INR Lab Assignment
Spanning Tree Protocol and 802.1Q*

Mick Pouw

Arno Bakker
(Arno.Bakker@os3.nl)

Feedback deadline:
November 24, 2017 10:00 CET

Abstract

In the first part of this lab session you will be looking at the behavior of the Spanning Tree Protocol. After that you will get some hands on experience with configuring VLANs and static routes using Linux tools.

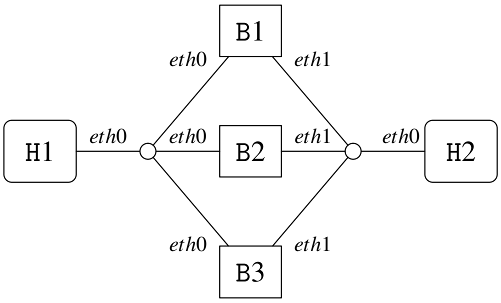

Figure 1: STP Network

Preparation

Before you start working, make sure you download **old** version of the **ogopogo** network scripts. The link will be provided below. Also inspect the man page of **brctl** and the theory behind the STP protocol.

Task 1. Create a new VM running the Ubuntu14.04 Trusty operating system. Give it 1GB of RAM and a public ip. Make sure you can log in using your ssh keys.

Task 2. Install the following packages: `python git user-mode-linux uml-utilities debootstrap vde2 screen`. Afterwards, clone the **old** git repository: `https://github.com/TeamOS3/ogopogo.git`.

*Based on earlier work by C. Dumitru and Jeroen van der Ham. Version November 10, 2017.

Task 3. Set the values of A,B,X,Y in the `rc.local` script to a mod 10, b mod 10, x and y, respectively, as derived in the previous lab. These variables are used to generate the MAC addresses in this lab. Run the `build_rootfs.sh`[1], which creates a disk image, and copy this to the `/tmp` directory.

Task 4. Create the config file that starts the network using the `bridge.cfg` example. Keep the sniffer there. Run `python ogo.py start bridge.cfg` to start the environment. The sniffer should leave pcap files in its home directory, which is the `/tmp` directory by default .

Witnessing the startup

Log the startup and running of the protocol for a while using the sniffers. Once you have the logs, they must be put into the correct order and correlated.

Hint: To speed up the analysis process you should use Wireshark.

Task 5. Create a diagram of the networks showing the state of all the bridge ports and the root bridge. *Hint:* ***brctl showstp*** *(not described in manual page).*

Task 6. Describe in detail (i.e. your own words, not verbatim dumps) what packets are sent, when and why. Explain **how** the root bridge was elected. Only mention the relevant packets. Upload the raw dump file (as generated by the sniffer) to your wiki, and provide a link in your log.

Task 7. a. What parameters are used in the BPDU packets?

b. What is the role of each parameter?

c. What are they set to?

Creating problems

Next you will disturb the network.

Task 8. What happens if you shutdown the root bridge? Describe all the events that take place from the moment the bridge goes down until the network has converged again.

Task 9. Restore the original situation. Then start a ping from H1 to H2. Next, on a bridge machine that has disabled ports, bring down the bridge, disable STP and then bring up the bridge again. Describe what happens mentioning the state of the ports on all bridges. After how much time do the other bridges notice your disruptive actions? What is the **downtime** experienced by the ping session? Based on the STP protocol variables explain this downtime.

Task 10. Enable STP again on all bridges. Does the network come back to the initial state? Why?

Task 11. **Bonus**: Get STP to work with the pogo toolset.

[1] You can alter this script to install nano or vim. You will have to use **vi** if you don't.

802.1Q aka VLANs

Now return to the VM created in lab3 and pull the newest version of pogo. Creating VLANs under Linux is very easy thanks to the `iproute` package, which provides the `ip` command. Use

Task 12. What command do you need to type to add a VLAN id 5 to `eth0`?

Task 13. Start the vlan.conf configuration in the pogo repository and configure each host manually, as follows:

- H1: `eth0.10,10.0.10.1/24 ; eth0.30,10.0.30.1/24`
- H2: `eth0.10,10.0.10.2/24 ; eth0.20,10.0.20.2/24 ; eth0.30,10.0.30.2/24`
- B1: `bridge0` containing `eth0.10, eth1.10`
- B2: `eth1.20,10.0.20.1/24`
- B3: `bridge0` containing `eth0.30, eth1.30`

Note: `ethX.Y,Z/T` means ethX with VLAN id Y configured having IP address Z with subnetmask T. *Hint: you might want to write down the commands in a text file first, to make sure you got everything right.*

Task 14. Once you have have configured everything perform the following ping tests:

- from 10.0.10.1 to 10.0.10.2
- from 10.0.30.1 to 10.0.30.2

Task 15. Explain the structure of a Ethernet frame that contains an ICMP echo request with source 10.0.20.1 captured by the sniffer on the righthand side of the network (between H2 and B2). Explain all the VLAN related fields in detail.

Next you will configure H2 to act as a “router-on-a-stick”.

Task 16. Add static routes on H1, H2, B2 to allow B2 to reach H1 over both VLANs. Display the routing tables on H1, H2 and B2 and the output of ping from B2 to 10.0.10.1 and 10.0.30.1.

Task 17. What is the maximum number of VLAN IDs active on a network segment? Be precise!

Task 18. Draw a network diagram that depicts the configuration. Make sure the VLANs are clearly marked.