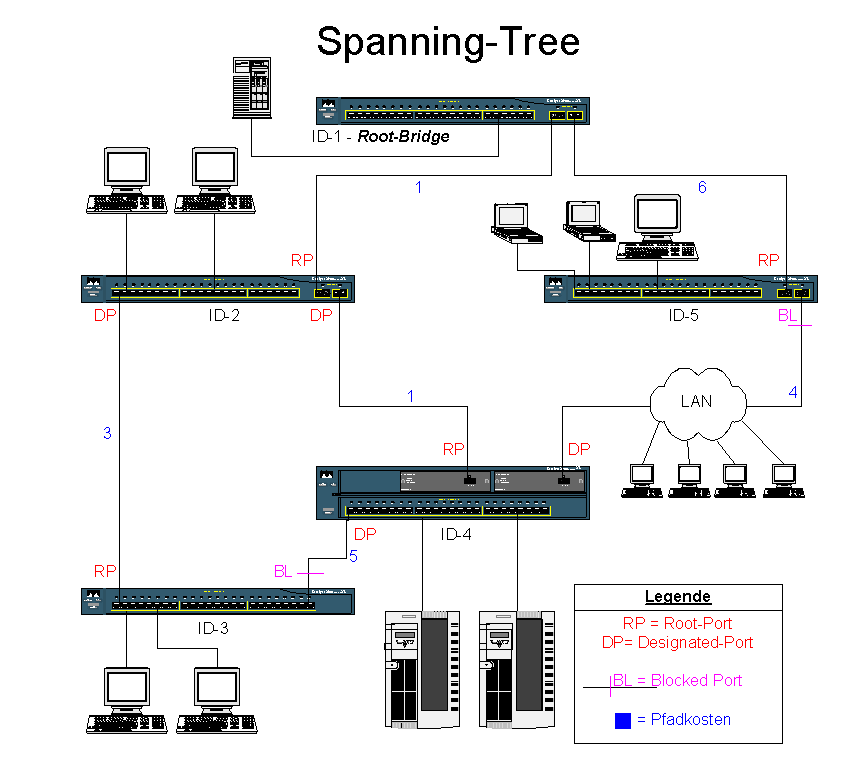
**STP (Spanning Tree Protocol)**

**What is STP?**

STP (Spanning Tree Protocol) is a feature used to prevent loops when using redundant switches.

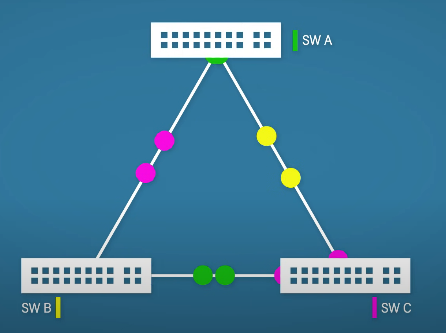
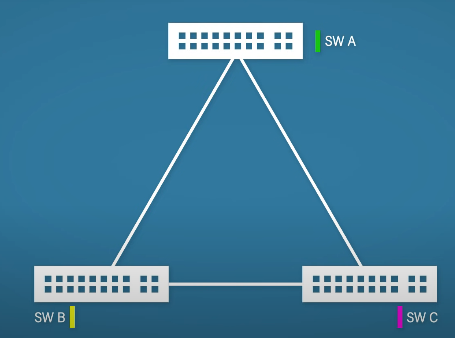
It is a Layer 2 protocol (data link, communication protocol used for devices such as switches, network interface cards, etc.), the second layer of the 7-layer OSI model.

We will first look at the problems that happen without using STP.

**Which problems occur without the use of STP?**

**Problem 1 – "broadcast storm"**

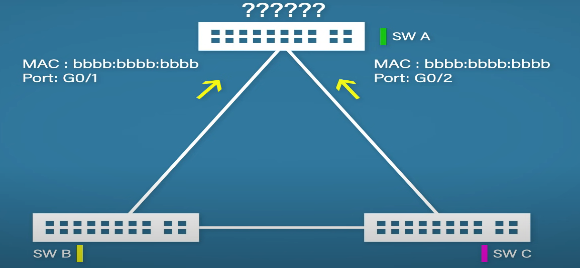
For example, let's say we have 3 switches – A B C.  
Each of the switches sends a broadcast message to the other, by the way they are connected. This creates a "broadcast storm", meaning an overflow of data caused by the infinite loop.  
In this example, the only way to stop this broadcast storm is by disconnecting/reloading a switch/





**Problem 2 – unstable mac address tables**

For example, switch B sends out a broadcast message. Switch A receives it in one of its ports, let's say, port G0/1. It reads the MAC address of that message. Now, switch C sends out the exact same message.

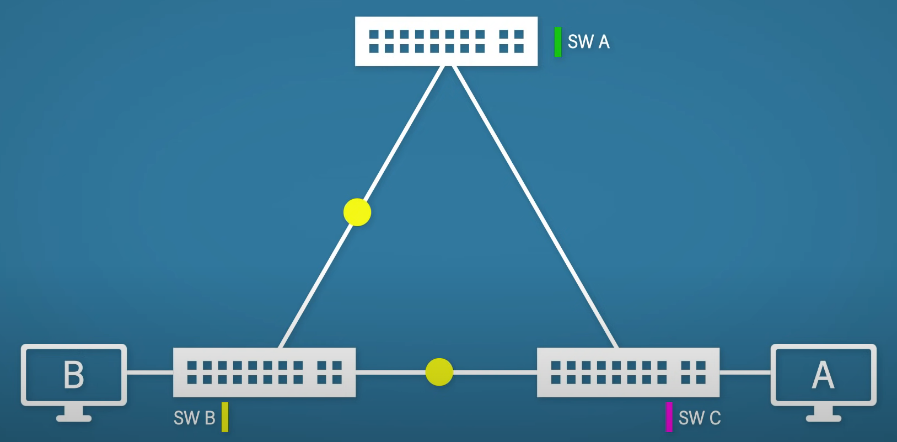
As you can see in the diagram, switch A now updates the same message to be in port G0/2 instead of port G0/1. Now, imagine there is a lot of the same data coming in switch A (the "broadcast storm" we were talking about in problem 1). It causes the switch to constantly change the mac addresses in the mac address table.

**Problem 3 – duplicate frames**

Let's look at the same diagram, but now add computer A and B.  
Computer B wants to send a message to computer A, not knowing computer A's MAC address.  
Computer B forwards the message to switch B.

Switch B, not knowing it needs to forward the message only to switch C, sends the message to A as well.

Let's say computer A gets the message first from switch C. The problem is, now switch A will forward the same message to switch C, yet again sending the message to computer A.

This problem is called duplicate frames.

We've discussed the problems, now, let's look at the solution to those problems.

The solution is rather simple and solves all the problems - switch C needs to ignore the connection with switch B.

Now let's leave this example and think of a bigger one. Imagine there are 100's of switches or even more. We can't just look at them and decide which switches need to ignore the connections with what switches.

This is where STP comes in.

**How does STP work?**

STP follows a strict process to decide which port to block, we will look at each step.

**Step 1 – Elect a root bridge**

The protocol designates the root bridge (the most "important" switch – the "king" of the switches).

**Step 2 – Place root interfaces into a forwarding state**

Each switch that is connected to the root bridge, its connection will be placed at a forward state, meaning switch A will oversee forwarding the message from switch B to switch C.

**Step 3 – Each non-root switch selects its root port**

Each non root bridge chooses a root port – the port which offers the best route to the root bridge.

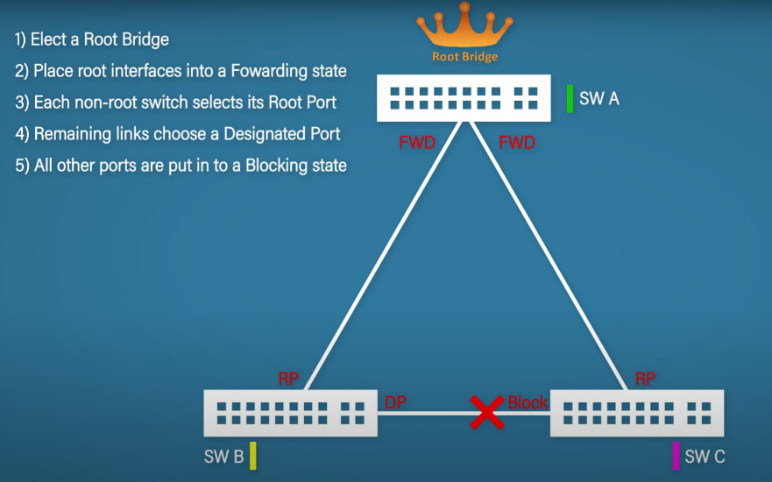
**Step 4 – Remaining links choose a designated port**

Non root bridge switches will choose whether to have a designated port or not.

For example, switch B will choose a designated port which will be connected to switch C.

**Step 5 – All other ports are put into a blocking state**

At the end of the process, all other ports will block the connection between other switches.



Let's sum up the roles:

1. Root Ports – The best ports to reach the Root Bridge
2. Designated Ports – Ports with the best route to the Root Bridge on a link
3. Non-Designated Ports – All other ports that are in a blocking state

Now we will talk about the possible states of each port:

1. Disabled – a port that is shutdown
2. Blocking – a port that is blocking traffic
3. Listening – not forwarding traffic and not learning MAC addresses
4. Learning – not forwarding traffic but learning MAC addresses
5. Forwarding – sending and receiving traffic like normal

Note: states 3 and 4 are transitional, these states occur while changing from one role to another.

**Root Bridge Election**

Let's look at how exactly the protocol elects the root bridge.

Each switch has what's called a bpdu (bridge protocol data units) which contains the root cost and BID (bridge ID).

The BID is built by adding the STP priority default value (32,678) with the VLAN number (1) and the MAC address.

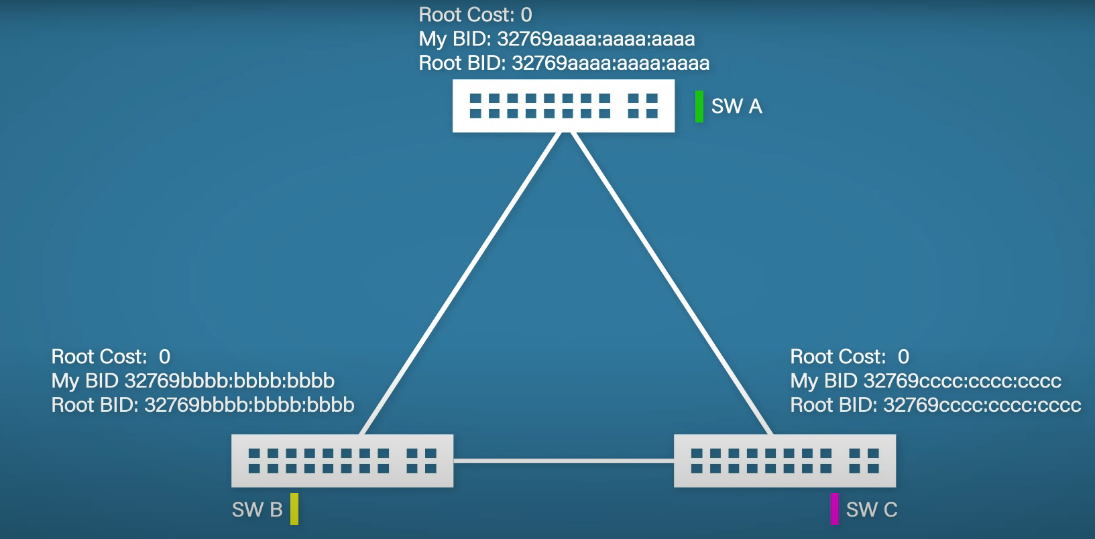
For example, 32769aaaa:aaaa:aaaa.

The switch with the lowest BID will become the root bridge.

It begs the question – what happens at the beginning of the protocol to the switches?

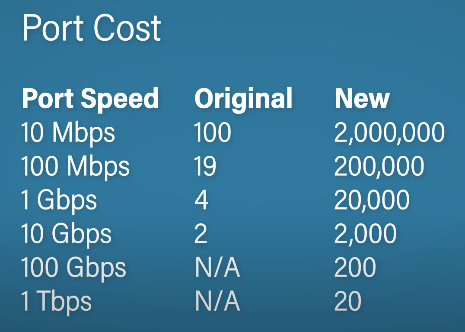
Each switch at the beginning of the protocol will select itself as the root bridge.  
That of course, creates a problem.  
Luckily, it is quickly solved.

The first data that is shared between the switches will notify them which switch has the lowest BID.

Then, they quickly change the ROOT BID to the lowest one.

**Root Ports**

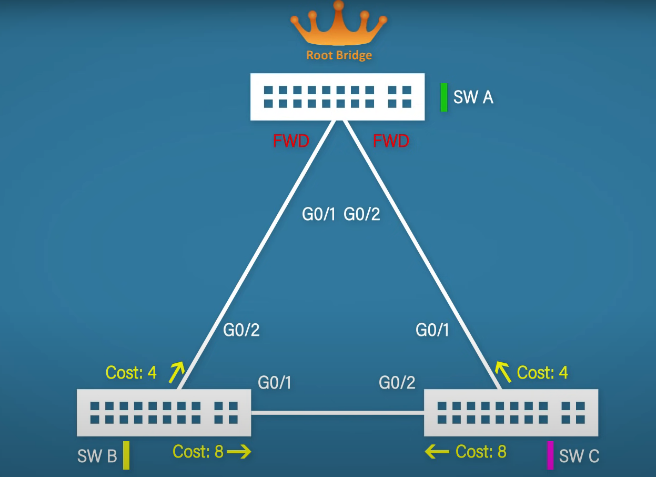
We talked about root ports before (step 3), but how does the switch decide which port is truly the best?

The answer is: root cost.

The root cost is the collection of each outgoing port to the route.

As technology advances, the original port cost was modified. Today, we use the new port costs.

For this demonstration, we will use the original costs as they are smaller.

As we can see, the root cost of the outgoing port from switch B to A is 4 (4 + 0 = 4).

The same is applied to switch C.

Now, the cost of switch B to C or the opposite is 8 (4 + 4 = 8).

The root port is based on the lowest cost so in this case, both switches choose the port with the cost of 4.

Now, a question pops up.

What happens if the root cost is the same on multiple ports? (duplicate links or the port cost got changed manually)

The decision comes to some tie breakers (in order), as follows:

1. Lowest root bridge ID – typically does not occur since all ports receive the same root bridge ID once STP has converged
2. Lowest root path cost – the port with the lowest total cost to reach the root bridge is preferred
3. Lowest sender bridge ID – the port connected to the switch with the lowest bridge ID is preferred
4. Lowest sender port ID - the port connected to the lowest port ID on the upstream switch is preferred
5. Lowest local port ID – the switch's own port with the lowest ID number is chosen

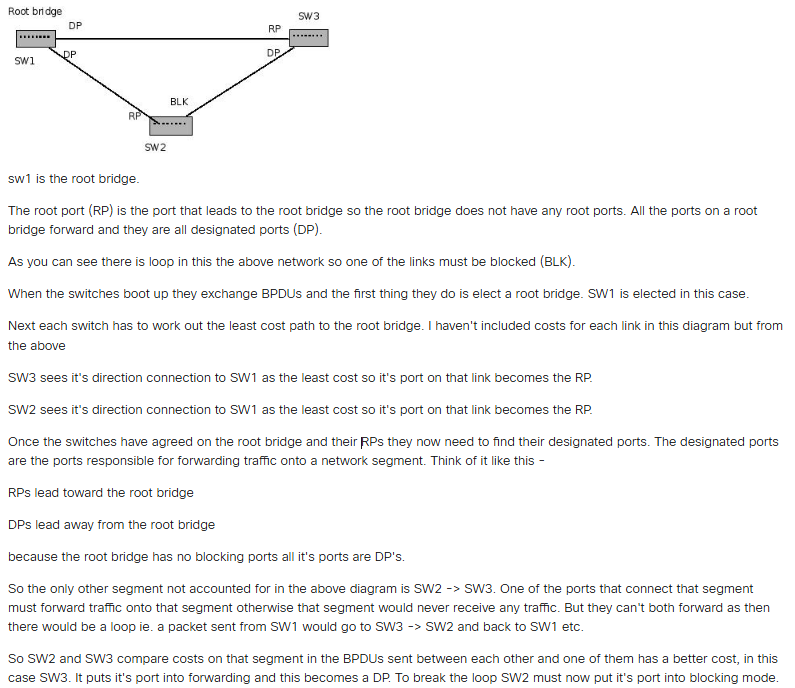
**Designated Ports**

Now we will look at how the switch chooses the DP (designated ports).

The decision is made by the same exact tie breaker sequence as in Root Ports.

Before we continue to the final stage – blocking, let's see the differences between root ports and designated ports.

1. Definition:
   * Root port: The port on a non-root bridge that provides the shortest path to the root bridge.
   * Designated port: The port on a LAN segment that has the lowest cost path to the root bridge.
2. Quantity per switch:
   * Root port: Each non-root switch has only one root port.
   * Designated port: A switch can have multiple designated ports.
3. Presence on root bridge:
   * Root port: The root bridge does not have any root ports.
   * Designated port: All ports on the root bridge are designated ports.
4. Direction:
   * Root port: Leads towards the root bridge.
   * Designated port: Leads away from the root bridge, typically downlink to other switches or network segments.
5. Selection process:
   * Root port: Chosen based on the lowest cost path to the root bridge from the switch.
   * Designated port: Selected for each LAN segment based on the switch that can provide the lowest cost path to the root bridge.
6. Relationship:
   * If one end of a segment is a root port, the other end is typically a designated port.
   * A root port can never be a designated port.
7. State:
   * Both root ports and designated ports are in a forwarding state when STP has converged.
8. Function:
   * Root port: Provides the best path from a non-root switch to the root bridge.
   * Designated port: Responsible for forwarding traffic onto a network segment.

If you still do not understand the differences between RP and DP, try looking at this explanation:

**Blocking**

Every port that is not either a root port or a designated port is put in the blocking state.

And that's the whole STP process!

We are not done yet. We still have a few things to talk about.

**STP Types**

Currently, we have 4 types of STP (going from older to newer):

1. STP / 802.1D – Original STP
2. PVST+ - Cisco improvement of STP adding a per VLAN feature
3. RSTP / 802.1W – Improved STP with much faster convergence
4. Rapid PVST+ - Cisco improvement of RSTP adding per VLAN feature

As you can see, Cisco loves to improve the STP protocol. Let's look at the VLAN feature.

**VLAN**

For starters, VLAN stands for "Virtual Local Area Network" – logical overlay network that groups together a subset of devices that share a physical LAN, isolating the traffic for each group.

The purpose of VLAN is to improve security, traffic management and to make a network simpler.

For example, imagine a company with 100 employees across different departments: Marketing, Finance, IT, and Human Resources. Without VLANs, all these departments would be on the same network, potentially causing security risks and performance issues.

By implementing VLANs, the network can be logically segmented as follows:

1. VLAN 10 - Marketing:
   * Assigned to all marketing staff computers and printers
   * IP range: 192.168.10.0/24
2. VLAN 20 - Finance:
   * Assigned to finance department devices, including accounting servers
   * IP range: 192.168.20.0/24
3. VLAN 30 - IT:
   * Assigned to IT staff computers and network infrastructure devices
   * IP range: 192.168.30.0/24
4. VLAN 40 - Human Resources:
   * Assigned to HR staff computers and printers
   * IP range: 192.168.40.0/24
5. VLAN 50 - Guest Network:
   * Assigned to a separate wireless network for visitors
   * IP range: 192.168.50.0/24

We can easily understand that the Cisco improvements of the STP add the ability to implement the protocol in VLAN as well.

**Timing**

We talked about all the process and its features, however there is a crucial topic we haven't touched on yet – time.

Let's look at the default time for each process:

1. Hello – 2 seconds.

Every 2 seconds is the time interval where the root bridge will create and send "hello" messages to all the network, letting all the switches know that everything is up to shape

1. MaxAge – 10 x Hello (20 seconds).

The time the switch will wait before it realizes something's wrong.

1. Forward Delay – 15 seconds.

The time between the listening and the learning states.

Previously we talked about states but haven't really explained the listening and learning states. We will now talk about how they are used in the blocking state.

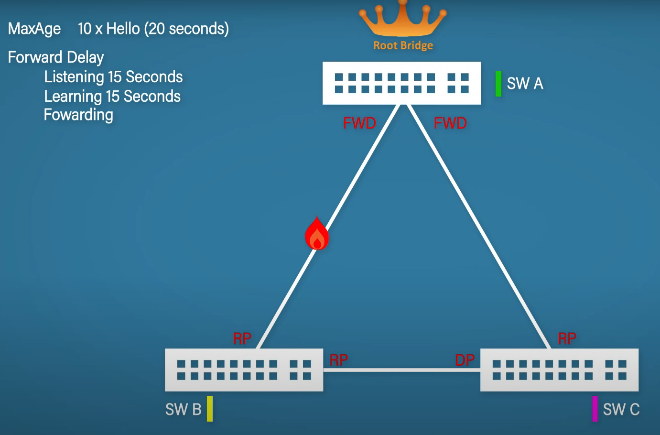
Going from the forward state to the blocking state is instantaneous and does not require to go through any other states.

However, the blocking state can't change states that easily.

Firstly, it must enter the listening state (the port listens to MAC addresses but without learning any, to stop any potential loops). This listening state is held for 15 seconds (forward delay).

After that, it enters the learning state (the switch starts to learn MAC addresses). Again, the state is held for 15 seconds (forward delay).

Once the 30 seconds have been completed, only then will the blocking state move to forward state.

Let's point out the obvious. If a connection is damaged for any reason, the whole process of restoring it takes too long in today's world.

To help speed up the general convergence we can use methods such as Portfast (Cisco technique that puts a switch interface into forwarding mode immediately, skipping the listening and learning states) and BPDU guard (security feature that prevents rogue devices from compromising the network by disabling ports when BPDUs are transmitted).

Those methods work well for the access port, so the ports that connect to devices like computers.

The best method is to use the latest STP type – currently the Rapid PVST protocol.

**Advantages and Disadvantages of STP**

**Advantages:**

– Prevents network loops by intelligently blocking redundant paths.  
– Enhances network reliability and stability by ensuring a single active path to each network segment.  
– Provides failover capability, allowing networks to continue functioning even if certain links fail.  
– Easy to implement and widely supported across various networking equipment and vendors.

**Disadvantages:**  
– Slow convergence time, especially in large networks, leading to potential network disruptions during topology changes.  
– Inefficient bandwidth utilization due to the blocking of redundant links.  
– Limited scalability, as STP may struggle to handle complex network topologies and large-scale deployments.  
– Vulnerable to misconfigurations and spanning tree-related issues, which can impact network performance and availability.

**Testing STP supporting switches**

Firstly, let's explain what use cases and corner cases are.

Use cases -  a list of actions or event steps typically defining the interactions between a user and a system to achieve a goal.

Corner cases - occur outside of normal operating parameters, specifically when multiple environmental variables or conditions are simultaneously at extreme levels, even though each parameter is within the specified range for that parameter.

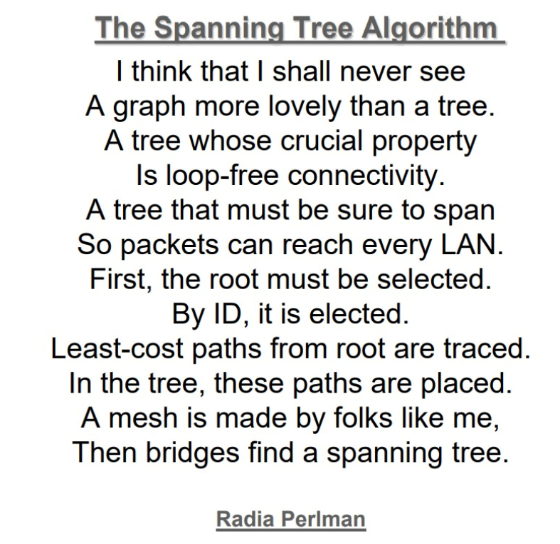
Here are some examples of use cases and corner cases:

1. Basic topology changes:
   * Add and remove links to verify proper reconfiguration
   * Simulate link failures to ensure rapid convergence
2. Root bridge election:
   * Test with different bridge priorities
   * Verify correct root bridge selection
3. Load balancing:
   * Create scenarios with multiple equal-cost paths
   * Verify traffic distribution across available links
4. Convergence time:
   * Measure time taken for network to stabilize after topology changes
   * Test under various network loads
5. Interoperability:
   * Test STP between switches from different vendors
   * Verify compatibility with newer protocols like RSTP or MSTP
6. Corner cases:
   * Introduce temporary loops to test loop detection and prevention
   * Simulate multiple simultaneous link failures
   * Test behavior with misconfigured bridge priorities
   * Verify proper handling of BPDUs with invalid information
7. Security:
   * Test protection against rogue switches attempting to become root bridge
   * Verify BPDU filtering and guard features

And that's it!

Thank you to anyone who read my explanation of STP, hope it helped.

Here is a little poem to brighten up this subject 😊



If you wish to do some exercises, click the last link in the document.

\*This document was written using the following materials:

[Spanning Tree Protocol - Wikipedia](https://en.wikipedia.org/wiki/Spanning_Tree_Protocol)

[פרוטוקול העץ הפורש - Wikiwand](https://www.wikiwand.com/he/%D7%A4%D7%A8%D7%95%D7%98%D7%95%D7%A7%D7%95%D7%9C_%D7%94%D7%A2%D7%A5_%D7%94%D7%A4%D7%95%D7%A8%D7%A9#Media/%D7%A7%D7%95%D7%91%D7%A5:Spanning_tree_topology.png)

[Spanning Tree Protocol Explained | Step by Step (youtube.com)](https://www.youtube.com/watch?v=japdEY1UKe4)

[Perplexity](https://www.perplexity.ai/)

[STP determining blocked port using cost (networklessons.com)](https://notes.networklessons.com/stp-determining-blocked-port-using-cost)

[Exercises with STP - CCNP Switch (ccnp300-115.blogspot.com)](https://ccnp300-115.blogspot.com/2016/04/exercises-with-stp.html)