Calculating an 802.1D STP Topology

By Keith Bogart
Learning@Cisco
TS Training Team

As an instructor, I enjoy teaching students how the IEEE 802.1D Spanning Tree Protocol (STP) works, and enlightening those students to the point where I can draw a topology diagram on a whiteboard and, no matter how complex the diagram is, the students can eventually figure out the spanning-tree role and the state of all switches in that diagram.

That is the same knowledge I will now attempt to impart to you, the reader, in this article.

There are many facets to 802.1D STP that I won’t cover in this article. The protocol has many enhancements that make it converge quicker and detect failures faster. This article is not meant to describe those features.

The goal of this article is very simple. After reading this article, the learner should be able to look at any topology diagram of LAN switches and (given the appropriate information about each switch) be able to determine the following information:

1. Which switch will be the spanning-tree root bridge
2. Which switch will be the secondary root bridge, in the event that the primary root bridge fails
3. Which switch ports are in the forwarding designated state
4. Which switch ports are in the forwarding –root port state
5. Which switch ports are in the blocking nondesignated state

I’m also going to start with the assumption that you are already aware of why 802.1D STP is needed in Layer 2 switched environments. In short, STP was developed to block bridging loops. In its simplest terms, a bridging loop means that a broadcast frame can be flooded out of one switch and, due to the cabling in the topology, eventually find its way back to that very same switch. This is a loop. This is bad, and I’m going to assume at this point that you understand why it is bad.

So let’s start with the basics.

To begin, many times in this article I will use the terms “bridge” and “switch” interchangeably. While there are distinct differences between bridges and switches, those differences are irrelevant from a spanning-tree perspective.

All Cisco switches run 802.1D STP on any active VLAN by default. There is nothing you need to do manually to turn it on. The only requirement that the switch has is that the VLAN is “active”—meaning that there are one or more ports that are UP/UP and that are in that VLAN. This port could be a switch port or a VLAN trunk; it doesn’t matter to the protocol. If a VLAN exists on a switch that doesn’t have any VLAN trunks (Inter-Switch Link [ISL] or IEEE 802.1Q—it doesn’t matter) carrying that VLAN, and there are no switch ports configured to be in that VLAN (ports connected to an edge device such as a laptop, PC, or server), then the CPU of that switch will consider that VLAN to be inactive and will ignore it from a spanning-tree perspective. Keep in mind that because ISL and 802.1Q VLAN trunks carry all VLANs by default, a single, functional VLAN trunk will also meet the criteria of making VLANs active.
Let’s start with the premise that you have the following simple topology:

![Topology Diagram]

In the preceding topology, the switch considers VLAN-1 to be active, because port 3/1 is configured to be in this VLAN and this port is actually connected to a live device. The moment that port 3/1 comes up at Layer 2, the CPU in this switch will begin to calculate 802.1D STP for VLAN-1. Notice that in this simple drawing there is no bridging loop, so technically speaking, STP is not needed. In reality, it would just be additional overhead for the CPU on this switch to run the protocol. However, unless you are absolutely sure that there will never be another switch connected to this one, and that there will never be any possibility of a bridging loop, it’s better to just leave STP turned on as a safeguard against future loops.

**Spanning-Tree Bridge ID**

The first thing the CPU does in any switch when starting STP is to determine a descriptive identifier for itself. This identifier is called the bridge ID. This bridge ID is frequently used in the spanning-tree process when two or more switches are attempting to elect something between them, such as who will be the root bridge, or which port on a shared segment will forward data frames. I’ll talk more about that in a moment. Think of the bridge ID as being similar to the name of the switch. Instead of the switch calling out, “Hey, my name is Tom,” it uses a bridge ID. A spanning-tree bridge ID contains two pieces of information: a bridge Priority value and a MAC address. These two values put together are called the bridge ID. The bridge priority, unless manually changed, is always the default value of 32768. So whether you bought a switch 20 years ago that has only 4 ports, or you bought the latest $200,000 switch with all the bells and whistles, they will always have the same bridge priority of 32,768. That being the case, the one value you can always count on as being unique from one switch or bridge to the next is the MAC address. So the combination of the bridge priority and bridge MAC address will always result in a unique bridge ID for each and every switch.

![Spanning-Tree Bridge ID Diagram]

**Spanning-Tree Root Bridge Election**

Next, every VLAN that is active and running STP will have a single bridge or switch acting as the root bridge for that VLAN. This happens by means of an election process that I’ll describe in a moment. All the switches will initially exchange special spanning-tree protocol data units (PDUs) that are called bridge protocol data units (BPDUs). A BPU D is just another kind of Ethernet frame, but in this case it carries information that STP needs for calculating the topology. PCs, servers, routers, and so on, will also receive a BPU D if they are connected to a switch, but because these devices don't typically run 802.1D STP, they will discard these frames.
The root bridge within STP has a few very important tasks, such as these:

1. It is the only switch (when running 802.1D STP) that is responsible for generating new BPDUs. All other switches in the topology will simply receive BPDUs from the root bridge and then forward them on to other, downstream switches. If the root bridge temporarily stops creating BPDUs (maybe because the CPU is too busy running other protocols), all other switches will be silent.
2. It controls the various timers that STP uses.
3. It informs the Layer 2-switched topology of something called topology changes (not covered in this article).

When a new switch comes online and determines that it needs to run 802.1D STP for one or more VLANs, it is not initially aware of the topology. It doesn’t know if it is the only switch or one of potentially hundreds of other switches in the topology. Not knowing this information, the switch takes the safest course of action and immediately begins flooding the topology with its own BPDUs and advertising itself as the spanning-tree root bridge. How exactly does it do this? Take a look at the sniffer trace of a BPDU:

```
BPDU: ----- Bridge Protocol Data Unit Header ----- 
BPDU: Protocol Identifier = 0000
BPDU: Protocol Version = 00
BPDU: BPDU Type = 00 (Configuration)
BPDU: BPDU Flags = 00
BPDU: 0... = Not Topology Change Acknowledgment
BPDU: ...0 = Not Topology Change
BPDU: .000 000. = Unused
BPDU: BPDU: Root Identifier = 8000.00400BA009A2
BPDU: Priority = 8000
BPDU: MAC Address = 00400BA009A2
BPDU: BPDU: Root Path Cost = 0
BPDU: Sending Bridge Id = 8000.00400BA009A2.8005
BPDU: Priority = 8000
BPDU: MAC Address = 00400BA009A2
```

The preceding sniffer trace doesn’t show the entire BPDU, but I’ve highlighted two important sections. First, notice the “Sending Bridge Id.” This is the same as the bridge ID I just described. The priority shown here as 8000 is really a hexadecimal number. Better put, it would be displayed as 0x8000, which (when converted to decimal) is 32,768. The MAC address of this switch is 00-40-0B-A0-09-A2. The final number of 8005 is not technically part of the bridge ID. This number represents another descriptive identifier, called the Port-ID. This is a unique number representing the port or interface that is transmitting this BPDU. I’ll talk more about that later.

The other highlighted field in the BPDU tells the whole world who the spanning-tree root bridge is for this VLAN. In this case, this switch is advertising itself as the root bridge.

In the world of STP, there can be only one root bridge per VLAN. As mentioned previously, switches that don’t initially understand the topology will generate BPDUs and claim to take on this role. But as soon as switches receive BPDUs from other switches in the topology, then an election process must take place to determine who the "real" root bridge should be.
The root bridge election process is very simple. Switches look for the switch that has the numerically lowest bridge ID, and elect that switch to be the spanning-tree root bridge.

Keeping in mind that a bridge ID is technically composed of two subfields (bridge priority and MAC address), the election process happens as follows:

1. All bridges exchange BPDUs with each other.
2. The bridge priority value is viewed and compared.
3. If one bridge has a numerically lower bridge priority than all the others, the election process stops and that bridge is elected the spanning-tree root bridge.
4. If all the bridge priority values are identical, the switches compare each other's MAC address.
5. Because each bridge is guaranteed to have a unique MAC address, one of them is numerically lower than all the others, and that bridge is elected the spanning-tree root bridge.

So in the following topology, which switch do you believe would end up becoming the root bridge?

![Topology Diagram]

Hopefully, you answered the bridge-switch with a priority of 30,768. Remember to always compare the bridge priority values first. Only if two or more of them are identical will you need to move on to comparing MAC addresses. And always read MAC addresses from left to right. So if the bridge priority values had all been identical, which switch would have won?

The switch with MAC address 04-00-00-cc-12-77 would have won. And you would have known this after comparing only the third number (04-0). There would have been no need to continue comparing the digits in the rest of the MAC addresses, because the “0” after the “04” is definitely lower than 04-1 or 04-8.

At this point, the root bridge no longer needs to proceed in any further elections. It will start generating and transmitting BPDUs out every port or interface it has in VLAN-1. If this switch also became the root bridge for other VLANs, it would generate BPDUs for those other VLANs as well. A switch can be a root bridge for more than one VLAN. As a matter of fact, if all the switches were left to their default values, whichever switch became the root bridge for VLAN-1 would also, by default, be the root bridge for VLAN-2, VLAN-3, and so on.

One remaining fact about the root bridge is that all the ports it has in this particular VLAN (including VLAN trunks) will be placed in the forwarding designated role. I will talk more about that in a moment.
Spanning-Tree Root Ports and Cost

The switches that lost the root bridge election have some more work to do before STP is stabilized.

Every port that is participating in STP (basically, every port that is “up”) will eventually end up in one of three spanning-tree port roles:

1. Root port
2. Designated port
3. Nondesignated (blocking) port

The next step that each switch (except for the root bridge) determines is which of its ports will become the root port. For each active VLAN on a switch, that switch will have only one root port. The root bridge itself has no root ports.

The root port is that port on the switch that has the fastest path back to the root bridge. Look at an example:

In the drawing, both Bob and Cindy can drive directly to the airport. Bob can drive directly down K Street, and it will take him 12 minutes to reach the airport. Cindy can drive directly down M Street, and it will take her only five minutes. But they are also both connected to L Street. Unless someone connected to L Street advertises that this is a viable path to the airport, neither Bob nor Cindy will even consider driving on L Street. After all, would you drive down a dark alley if you weren’t sure it led to where you wanted to go? So as a courtesy, both Bob and Cindy advertise to each other (along L Street) the paths they know about that lead to the airport. If Bob and Cindy knew about multiple ways to get to the airport, they would only inform each other of the best way (the fastest).

Notice that when Bob shouts to Cindy that he can get to the airport in 12 minutes, he doesn’t tell her about the 6 additional minutes it would take for her to travel across L Street. He only tells her his perspective. And from Bob’s perspective, he only knows that K Street leads to the airport. Likewise, when Cindy shouts, “Hey, I have a way to the airport over here and it only takes me five minutes,” she also doesn’t take into consideration the six minutes it would take Bob to drive to her house on L Street.
Both Bob and Cindy now know they each have two possible ways to get to the airport. For Bob, one way will take a total of 12 minutes (along K Street). The other way (along L Street) will actually take less time—a combined total of 11 minutes. So even though Bob is directly connected to the airport by way of K Street, it would actually be faster for him to get there by first driving down L Street, through Cindy’s house, and then along M Street. So Bob would choose his connection to L Street as his root port.

Cindy’s direct connection along M Street is still the fastest way for her to get to the airport, so this connection will be selected as her root port.

This example used driving and time expressed in minutes to make a point. But what about switches? How do they determine how fast they can get to the root bridge? Electrons can travel from one switch up to the root bridge in microseconds. Is that the value used?

No. In spanning-tree terminology, the speed of a link is represented by a numerical value called cost. You don’t need to know the exact formula used to derive cost, but you should know that cost is directly related to bandwidth. The higher the bandwidth is on a link, the lower the value of its cost.

Remember when two or more bridges were trying to determine who would be the root bridge? They compared each other’s bridge ID, and the bridge with the lowest value was the winner. In STP, whenever you are comparing two or more values against each other to see which one is “better,” the lowest value always wins.

Using that same rule for cost, you can understand, hopefully, why a higher-bandwidth link translates to a lower cost. The lower the number, the better it is. So while a 10-Mb/s Ethernet link has a cost of 100, a Gigabit Ethernet link has a cost of only 4. These are the cost values you should memorize:

<table>
<thead>
<tr>
<th>Speed of link</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-Mb/s Ethernet</td>
<td>100</td>
</tr>
<tr>
<td>100-Mb/s Fast Ethernet</td>
<td>19</td>
</tr>
<tr>
<td>1000-Mb/s Gigabit Ethernet</td>
<td>4</td>
</tr>
</tbody>
</table>

The other rule regarding cost you need to memorize is this:

*A switch always advertises its perceived cost to the root bridge to its neighbors.*
Let’s look at the following example:

From the root bridge’s perspective it doesn’t cost anything to reach itself. So the root bridge always advertises BPDUs with an initial cost of 0. Here’s another way of looking at it. If someone asked you, “When you’re at the airport, how many minutes does it take you to drive to the airport?” you’d probably scratch your head and assume this was a trick question. But the honest answer would be, “If I’m already at the airport, it doesn’t take me any time at all, because I’m already there, so the answer is 0 minutes.”

When Bridge A receives this BPDU with a cost of 0, it receives it on interface Gigabit Ethernet 4/1. Knowing that this is a Gigabit Ethernet interface, it calculates its local cost value for this port (a value of 4) and adds it to the cost in the BPDU. Now the total cost for Bridge A to reach the root bridge is 4. Considering that this is the only path to the root bridge it knows about, it will advertise this cost out port 5/2 so that Bridge B can learn about it.

Similarly, when Bridge B receives this BPDU with a cost of 0, it receives it on interface Fast Ethernet 3/1. Knowing that this is a Fast Ethernet interface, it calculates its local cost value for this port (a value of 19) and adds it to the cost in the BPDU. Now the total cost for Bridge B to reach the root bridge is 19.
Let’s focus our attention on Bridge B for a moment.

Bridge B has not only received a BPDU directly from the root bridge on port 3/1 with an initial cost of 0, but milliseconds later it receives another BPDU on port 3/2 from Bridge A. In this BPDU, Bridge A is advertising that it can reach the root bridge with a cost of 4. Once again, Bridge B has to take into account the path between itself and Bridge A and add that to the equation. The local interface cost of port 3/2 on Bridge B is 19 (this is a Fast Ethernet link). When added to the advertised cost of 4 from Bridge A, Bridge B now knows that it has an alternative path to the root bridge with a combined total cost of 23.

Bridge A makes this same calculation based on the BPDU that it receives from Bridge B and determines that its combined cost to reach the root bridge from port 5/2 would be 38.

Both Bridge A and Bridge B are now aware that they have two paths to the root bridge. Sticking with the same rule that “lowest is better,” Bridge A will select port 4/1 as its root port, and Bridge B will select port 3/1 as its root port.
A root port is always in the “forwarding” state. This means that this port is wide open to send and receive all types of traffic. It is forwarding Ethernet frames.

**Spanning-Tree Designated Ports**

So what about the ports that are not root ports? What are they?

I’ve already mentioned that the root bridge itself has no root ports, so what will be the spanning-tree role of ports on that bridge or switch?

As I mentioned previously, every port that is participating in STP will have one of three possible roles:

1. Root port
2. Designated port
3. Nondesignated (blocking) port

On Bridge A, we know that port 5/2 is not the root port, so by process of elimination it must be either a designated port or a nondesignated port.

Now for the next rule you should memorize: *Every collision domain has a single spanning-tree designated port.*

Notice that I used the words “collision domain” and not simply “cable.” In this topology, cable segment X is considered a single collision domain, and so one of the ports connected to this cable will become a designated port:

But in this topology, everything connected to the hub is in a single collision domain as well (collision domain Y), and so only one of the many ports on the switches connected to this hub will end up becoming a designated port:
So what exactly is a designated port? In its simplest terms, a designated port is the port within that collision domain that has the fastest path back to the root bridge. Let's use our airport analogy again:

In the preceding drawing, both Bob and Cindy have opened their own limousine airport service, and both have their own route to the airport. The airport here would be analogous to the root bridge. Bob and Cindy also live on the same street (a neighborhood where the speed limit is a meager 25 mph). A single street can support only one limousine service (analogous to the designated port concept), so Bob and Cindy both advertise the fastest path they have for getting to the airport, because they want to attract customers. Bob advertises that if people walk through his front door, he can take them to the airport at a speed of 45 mph. Cindy advertises that if people walk through her front door, she can take them to the airport at a speed of 70 mph. Obviously, the people on this street will use Cindy's limousine service, because she can get them to the airport faster. So on this street, Cindy's front door will be open to the public and will serve as the designated port.
Let’s bring the focus back to spanning-tree terminology. You now know that STP uses “cost” to reflect the bandwidth of a link.

In the preceding figure, we see a total of three collision domains. Each cable in the figure represents a collision domain. On the Gigabit Ethernet cable, which switch do you think has the lowest combined cost to reach the root bridge—the root bridge itself, or Bridge A?

Hopefully, you answered the root bridge. After all, the root bridge advertises itself with a cost of 0, and nothing can beat that! This means that every port on the root bridge (in that particular VLAN) will be a designated port.

Here’s another rule of thumb you can memorize: Once you determine the root port on a cable, you can automatically go to the other end of that cable and mark that port on your topology diagram as a designated port.

So the only collision domain or cable left to determine in this topology is the Fast Ethernet connection between Bridge A and Bridge B.
Both bridges want to win this process, but there can be only one designated port per cable (collision domain). Considering that STP considers lower values to be better, it is logical that even though Bridge A has two costs to reach the root bridge (a cost of 4 on port 4/1 and a cost of 38 on port 5/2), it will advertise the absolute lowest cost it has to Bridge B to attempt to win this designated port-election process.

In this topology, Bridge A will win and port 5/2 will become the designated port on this cable. Just like root ports, designated ports are always in the forwarding state.

What about port 3/2 on Bridge B? By process of elimination, we know it is not the root port. It’s not the designated port, so it must be a nondesignated port. Nondesignated ports are those special ports that stop the potential bridging loop. They do this by going into the “blocking” state. A port that is in the blocking state is not allowed to send any user data frames. In other words, any frames from your laptop, PC, or server that reach a blocking port are discarded. The only types of Ethernet frames that are allowed to be transmitted on a nondesignated blocking port are management frames (like Cisco Discovery Protocol, VLAN Trunking Protocol, Dynamic Trunking Protocol, and so on).
In the topology I just used, everything was pretty easy because the link speeds (costs) were different. But what if I made one minor change?

Notice that the Gigabit Ethernet link between the root bridge and Bridge A has been changed to a Fast Ethernet link. This change would result in the following new cost values:

As we can see, Bridge A still shows port 4/1 as being its root port, because it has less total cost to use this port to reach the root bridge than if it used port 5/2 with an aggregated cost of 38. And the same holds true for Bridge B; its root port hasn’t changed. But how will these switches determine which port will be the designated port on the cable that they share? Both switches are advertising the same cost to each other (a cost of 19).

In this situation, when trying to determine the designated port in a collision domain and two or more of the costs are identical, a tie breaker is used: the bridge ID.
Recall that every bridge has a unique bridge ID. Just as this bridge ID was initially used to determine the root bridge, it can now be used again (when the costs are identical) to determine the designated port.

Notice the difference. When we previously had a Gigabit Ethernet link between the root bridge and Bridge A, it was clear that port 5/2 on Bridge A was the designated port (because this bridge had a lower advertised cost to the root bridge than Bridge B). But now, because the costs are identical, Bridge B wins! Bridge B’s bridge ID is lower than Bridge A’s bridge ID. Recall that the bridge ID is composed of the bridge priority followed by the MAC address. Right away we can see that Bridge B has a lower bridge priority than Bridge A, so we don’t even need to compare the MAC addresses.
At this point, you should be able to work out a basic STP topology for yourself. Try it! Use the following topology and see if you can answer these questions (the solution is at the bottom of this article):

1. Which switch will be the spanning-tree root bridge?
2. Which port(s) will be spanning-tree root ports and in the forwarding state?
3. Which port(s) will be spanning-tree designated ports and in the forwarding state?
4. Which port(s) will be spanning-tree nondesignated ports and in the blocking state?

Lastly, there is one more permutation of a topology that you most likely will experience.

Hopefully, by this point you understand why the switch on the left was elected the root bridge (it had a lower bridge ID). And I’ve explained that all the ports on the root bridge are always designated ports because they have a cost of 0.

Normally, the next step in the process would be to determine the root port on Bridge B, and we would do that by trying to figure out which port on Bridge B had the lowest cost back to the root bridge. But in this picture, both ports on Bridge B have the same cost to the root bridge—a cost of 19.

I then mentioned that the next tiebreaker (when the cost was identical) was to look at the bridge ID of your neighbor. But the BPDU that Bridge B receives on port 3/2 contains the same sending bridge ID (32768-04-11-11-cc-12-77) as the BPDU it receives on port 3/1.
There is a final tiebreaker in this situation. Let’s go back to our sniffer trace of the BPDU:

| BPDU: Sending Bridge Id = 8000.00400BA009A2.8005 |
|------------------|------------------|
| BPDU: Priority   = 8000                     |
| BPDU: MAC Address = 00400BA009A2            |
| BPDU: Port       = 8005                     |

What you see here is only a small portion of what is contained within the BPDU, but I want you to pay special attention to the highlighted field. This field in the BPDU is called the sending port ID. Not only does a switch come up with a unique bridge ID to describe itself, but it also assigns a unique port ID to each port. When transmitting BPDUs out of that port it “tags” that BPDU with the sending port ID. And just as the bridge ID field in a BPDU is actually two subfields (bridge priority and bridge MAC address), the sending port ID field also is comprised of two subfields (port priority and port ID value).

By default, Cisco IOS switches assign a port priority value of 128 to all ports. In the preceding example the value of 8 in “8005” is a hexadecimal number for port priority that (when translated into decimal) reflects this default value of 128.

You don’t need to know the exact mechanism used to derive port IDs on switches, but you can follow this general rule: When viewing multiple ports on a switch, a lower port number will also translate to a lower sending port ID.

So in the preceding topology, Bridge B will select port 3/2 as its root port. Notice that it does not take into consideration its own port ID in this process. It looks only at the sending port ID values contained in the BPDUs that it receives.

And by the process of elimination, port 3/1 must go into the blocking state.
Now try another couple of exercises and see how you do:

Exercise Topology #2

Exercise Topology #3
Solutions to Exercises

Solution to Exercise Topology #1

Solution to Exercise Topology #2
Solution to Exercise Topology #3