In this chapter, we will learn how to use Wireshark to inspect packets and isolate network and system problems. In this chapter, we will look at a single problem and show you how deeply you can dive into the data that is captured by Wireshark for your analysis. We will cover the inspection of a problem posed on a switched Ethernet network, very common in today’s environments where bridging loops create storms that impact the network in adversely.

In this chapter’s example, we will cover a spanning tree protocol (STP) issue and go deeper into packet analysis and what Wireshark can do, show you, and help you analyze in hopes to solve a problem.

Make sure that if you are encouraged to test any of the theories in this chapter, you do it on a test network. Spanning tree loops and the packet storms that follow it can cripple your network to the point where it cannot be used.

8.1 GETTING STARTED
As we learn more about Wireshark, we will discuss problems found on a network and specifically why they occur from the packet level. Here we will take an in depth look at a few common (and not so common) problems and what you are looking for in the packets, how to use the tool to get and view this information and how to use Wireshark to solve them. We will also look at other tools you can use to augment the use of Wireshark to solve complex network and system issues.
8.2 UNDERSTANDING THE TECHNOLOGY

STP is a network technology that helps logically manage a switched (or bridged) network that has redundant connections so that you do not have “loops” in the network topology. It is a protocol that runs on the network switches and can be configured to be optimized, however, it is generally operational by default. STP (which we will call Spanning Tree for short ongoing) is an IEEE standard. It is known by the identifier 802.1D.

The way it works is simple. It will allow data to traverse on one connection, but will block the redundant connection to prevent a loop, which in turn can cause a broadcast storm that floods the network with packets and causes all devices to process this data at a rapid rate thereby causing other symptoms, such as high CPU and I/O use. When Spanning Tree is put in place, it can keep a network very stable and it recommended whenever you have more than one connection to any single device. A Spanning Tree is created by the switches on the network and can be configured.

A device called a root bridge (usually the system with the lowest bridge ID computed by a priority number, a port number, and a MAC address) maintains the Spanning Tree for that particular network segment.

Figure 8.1 shows an example of a very simple network hierarchy where five network switches are connected together to form a LAN.

In Figure 8.1, we see ports (where connecting links uplink to other switches) configured in a way where data can traverse the network without creating a loop. The root port (RP) forwards the data based on a computation of least cost path from the switch in which it is connected. The designated port (DP) is the least cost path for that segment that connects each switch. This creates the least cost path. However, if a loop is present from a redundant port, that port is put into a blocked state and becomes a blocked port (BP).

When the Spanning Tree is configured, running and optimal and all ports are stable as well the Spanning Tree is considered “converged.”

Ports become a RP, DP, or BP by going through a series of states. When a connection is made to a switch the port (through Spanning
Tree) will go through the process of listening, learning, forwarding, blocking, or disabling. These are explained as follows:

1. Listening: When a port is listening, it does not pass traffic. It does not populate the MAC address table where the switch makes its switching decisions.
2. Learning: When a port is learning, it does not pass traffic. It does populate the MAC address table where the switch makes its switching decisions.
3. Forwarding: When a port is forwarding, it is sending traffic based on the MAC address table where the switch makes its switching decisions.
4. Blocking: If a port is identified as a potential for a loop, the port is put into blocking state. This does not disable the port, it only blocks traffic to and from it. The reason why this is important is because when you want redundancy this port becomes part of the redundancy. For example, you may have a switch to fail over to another switch if it fails. If one switch fails, the BP switches to forwarding and data continues to flow through the network.
5. Disabling: Normally a port is in disabling state when it is manually shutdown or disconnected completely.
It is recommended that you use rapid spanning tree protocol (RSTP) IEEE 802.1w to limit the default timers used in order to reach convergence. RSTP uses roles, such as root, designated, alternate, backup, and disabled. It ports the port states learning, forwarding, and discarding.

The states in which the Spanning Tree transform into and how it maintains convergence is done by information sent to and from each switch through bridge protocol data units (BPDUs).

A BPDU is the data that traverses the Spanning Tree topology devices through the network to control how the Spanning Tree operates. When using Wireshark, this is specifically what you will capture in order to troubleshoot Spanning Tree problems. A BPDU frame is broken down into 12 fields as shown in Figure 8.2.

These fields will become relevant when you start to capture Spanning Tree data and review within the Wireshark capture window.

| Protocol identifier | Protocol version identifier | BPDU type | Flags | Root identifier | Root path cost | Bridge identifier | Port identifier | Message age | Max age | Hello time | Forward delay |

Figure 8.2 BPDU fields.
8.3 CAPTURING AND FILTERING DATA

Now that we understand how a Spanning Tree is supposed to operate, let’s look at some common problems that may occur within it and why when capturing data with Wireshark is equally important to understand. Some common problems you may encounter are (but not limited to):

- Spanning Tree 802.1D used instead of 802.1w
- No root bridge configured or a root bridge configured on an underpowered device not centered in your topology
- Using redundancy [with protocols such as Cisco’s hot standby router protocol (HSRP)] and designing Spanning Tree incorrectly when using it
- Too many redundant links in a blocking state
- Not using STP or any other technology of its kind.

In this example, we connected to a switch in the core of the network closest to the center of the topology. Spanning Tree when captured by Wireshark can be filtered by using the Expression Filter as shown in Figure 8.3. As you review the filter expressions, it should clear that everything we just learned about is something you could filter for within the captured data you collect. In this example, we will look for the location of the root bridge using the captured BPDU frames.

Once we filter the data, we can see the root bridge captured in Figure 8.4. In this example, we filtered for the STP. BPDUs sent through the network every 2 seconds are sent via a multicast address. The address is 01:80:C2:00:00:00.

To do a deep dive of this, we need to select a frame with this multicast address and filter deeper for the root bridge. We can then move from the Summary pane into the Detail pane as shown in Figure 8.5.

The root bridge can be found within the sections of the Details pane, which we will dig into next.

8.4 INSPECTION OF THE DATA

Now that we have captured and filtered for the relevant data, we are now ready to do a deeper inspect of the data within the frame. We need to recall the BPDU frame we learned about in Figure 8.2. In the
Figure 8.3 Filtering STP.

Figure 8.4 Wireshark with captured data.
Details pane, we can review the specifics we learned about such as first reviewing the Frame field as shown in Figure 8.6. Here we can review when the frame was captured and its size.

Next, we can review the Ethernet field as shown in Figure 8.7. Here we can review the source and destination MAC address. We can see in the destination section the multicast address 01:80:C2:00:00:00. We can see the source MAC address comes from a Cisco switch.

Once we have found where the frame is coming from and where it is going to, we can inspect the LLC field. In Chapter 1, we reviewed the OSI model and learned that at layer 2 of the model we see its split into two layers, one being the LLC. When frames operate at layer 2 (using MAC addresses), the LLC field maintains specific data at a higher level in the model. In Figure 8.8, we can see that data as the BPDU information.

Once we have reviewed that data, we can see in the STP field (as shown in Figure 8.9) the specifics we were looking for, primarily the root bridge assignment as well as the MAC address associated with it.
Now we know what system holds the identity of the root bridge. To find what device this is on your network, you can either look in the MAC address tables and find the relevant port, then map the ARP tables to IP addresses if a layer 3 assignment is given.

8.5 ANALYSIS TOOLS

To maintain a healthy Spanning Tree and be able to monitor its stability, you should consider not only designing it correctly, but also
managing and monitoring it the best you can. There are tools available to help you to do this.

Two notable tools that can help you manage and monitor your Spanning Tree network is the SolarWinds (www.solarwinds.com) network monitoring tools or the Cisco (www.cisco.com) network monitoring tools called Cisco Prime.

Other technologies you can put into place are configurations, such as RootGuard, PortFast, UplinkFast, LoopGuard, BPDUGuard, BPDUFilter, and BackboneFast from Cisco, which help you to further manage the stability of your Spanning Tree network.

8.6 SUMMARY

In this chapter, we looked at the specific details you could find when inspecting packets using Wireshark. In the example given, we took a look at how understanding a technology like Spanning Tree is critical to the process before a capture is even taken. Then, understanding the topology and where to capture data from equally important to finding specifically what you are looking for. We then filtered the data to find what we needed and inspected the data closely to find what we were looking for. In the next chapter, we will look deeper into the data to find out ways Wireshark can assist you with finding more complex issues and how to use other tools to help find a problem’s root cause.