

Spanning_Tree_Protocol

The **Spanning Tree Protocol (STP)** is a method for dynamically calculating the "best" spanning-tree of a computer network with or without loops. By definition a tree is loop free.

STP does its work at layer 2 (data-link) of the OSI model. That's to say it runs at the Ethernet layer and is totally unrelated to dynamic IP-based routing protocols like RIP, OSPF or OLSR which can take into account factors that may be desirable in choosing a particular path, such as bandwidth, reliability, latency etc.

A computer network's loop links may function as backup. If any link in a loop is lost, the rest of the other loops equipment is still connected.

Mesh networks without STP enabled may experience Loopback. The need for STP occurs when your router is being used in a mesh network with multiple WDS-enabled repeaters, or in an ad hoc network with multiple ad hoc connections.

If your router is not being used in either of the given scenarios, then likely STP is not necessary. Therefore, it is recommended that you disable it because newly-connected ports sit in a learning state for approximately 50 seconds before entering a non-blocking mode. This can cause certain services on the client device (such as DHCP) to time-out.

Warning: Users of Comcast Cable should always disable this option and leave it disabled, as STP conflicts with the routers DHCP client for the WAN connection.

How-to

Turn STP on like this.

Spanning_Tree_Protocol

(wrt/index.asp)

Check it like this.

(wrt/Networking.asp)

Fine Tuning STP

<http://www.dd-wrt.com/forum/viewtopic.php?t=143892>

Now the initial state: Three routers Main MAC:xxxxxxxxxxx Asus RT-N16 running
dd-wrt.v24-14929_NEWD-2_K2.6_big Media MAC:xxxxxxxxxxx1 WRT-310Nv1 running
dd-wrt.v24-14929_NEWD-2_K2.6_std_nokaid_small Mid MAC:xxxxxxxxxxx2 WRT-310Nv1 running
dd-wrt.v24-14929_NEWD-2_K2.6_std_nokaid_small

Main Has the WAN link to a PPPoE DSL Link1 = WDS neighbors Main-Mid Link2 = WDS neighbors
Mid-Media Link3 = WDS neighbors Media-Main

Spanning_Tree_Protocol

sethello	<bridge> <time>	set hello time
setmaxage	<bridge> <time>	set max message age
setpathcost	<bridge> <port> <cost>	set path cost
setportprio	<bridge> <port> <prio>	set port priority
setportsnooping	<bridge> <port> <addr>	set port snooping
clearportsnooping	<bridge> <port> <addr>	clear port snooping
showportsnooping	<bridge>	show port snooping
enableportsnooping	<enable>	enable port snooping
show		show a list of bridges
showmacs	<bridge>	show a list of mac addrs
showstp	<bridge>	show bridge stp info
stp	<bridge> {on off}	turn stp on/off

I peeked around to see what bridges were available Code:

```
root@Main:~# brctl show bridge name bridge id STP enabled interfaces br0 8000.xxxxxxxxxxxx yes vlan1
```

```
eth1
wds0.1
wds0.2
```

Which tells me I have one bridge named br0

Code:

```
root@Main:~# brctl showstp br0 br0
```

```
bridge id          8000.xxxxxxxxxxxx
designated root    8000.xxxxxxxxxxxx1
root port         3
max age           20.00
hello time        2.00
forward delay     1.00
ageing time       300.00
hello timer       0.00
topology change timer 0.00
flags             TOPOLOGY_CHANGE_DETECTED
path cost         100
bridge max age    20.00
bridge hello time 2.00
bridge forward delay 1.00
tcn timer         1.79
gc timer          0.83
```

vlan1 (1)

```
port id           8001
designated root    8000.xxxxxxxxxxxx1
designated bridge  8000.xxxxxxxxxxxx
designated port    8001
designated cost    100
state             forwarding
path cost         100
message age timer 0.00
forward delay timer 0.00
hold timer        0.83
flags
```

eth1 (2)

```
port id           8002
designated root    8000.xxxxxxxxxxxx1
state             forwarding
path cost         100
```

Spanning_Tree_Protocol

```
designated bridge      8000.xxxxxxxxxxxxxx  message age timer      0.00
designated port        8002                  forward delay timer     0.00
designated cost        100                   hold timer              0.83
flags
```

wds0.1 (3)

```
port id               8003                  state                   forwarding
designated root        8000.xxxxxxxxxxxxxx1 path cost                100
designated bridge      8000.xxxxxxxxxxxxxx1 message age timer       19.85
designated port        8004                  forward delay timer     0.00
designated cost        0                     hold timer              0.00
flags
```

wds0.2 (4)

```
port id               8004                  state                   forwarding
designated root        8000.xxxxxxxxxxxxxx1 path cost                100
designated bridge      8000.xxxxxxxxxxxxxx message age timer       0.00
designated port        8004                  forward delay timer     0.00
designated cost        100                   hold timer              0.83
flags
```

root@Main:~#

Let's take a moment to interpret what we are seeing bridge id is the MAC of your local bridge interface with "8000." out in front (8000.xxxxxxxxxxxxxx) designated root is the MAC of the root bridge (spanning tree boss) with "8000." out in front (8000.xxxxxxxxxxxxxx1) root port is the port used to communicate with the root bridge (this case 3 which is wds0.1) path cost is the culmulative cost of the path to root (how far away you are from the root bridge) max age and bridge max age are the maximum age of the STP topology which is 20 seconds by default hello time and bridge hello time is the number of seconds between hello messages on the bridge forward delay and bridge forward delay are the time STP will wait before forwarding packets onto an existing bridge which gives the new device time to find out what is going on before she starts talking into a network ageing time is the length of time a MAC address is held in the forwarding table the forwarding table remembers the source MAC when a packet crosses, then it will store that MAC/port relation so when an packet comes in destined for the MAC the switch will then forward the packet back down the port where it is last known to be hello timer, tcn timer, topology change timer, and gc timer are all the current values in the live timers flags is again a live value containing the last active flag value (my example indicates a topology change has been detected)

Looking at the remaining info are the port names(vlan1, eth1, wds0.1, wds0.2), physical port (number following the port names "1, 2, 3, 4"), port ids (8001-8004), state (forwarding), path cost (all ethernet ports are default 100, we will revisit this in a moment) , and a bunch of timers and stuff.

The root bridge appears to be my Mid Router and the path costs are default all the way around.

We would repeat the above procedure on the other two devices to ensure they all believed the root was the Mid Router and that all the path costs were the same. in my case they were. So STP should have calculated a distance to the root bridge via link3 to be 200 (path cost of link3 added to the path cost of link 2). Which it did not do in this case, because I initially had link3 disabled on the Media router. (so it would work)

Spanning_Tree_Protocol

first thing I wanted to do and I suggest for all involved. Bring the root bridge to the WAN router, Main Router, in this case the newest and most powerful device ASUS RT-N16

By default, DDWRT assigns a root bridge priority of 100, I want my Main Router to be root so I influence this decision by lowering the Main Routers bridge priority to 99. NOTE; real switch vendors set their default bridge priority much higher than this, cisco and brocade uses 32768 out of the box. There are some other out of the box default settings that DDWRT may want to review for compatibility with industry standard commercial hardware. Priority is one of the "needs to play nice" ones where the default 100 will make the DDWRT device assume root bridge from an existing STP domain.

http://www.brocade.com/support/Product_Manuals/ServerIron_SwitchRouterGuide/STP.pdf

<http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.1/8aew/configuration/guide/spantree.html#wp1020334>

Code:

```
root@Main:~# brctl setbridgeprio br0 99
```

Code:

```
root@Main:~# brctl showstp br0 br0
```

```
bridge id          0063.xxxxxxxxxxxxxx
designated root    0063.xxxxxxxxxxxxxx
root port          0
max age           20.00
hello time        2.00
forward delay     1.00
ageing time       300.00
hello timer       1.81
topology change timer 0.00
flags
path cost         0
bridge max age   20.00
bridge hello time 2.00
bridge forward delay 1.00
tcn timer         0.00
gc timer          0.81
```

vlan1 (1)

```
port id           8001
designated root    0063.xxxxxxxxxxxxxx
designated bridge  0063.xxxxxxxxxxxxxx
designated port    8001
designated cost    0
flags
state             forwarding
path cost         100
message age timer 0.00
forward delay timer 0.00
hold timer        0.81
```

eth1 (2)

```
port id           8002
designated root    0063.xxxxxxxxxxxxxx
designated bridge  0063.xxxxxxxxxxxxxx
designated port    8002
designated cost    0
flags
state             forwarding
path cost         100
message age timer 0.00
forward delay timer 0.00
hold timer        0.81
```

wds0.1 (3)

Spanning_Tree_Protocol

```
port id          8003          state          forwarding
designated root  0063.xxxxxxxxxxxx path cost      100
designated bridge 0063.xxxxxxxxxxxx message age timer 0.00
designated port   8003          forward delay timer 0.00
designated cost   0             hold timer     0.81
flags
```

wds0.2 (4)

```
port id          8004          state          forwarding
designated root  0063.xxxxxxxxxxxx path cost      100
designated bridge 0063.xxxxxxxxxxxx message age timer 0.00
designated port   8004          forward delay timer 0.00
designated cost   0             hold timer     0.81
flags
```

root@Main:~#

root@Mid:~# brctl showstp br0 br0

```
bridge id          8000.xxxxxxxxxxxx1
designated root    0063.xxxxxxxxxxxx
root port         4
max age           20.00          path cost      100
hello time        2.00           bridge max age 20.00
forward delay     1.00           bridge hello time 2.00
ageing time       300.00         bridge forward delay 1.00
hello timer       0.00           tcn timer      0.00
topology change timer 0.00         gc timer       1.24
flags
```

eth0 (1)

```
port id          8001          state          forwarding
designated root  0063.xxxxxxxxxxxx path cost      100
designated bridge 8000.xxxxxxxxxxxx1 message age timer 0.00
designated port   8001          forward delay timer 0.00
designated cost   100          hold timer     0.23
flags
```

vlan1 (2)

```
port id          8002          state          forwarding
designated root  0063.xxxxxxxxxxxx path cost      100
designated bridge 8000.xxxxxxxxxxxx1 message age timer 0.00
designated port   8002          forward delay timer 0.00
designated cost   100          hold timer     0.23
flags
```

wds0.1 (3)

```
port id          8003          state          forwarding
designated root  0063.xxxxxxxxxxxx path cost      100
designated bridge 8000.xxxxxxxxxxxx1 message age timer 0.00
designated port   8003          forward delay timer 0.00
```


Spanning_Tree_Protocol

```
designated cost      100                hold timer          0.23
flags
```

wds0.2 (4)

```
port id             8004                state               forwarding
designated root     0063.xxxxxxxxxxxx  path cost           100
designated bridge   0063.xxxxxxxxxxxx  message age timer   19.20
designated port     8003                forward delay timer 0.00
designated cost     0                   hold timer          0.00
flags
```

root@Mid:~#

root@Media:~# brctl showstp br0 br0

```
bridge id          8000.xxxxxxxxxxxx2
designated root    0063.xxxxxxxxxxxx
root port         3                   path cost           200
max age           20.00              bridge max age      20.00
hello time        2.00               bridge hello time   2.00
forward delay     1.00               bridge forward delay 1.00
ageing time       300.00
hello timer       0.00               tcn timer            0.12
topology change timer 0.00              gc timer             1.08
flags             TOPOLOGY_CHANGE_DETECTED
```

eth0 (1)

```
port id             8001                state               forwarding
designated root     0063.xxxxxxxxxxxx  path cost           100
designated bridge   8000.xxxxxxxxxxxx2 message age timer    0.00
designated port     8001                forward delay timer 0.00
designated cost     200                 hold timer          1.08
flags
```

vlan1 (2)

```
port id             8002                state               forwarding
designated root     0063.xxxxxxxxxxxx  path cost           100
designated bridge   8000.xxxxxxxxxxxx2 message age timer    0.00
designated port     8002                forward delay timer 0.00
designated cost     200                 hold timer          1.08
flags
```

wds0.1 (3)

```
port id             8003                state               forwarding
designated root     0063.xxxxxxxxxxxx  path cost           100
designated bridge   8000.xxxxxxxxxxxx1 message age timer    19.36
designated port     8003                forward delay timer 0.00
designated cost     100                 hold timer          0.00
flags
```

root@Media:~#

Spanning_Tree_Protocol

And we can now see that the ASUS xxxxxxxxxxxx is the designated root across all three devices and the path cost to get to the root from the Media router is 200 (link1 plus link2)

now I want link3 to be less preferred so I want to make it's cost higher than the natural 100 (cost of link1 directly back to the root) but first i have to figure out which link is which.. hmm. that is stored in the nvram variables

so in Main

Code:

```
root@Main:~# nvram show | grep wds[23]_ | sort w10_wds2_desc=Mid w10_wds2_enable=3
w10_wds2_hwaddr=xx:xx:xx:xx:xx:x1 w10_wds2_if=wds0.1 w10_wds2_ipaddr= w10_wds2_netmask=
w10_wds2_ospf= w10_wds3_desc=Media w10_wds3_enable=3 w10_wds3_hwaddr=xx:xx:xx:xx:xx:x2
w10_wds3_if=wds0.2 w10_wds3_ipaddr= w10_wds3_netmask= w10_wds3_ospf=
```

I see that the one I am interested in is wds0.2 so my command is

Code:

```
root@Main:~#brctl setpathcost br0 wds0.2 300 root@Main:~# brctl showstp br0 br0
```

```
bridge id          0063.xxxxxxxxxxxxxx
designated root     0063.xxxxxxxxxxxxxx
root port          0
max age            20.00
hello time         2.00
forward delay      1.00
ageing time        300.00
hello timer        0.69
topology change timer 0.00
flags
path cost          0
bridge max age    20.00
bridge hello time 2.00
bridge forward delay 1.00
tcn timer         0.00
gc timer          0.69
```

vlan1 (1)

```
port id            8001
designated root     0063.xxxxxxxxxxxxxx
designated bridge   0063.xxxxxxxxxxxxxx
designated port     8001
designated cost     0
flags
state              forwarding
path cost          100
message age timer  0.00
forward delay timer 0.00
hold timer         0.00
```

eth1 (2)

```
port id            8002
designated root     0063.xxxxxxxxxxxxxx
designated bridge   0063.xxxxxxxxxxxxxx
state              forwarding
path cost          100
message age timer  0.00
```

Spanning_Tree_Protocol

```
designated port      8002                forward delay timer  0.00
designated cost      0                    hold timer           0.00
flags
```

wds0.1 (3)

```
port id             8003                state                forwarding
designated root      0063.xxxxxxxxxxxx  path cost            100
designated bridge    0063.xxxxxxxxxxxx  message age timer    0.00
designated port      8003                forward delay timer  0.00
designated cost      0                    hold timer           0.00
flags
```

wds0.2 (4)

```
port id             8004                state                forwarding
designated root      0063.xxxxxxxxxxxx  path cost            300
designated bridge    0063.xxxxxxxxxxxx  message age timer    0.00
designated port      8004                forward delay timer  0.00
designated cost      0                    hold timer           0.00
flags
```

root@Main:~#

now I have to enable the port in the Media router so I can see it's port name so we do that in the gui repeat the

Code:

```
root@Media:~# nvram show | grep wds[12]_ | sort w10_wds1_desc=Main w10_wds1_enable=3
w10_wds1_hwaddr=xx:xx:xx:xx:xx:xx w10_wds1_if=wds0.1 w10_wds1_ipaddr= w10_wds1_netmask=
w10_wds1_ospf= w10_wds2_desc=Mid w10_wds2_enable=3 w10_wds2_hwaddr=xx:xx:xx:xx:xx:x1
w10_wds2_if=wds0.2 w10_wds2_ipaddr= w10_wds2_netmask= w10_wds2_ospf=
```

and we see it is wds0.1 now. order is important this value will change based on where on the list it lies be careful when adding and disabling values on the webpage, these will move about and may cause unintended side effects.

Code:

```
root@Media:~#brctl setpathcost br0 wds0.1 300 root@Media:~# brctl showstp br0 br0
```

```
bridge id          8000.xxxxxxxxxxxx2
designated root     0063.xxxxxxxxxxxx
root port          4                path cost         200
max age            20.00           bridge max age    20.00
hello time         2.00            bridge hello time 2.00
forward delay      1.00            bridge forward delay 1.00
```

Spanning_Tree_Protocol

```
ageing time      300.00
hello timer      0.00      tcn timer        0.04
topology change timer  0.00      gc timer        105.30
flags
```

eth0 (1)

```
port id      8001      state      forwarding
designated root 0063.xxxxxxxxxxxx path cost    100
designated bridge 8000.xxxxxxxxxxxx2 message age timer 0.00
designated port 8001      forward delay timer 0.00
designated cost 200      hold timer    0.66
flags
```

vlan1 (2)

```
port id      8002      state      forwarding
designated root 0063.xxxxxxxxxxxx path cost    100
designated bridge 8000.xxxxxxxxxxxx2 message age timer 0.00
designated port 8002      forward delay timer 0.00
designated cost 200      hold timer    0.66
flags
```

wds0.1 (3)

```
port id      8003      state      blocking
designated root 0063.xxxxxxxxxxxx path cost    300
designated bridge 0063.xxxxxxxxxxxx message age timer 3.41
designated port 8004      forward delay timer 0.00
designated cost 0      hold timer    0.00
flags
```

wds0.2 (4)

```
port id      8004      state      forwarding
designated root 0063.xxxxxxxxxxxx path cost    100
designated bridge 8000.xxxxxxxxxxxx1 message age timer 2.40
designated port 8003      forward delay timer 0.00
designated cost 100      hold timer    0.00
flags
```

root@Media:~#

Some References <http://www.faqs.org/docs/Linux-HOWTO/BRIDGE-STP-HOWTO.html#STP>
<http://linux.die.net/man/8/brctl> <http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge>

setageing <bridge> <time> set ageing time (Default 300 seconds) Sets the aging time. The aging time is the number of seconds a MAC-address will be kept in the forwarding database after having received a packet from this MAC address. The entries in the forwarding database are periodically timed out to ensure they won't stay around forever. Normally there should be no need to modify this parameter. sets the ethernet (MAC) address ageing time, in seconds. After <time> seconds of not having seen a frame coming from a certain address, the bridge will time out (delete) that address from the Forwarding DataBase (fdb).

Spanning_Tree_Protocol

`setbridgeprio <bridge> <prio>` set bridge priority (default 100, lower is more preferred, industry standard default is 32768) Sets the bridge's relative priority. The bridge with the lowest priority will be elected as the root bridge. The root bridge is the "central" bridge in the spanning tree. sets the bridge's priority to <priority>. The priority value is an unsigned 16-bit quantity (a number between 0 and 65535), and has no dimension. Lower priority values are 'better'. The bridge with the lowest priority will be elected 'root bridge'.

`setfd <bridge> <time>` set bridge forward delay (default is 1, should be 4) Sets the forwarding delay time. The forwarding delay time is the time spent in each of the Listening and Learning states before the Forwarding state is entered. sets the bridge's 'bridge forward delay' to <time> seconds

`sethello <bridge> <time>` set hello time (default is 2, can be 1) Sets the hello time. Every (this number) seconds, a hello packet is sent out by the Root Bridge and the Designated Bridges. Hello packets are used to communicate information about the topology throughout the entire Bridged Local Area Network. sets the bridge's 'bridge hello time' to <time> seconds

`setmaxage <bridge> <time>` set max message age (default is 20, w/hello at 1, can be 4) Sets the maximum message age. If the last seen (received) hello packet is more than this number of seconds old, the bridge in question will start the takeover procedure in attempt to become the Root Bridge itself. sets the bridge's 'maximum message age' to <time> seconds

`setpathcost <bridge> <port> <cost>` set path cost (default is 100, higher is less preferred) Sets the cost of receiving (or sending, I'm not sure) a packet on this interface. Faster interfaces should have lower path costs. These values are used in the computation of the minimal spanning tree. Paths with lower costs are likelier to be used in the spanning tree than high-cost paths (As an example, think of a gigabit line with a 100Mbit or 10Mbit line as a backup line. You don't want the 10/100Mbit line to become the primary line there.) The Linux implementation currently sets the path cost of all eth* interfaces to 100, the nominal cost for a 10Mbit connection. There is unfortunately no easy way to discern 10Mbit from 100Mbit from 1Gbit Ethernet cards, so the bridge cannot use the real interface speed. sets the port cost of the port <port> to <cost>. This is a dimensionless metric.

`setportprio <bridge> <port> <prio>` set port priority sets the port <port>'s priority to <priority>. The priority value is an unsigned 8-bit quantity (a number between 0 and 255), and has no dimension. This metric is used in the designated port and root port selection algorithms. You use this one to distinguish between two equal cost ports.

To get mine to be stable and responsive to topology changes I tweaked it as follows I set the Hello to every one second vs two seconds, to make it detect a topology change faster I changed the maximum age from twenty seconds to four seconds to make it refresh the topology more often. I set the Forwarding delay to four seconds vs one second, to make it wait longer before participating in a STP domain

in the startup commands i put the following Code:

Main

```
/usr/sbin/brctl sethello br0 1 /usr/sbin/brctl setmaxage br0 4 /usr/sbin/brctl setfd br0 4 /usr/sbin/brctl setbridgeprio br0 99 /usr/sbin/brctl setpathcost br0 wds0.2 300
```

Code:

Spanning_Tree_Protocol

Mid

```
/usr/sbin/brctl sethello br0 1 /usr/sbin/brctl setmaxage br0 4 /usr/sbin/brctl setfd br0 4
```

Code:

Media

```
/usr/sbin/brctl sethello br0 1 /usr/sbin/brctl setmaxage br0 4 /usr/sbin/brctl setfd br0 4 /usr/sbin/brctl  
setpathcost br0 wds0.1 300
```

Now she is up and stable to test i took a wireless client from on the Media router started a streaming ping to google

removed power from the Mid router she took about 30 seconds to switch paths

restored power to the Mid router long pause while she booted while the ping was successful then ping stopped for about 2 mins then restored via the Mid router

removed power again from Mid router ping stopped for about 2 mins then restored over link3

restored power again to the Mid router long pause while she booted while the ping was successful then ping stopped for about 30 seconds then restored via the Mid router

not exactly a scientific test but she sufficed for my purposes. I can fail to an alternate path in less than 3 minutes if one of my routers go down. Which is really good for consumer applications.

External Links

http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/sw_ntman/cwsi2/cwsiug2/vlan2/stpapp.htm