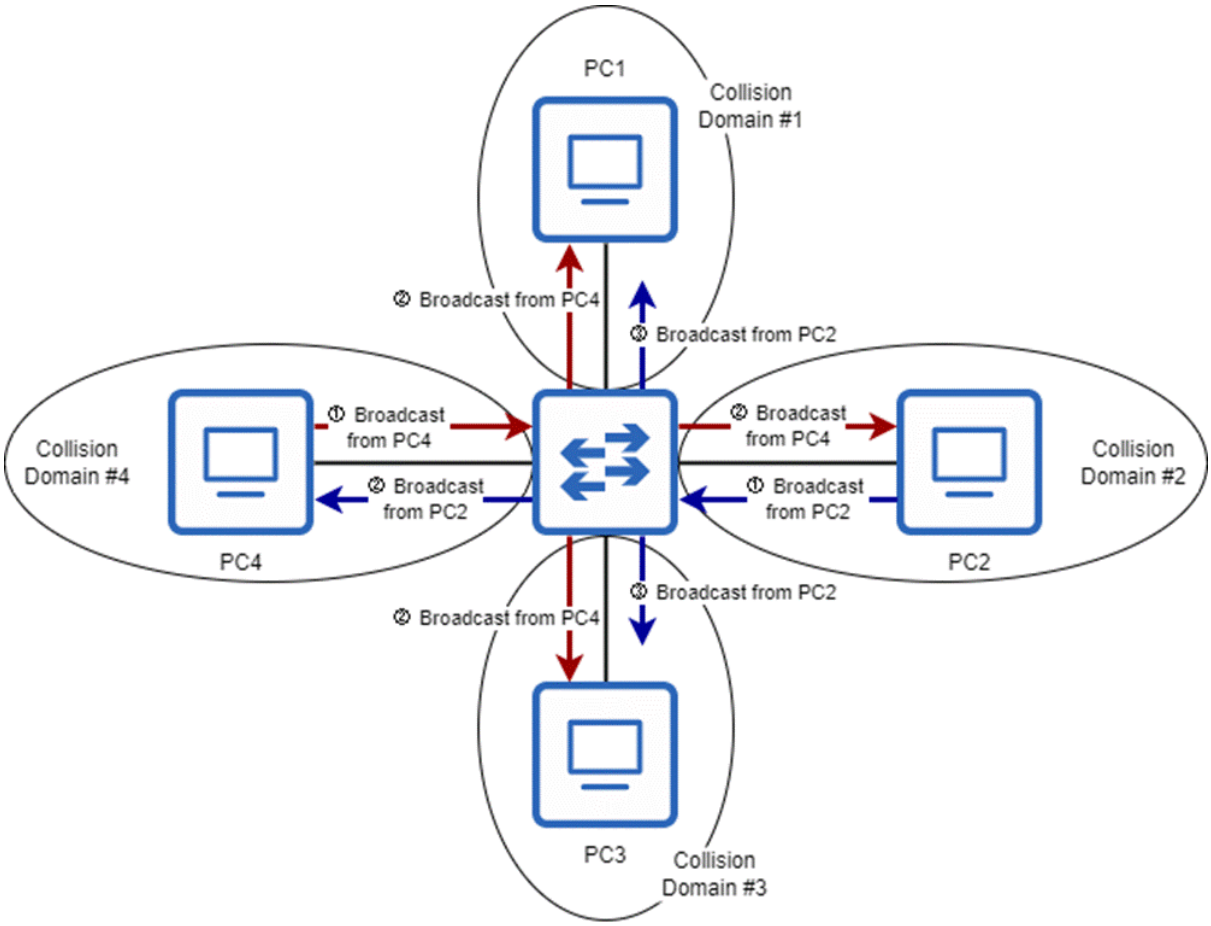
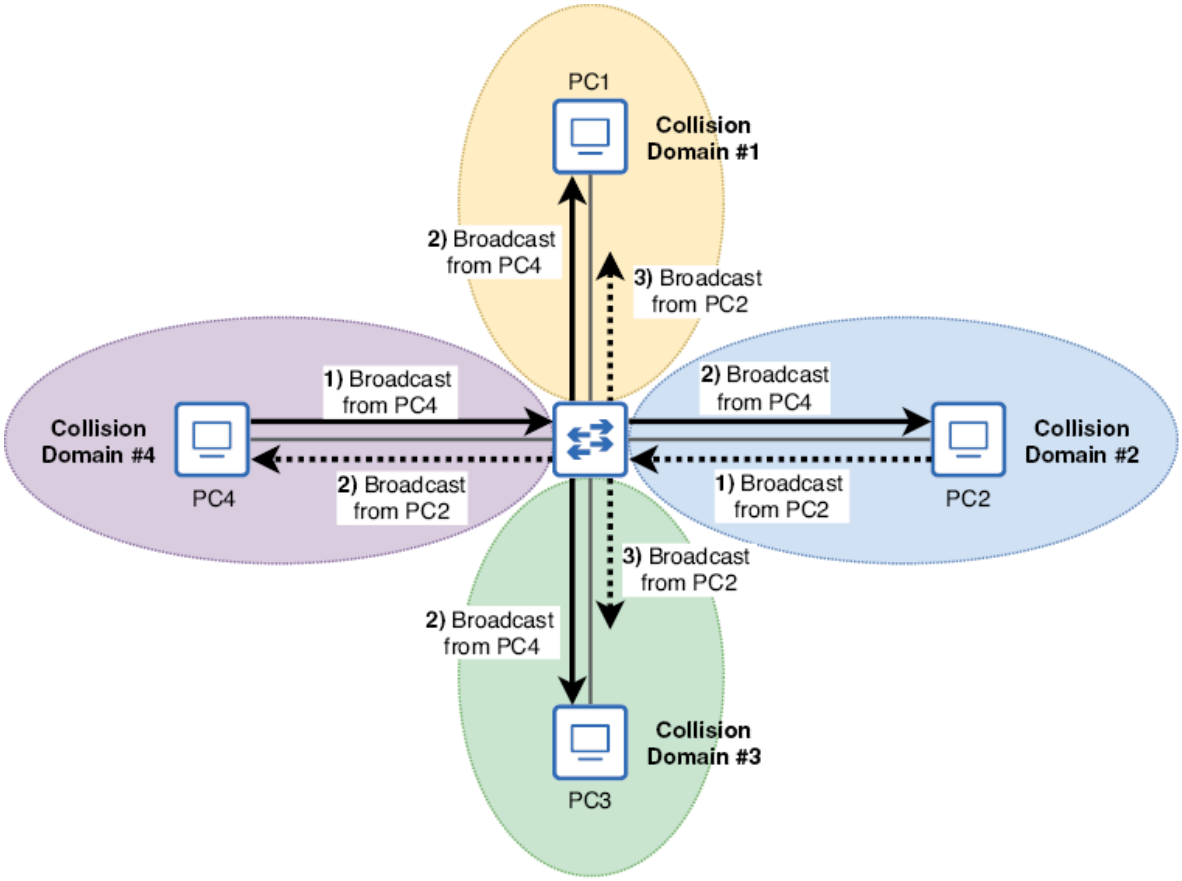


Figure 8.3 Four PCs are connected to a switch, each in its own collision domain. (1) PC2 and PC4 each send a broadcast frame at the same time. (2) The switch floods the frames, but it does not flood PC2’s frame to PC1 or PC3; it buffers the frame in memory. (3) The switch floods PC2’s frame to PC1 and PC3 after it has finished flooding PC4’s frame.



Carrier-sense multiple access with collision detection

Multiple Access means that a shared medium is used (accessed by multiple devices).

- *Collision Detection* means that, if a collision occurs, devices connected to the medium will detect it (and send a signal to notify other devices of the collision).
- CSMA/CD helps devices connected to a hub avoid collisions, but also deal with collisions when they inevitably happen; interfaces operating in half-duplex mode must use CSMA/CD. The CSMA/CD process is as follows:
- Before sending a frame, devices wait until they detect that other devices are not sending.

When a collision occurs, devices that detect the collision will send a jamming signal to inform the other devices of the collision.

1. Each device then waits a random period of time before sending frames again.
2. The process repeats.

Exam Tip

Hubs are rarely used in modern networks, having been almost entirely replaced by switches. However, collisions, collision domains, and CSMA/CD are foundational networking concepts that may appear on the CCNA exam.

Exam Tip

Like an interface's speed, its duplex can also be manually configured or automatically determined using autonegotiation. The command to configure an interface's duplex is duplex {auto | full | half}, and the default setting is auto (which uses autonegotiation). Let's configure SW1's interface duplex settings, as in the example in figure 8.1. In the example below, I first confirm the current status with show interfaces status.

Configuring an interface duplex

Note that the current duplex state for SW1's active interfaces (F01, F0/2, and G0/1) is a-full. This means that they are operating in full duplex, and it was decided using autonegotiation. Interfaces which are not active (not connected to another device) are auto, meaning autonegotiation is enabled, but SW1 hasn't decided if those interfaces will operate in half or full duplex (because they aren't connected to another device yet).

```
1  SW1# show interfaces status
2  Port      Name               Status       Vlan  Duplex  Speed Type
3  Fa0/1     ## end users ##    connected    1     a-full  a-100 10/100BaseTX
4  Fa0/2     ## end users ##    connected    1     a-full  a-100 10/100BaseTX
5  Fa0/3     ## not in use ##   notconnect   1     auto    auto 10/100BaseTX
6  . . .
7  Gi0/1     ## to R1 ##        connected    1     a-full  1000 10/100/1000BaseTX
8  Gi0/2     ## not in use ##   notconnect   1     auto    auto 10/100/1000BaseTX
```

In the Speed column of the example above, F0/1 and F0/2 show a-100, meaning autonegotiation was used to decide upon a speed of 100 Mbps. G0/1 simply displays 1000, because I manually configured a speed of 1000 Mbps.

Note

SW1# configure terminal

SW1(config)# interface g0/1

SW1(config-if)# duplex full

```
SW1(config-if)# interface range f0/1-2

SW1(config-if)# duplex auto

SW1(config-if)# do show interfaces status

Port      Name               Status       Vlan     Duplex     Speed  Type

Fa0/1     ## end users ##    connected    1        a-full     a-100  10/100BaseTX

Fa0/2     ## end users ##    connected    1        a-full     a-100  10/100BaseTX

Fa0/3     ## not in use ##   notconnect   1        auto       auto   10/100BaseTX

. . .

Gi0/1     ## to R1 ##        connected    1        full       1000   10/100/1000BaseTX
Gi0/2     ## not in use ##   notconnect   1        auto       auto   10/100/1000BaseTX
```

Although duplex is an important concept to understand, in modern wired networks you can expect all devices to operate in full duplex; there's no reason to use half duplex. However, wireless LANs operate in half duplex, so we will return to the topic of half duplex when we cover wireless LANs in part 11 of this book.

Note

In section 8.1, we covered that autonegotiation can be used to automatically determine the speed and duplex at which an interface operates, without manual configuration. In most cases you can leave autonegotiation enabled without any issues, although some engineers prefer to manually configure speed and duplex for connections between network infrastructure devices (such as between routers and switches). The reason for this is that manually configuring the speed and duplex settings means that there is one less thing to potentially not work properly (autonegotiation) and cause problems. However, manual configuration does include the potential for human error, and it is extremely rare for autonegotiation to malfunction, so this is not a hard-and-fast rule.

8.2 Autonegotiation

In the autonegotiation process, each device advertises its capabilities to its neighbor, and the two agree upon the best operational mode supported by both neighbors. Table 8.1 lists some operational modes in order of priority - greater speeds are prioritized over lesser speeds, and full duplex is prioritized over half duplex (as you would probably expect).

Table 8.1 Operational modes (view table figure)

Priority	Operational mode
1	10 Gbps, full duplex
2	1 Gbps, full duplex
3	100 Mbps, full duplex
4	100 Mbps, half duplex
5	10 Mbps, full duplex
6	10 Mbps, half duplex

Note

Figure 8.4 A router and a switch advertise their speed and duplex capabilities to each other. The best option supported by both R1 and SW1 is 100 Mbps/full duplex, as highlighted in bold. Although R1 G0/1 is capable of 1 Gbps/full duplex, it will operate at 100 Mbps/full duplex to match SW1 F0/1.

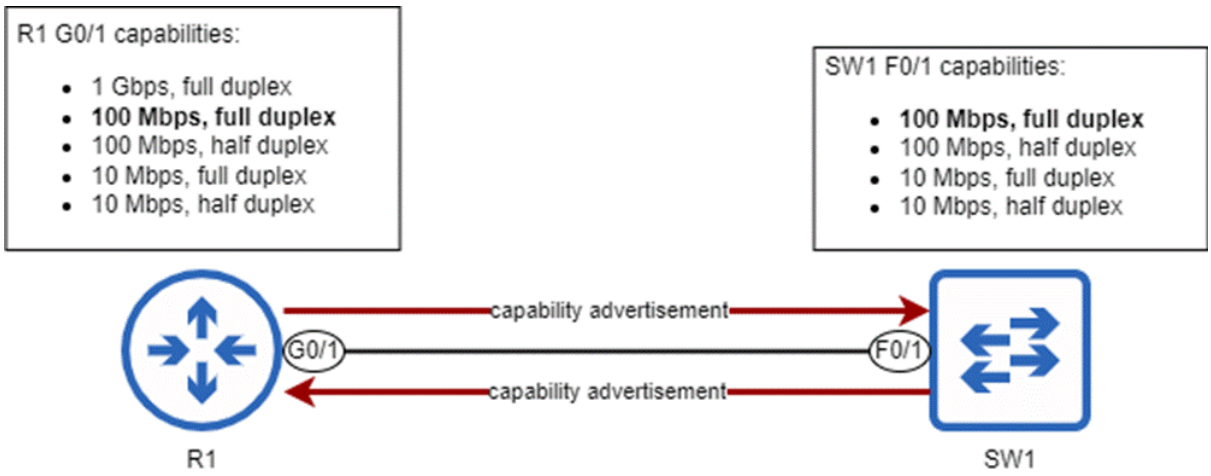
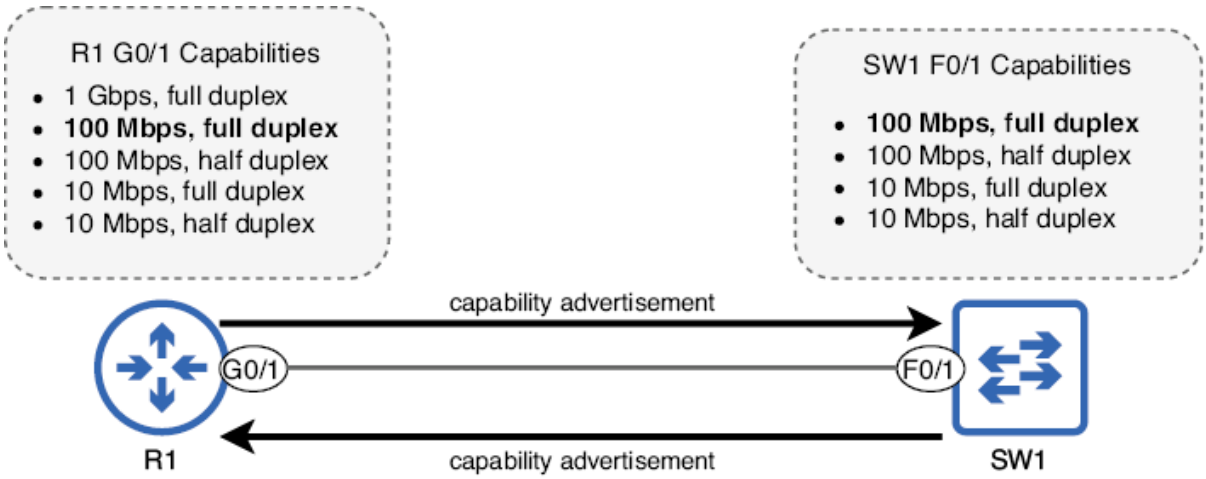


Figure 8.4 demonstrates what happens when both devices are using autonegotiation, but what if speed and duplex are manually configured on one end of the connection, and autonegotiation is used on the other end? In such a situation, the device with autonegotiation enabled will behave as follows:

Figure 8.4 A router and a switch advertise their speed and duplex capabilities to each other. The best option supported by both R1 and SW1 is 100 Mbps/full duplex, as highlighted in bold. Although R1 G0/1 is capable of 1 Gbps/full duplex, it will operate at 100 Mbps/full duplex to match SW1 F0/1.



Duplex: If the speed is 10 Mbps or 100 Mbps, use half duplex. If the speed is 1000 Mbps or greater, use full duplex.

- The above behaviors can result in some problems, and figure 8.5 demonstrates one of those problems. R1 G0/1 is using autonegotiation, but SW1 F0/1's speed and duplex are manually configured. R1 is able to sense the speed that SW1 F0/1 is operating at (100 Mbps) and match its speed. However, following the above rules, R1 G0/1 operates in half duplex, not full duplex. This creates a *duplex mismatch*. And if R1 failed to sense SW1's operating speed, R1 G0/1 would operate at 10 Mbps, resulting in a *speed mismatch*. We will cover speed and duplex mismatches in section 8.3.
- **Figure 8.5 A router and switch connected together, but only the router is using autonegotiation. R1 senses SW1's speed (100 Mbps) and matches its speed to SW1. However, because R1 G0/1 is operating at 100 Mbps, it operates in half duplex. This creates a duplex mismatch between R1 G0/1 (half duplex) and SW1 F0/1 (full duplex).**

Exam Tip

Cisco IOS devices maintain various counters to keep track of errors encountered when sending or receiving messages (such as when messages collide). When a device encounters errors sending or receiving messages on an interface, it increments the relevant counter(s). You can view these counters in the output of show interfaces, as shown in the example below.

8.3 Interface errors

In the above output, I have highlighted some errors that you should be able to identify for the CCNA. Let's take a look at what each error type means:

SW1# show interfaces f0/1

. . .

164850273 packets input, 138587749740 bytes, 0 no buffer

Received 606 broadcasts (0 multicasts)

0 runts, 0 giants, 0 throttles

0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored

0 watchdog, 0 multicast, 0 pause input

0 input packets with dribble condition detected

165209751 packets output, 180164587250 bytes, 0 underruns

0 output errors, 0 collisions, 0 interface resets

0 unknown protocol drops

0 babbles, 0 late collision, 0 deferred

0 lost carrier, 0 no carrier, 0 pause output

0 output buffer failures, 0 output buffers swapped out

Giants are frames received with a payload greater than the interface's MTU (Maximum Transmission Unit), which is typically 1500 bytes. Giants are usually a sign of a misconfiguration; devices in the network are not using consistent MTU values.

- *Input errors* is a counter that includes all errors for received frames.
- *CRC* (Cyclic Redundancy Check) counts frames that failed the FCS (Frame Check Sequence) check in the Ethernet trailer. This could be the result of electromagnetic interference (EMI) causing data corruption.
- *Output errors* is a counter that includes all errors for transmitted frames.
- *Collisions* is a counter for all collisions that happen when the device is transmitting a frame. If the device is connected to a hub, collisions are expected. In a switched LAN, this counter should remain at 0.
- *Late collision* is a counter for collisions that happen after the 64th byte of the frame has been transmitted. This counter is significant because, due to the timing of CSMA/CD, collisions should only occur within the first 64 bytes of a frame's transmission. If a collision occurs after that, it often indicates that one of the devices is not using CSMA/CD to check the medium before transmitting (probably due to a duplex mismatch).
- As indicated in their descriptions, some of the above errors are expected as a result of collisions, and are therefore normal occurrences in LANs using hubs. Others are a sign of a problem in the network, such as a misconfiguration or hardware malfunction. In addition to the above errors, for the CCNA exam you must also be

familiar with speed mismatches and duplex mismatches, and the consequences of each.

8.3.1 Speed mismatches

Speed mismatches - when two connected interfaces attempt to communicate at different speeds - are a fairly simple problem. They are almost always caused by a misconfiguration (ie. one interface configured with `speed 100` and the other configured with `speed 1000`). The result of a speed mismatch is that both interfaces will be in a down/down state (referring to the `Status` and `Protocol` columns in the output of `show ip interface brief`). The two devices will not be able to communicate with each other. Figure 8.6 demonstrates this

The following examples show the output of `show ip interface brief` for both R1 and SW1 after configuring mismatching speeds on each:

Figure 8.6 A speed mismatch between a router and a switch. Because of the speed mismatch, their interfaces are in a down/down state; R1 and SW1 cannot communicate.

```
R1(config)# interface g0/1
R1(config-if)# speed 1000
```

```
SW1(config)# interface f0/1
SW1(config-if)# speed 100
```

Speed mismatches are a risk when manually configuring interface speeds; there is always a chance for human error. However, if you're careful, they shouldn't occur.

R1# show ip interface brief

Interface IP-Address OK? Method Status Protocol

GigabitEthernet0/1 192.168.1.1 YES NVRAM down down

. . .

SW1# show ip interface brief

Interface IP-Address OK? Method Status Protocol

FastEthernet0/1 Unassigned YES NVRAM down down

. . .

For the CCNA exam, remember the result of a speed mismatch: both interfaces will be in a down/down state. The interfaces will not be operational.

Exam Tip

Duplex mismatches - when two connected interfaces are operating at different duplex settings - can be a bit harder to identify than speed mismatches. The reason for this is that, even with a duplex mismatch, both interfaces will be operational - they will be in an up/up state and able to forward network traffic. A duplex mismatch can occur when each end of the connection is configured with a different duplex setting (duplex full and duplex half), or when one end is using autonegotiation and the other isn't (as we saw in figure 8.5).

Note

Two devices connected with UTP or fiber Ethernet cables can both send and receive traffic at the same time without collisions occurring. If there is a duplex mismatch, the half-duplex device may detect false collisions when the full-duplex device transmits, but a collision hasn't actually physically occurred. Actual collisions should only occur in a wired LAN when connecting multiple hosts together with a hub.

Note

It is considered best practice to disable unused switch ports with shutdown.

Summary

- Router interfaces are disabled by default (they have the **shutdown** command applied), but switch interfaces are enabled by default.
- It is considered best practice to disable unused switch ports with **shutdown**.
- An interface description is a string of text used to describe an interface. It is optional (but recommended) and can be configured with the **description** *description* command in interface configuration mode.
- You can use the **interface range** command to configure multiple interfaces at once.
- You can use **show interfaces description** to view a list of interfaces along with their status and description.
- You can use **show interfaces status** (on switches only) to view a list of interfaces and other information, like their description, status, duplex, and speed.
- An interface's speed is the maximum rate at which it can send and receive traffic. Most interfaces support multiple speeds, such as 10/100 or 10/100/1000.
- An interface's speed can be configured with **speed** {*speed* | **auto**}, where *speed* is specified in Mbps. The default setting is **speed auto**, which uses autonegotiation to determine the interface's operating speed.
- You can use **show running-config interface** *interface-name* to view the active configurations for a specific interface or **show running-config | section interface** to view the active configurations for all interfaces.
- To avoid clutter, default configurations often do not appear in the running-config (or startup-config). For example, the **speed auto** command is not shown.
- An interface's duplex setting determines whether it is able to send and receive data at the same time. An interface operating in half duplex can send and receive data but not

at the same time. An interface operating in full duplex can send and receive data at the same time.

- The opposite of duplex is simplex, which is one-way communication.
- Wireless LANs operate in half duplex. Wired LANs using switches operate in full duplex, but wired LANs using hubs operate in half duplex.
- Hubs are Layer 1 devices that simply repeat signals received on an interface out of all other interfaces; all devices in the LAN receive every frame sent in the LAN. They are legacy hardware, rarely (if ever) used in modern networks.
- Hubs do not have memory to store frames before flooding them. If a hub receives two frames at once, it will attempt to flood both at once, resulting in a *collision*.
- A collision domain is a network segment in which simultaneous transmissions will result in collisions. Only one host in a collision domain can transmit at a time.
- All hosts connected to a hub are in the same collision domain, but all hosts connected to a switch are in their own collision domain (and therefore don't have to worry about collisions).
- Devices operating in half duplex use carrier-sense multiple access with collision detection (CSMA/CD) to detect and deal with collisions.
- An interface's duplex can be configured with **duplex** { **auto** | **full** | **half**}. The default is **duplex auto**.
- Autonegotiation allows devices to automatically determine what speed and duplex settings an interface should use. The two devices advertise their capabilities to each other and select the best mode supported by both devices.
- If only one device is using autonegotiation, it will try to sense the speed of the other device. If that fails, it will use the slowest supported speed. If the speed is 10 or 100 Mbps, it will use half duplex. If the speed is 1000 Mbps or greater, it will use full duplex.
- Cisco IOS uses various counters to keep track of errors encountered when sending and receiving messages. They can be viewed in the output of **show interfaces**. Some examples are runts, giants, CRC, collisions, and late collisions.
- A speed mismatch occurs when two connected interfaces attempt to communicate at different speeds. This will result in both interfaces being down/down; the interfaces will not be operational.
- Speed mismatches are usually caused by a misconfiguration.
- A duplex mismatch occurs when two connected interfaces operate at different duplex settings: half and full. The interfaces will be operational, but performance will be greatly reduced.

- Duplex mismatches can be caused by one side being configured as full duplex and the other as half duplex. They can also be caused by one side using autonegotiation and the other not.