Laboratory 5 Switching and Bridging

Introduction

This lab will introduce basic elements about Ethernet seitches and bridges:

- Packet forwarding with transparent bridges (switches)
- Spanning Tree Protocol (STP)
- Virtual LANs (VLAN)
-

A transparent bridge is a common type of bridge that observes incoming network traffic to identify media access control (MAC) addresses. These bridges operate in a way that is transparent to all the network's connected hosts. Transparent bridges maintain a list of MAC addresses, as do routers, based on all the received frames' source data-link MAC addresses. These tables are used for address look-up while forwarding a frame.

Transparent bridges save and maintain the source-route addresses of incoming frames by listening to all the connected bridges and hosts. They use a transparent bridging algorithm to a accomplish this. The algorithm has five parts:

- Learning
- Flooding
- Filtering
- Forwarding
- Avoiding loops

For example, consider three hosts, A, B and C, and a bridge with three ports. Host A is connected to Bridge Port 1, Host B is connected to Bridge Port 2 and Host C is connected to Bridge Port 3. Host A sends a frame to the bridge that is addressed to Host B. The bridge checks the frame's source address and creates an address and port number entry for Host A in its forwarding table. The bridge then examines the frame's destination address, but does not find it in its forwarding table. As a result, the bridge sends the frame to all the other ports (2 and 3). This is called flooding. The frame is then received by Host B and Host C, which also check the destination address. Host B recognizes a destination address match and sends a response to Host A.

On the return path, the bridge adds an address and port number entry for Host B to its forwarding table. The bridge already has Host A's address in its forwarding table, so it forwards the response only to Port 1. In this way, none of the Port 3 hosts are burdened with response requirements. Through this process, two-way communication between Host A and Host B is facilitated without the need for further flooding [1].

The Spanning Tree Protocol (STP) is a network protocol that builds a loop-free logical topology for Ethernet networks. The basic function of STP is to prevent bridge loops and the broadcast packets that results from them. Spanning tree also allows a network design to include backup links to provide fault tolerance if an active link fails [2].

Root bridge: is the bridge selected as the root of the spanning tree.

When STP is running the ports of the bridge/switch can have three states:

- Root port
- Designated port
- Blocking port

Root port: After the Root Bridge is identified, all other switches determine the quickest path from themselves to the Root Bridge. The port used to send packets to the Root Bridge is selected as Root Port.

Designated port: The port on the next closest switch (neighbor switch) to the Root Bridge that is facing the reference switch is called the Designated Port.

Blocking port**:** All the ports of the bridge that are neither Root Port nor Designated Port are put by STP into Blocking state, which mean they are Blocking Ports.

A virtual LAN (VLAN) is any broadcast domain that is partitioned and isolated in a computer network at the data link layer (OSI layer 2). LAN is the abbreviation for local area network and in this context virtual refers to a physical object recreated and altered by additional logic. VLANs work by applying tags to network frames and handling these tags in networking systems – creating the appearance and functionality of network traffic that is physically on a single network but acts as if it is split between separate networks. In this way, VLANs can keep network applications separate despite being connected to the same physical network, and without requiring multiple sets of cabling and networking devices to be deployed [3].

Part 2: Experimental part

Remark:

Do not forget to set the Netkit environment variables and check your configuration:

- `export NETKIT_HOME=~/netkit`
- `export MANPATH=:$NETKIT_HOME/man`
- `export PATH=$NETKIT_HOME/bin:$PATH`

Check your configuration:

- `./check_configuration.sh`

Study of Transparent Bridges

In this part a topology consisting of two Ethernet switches/bridges (the terms switch and bridge will have the same meaning throughout this lab) and four computers will be implemented as in Figure 1. In Figure 1, for each interface the last two octets of the MAC address are specified (the other four octets are 0).

Remark: *Switches, like all the devices emulated with Netkit, are implemented with Linux containers. They are basically Linux machines. By default, a Linux machine with multiple interfaces acts as a router. To act as a switch the bridge function must be activated on the Linux machine and its network interfaces must be attached to this bridge.*

1. Network topology

The network topology is presented in Figure 1.

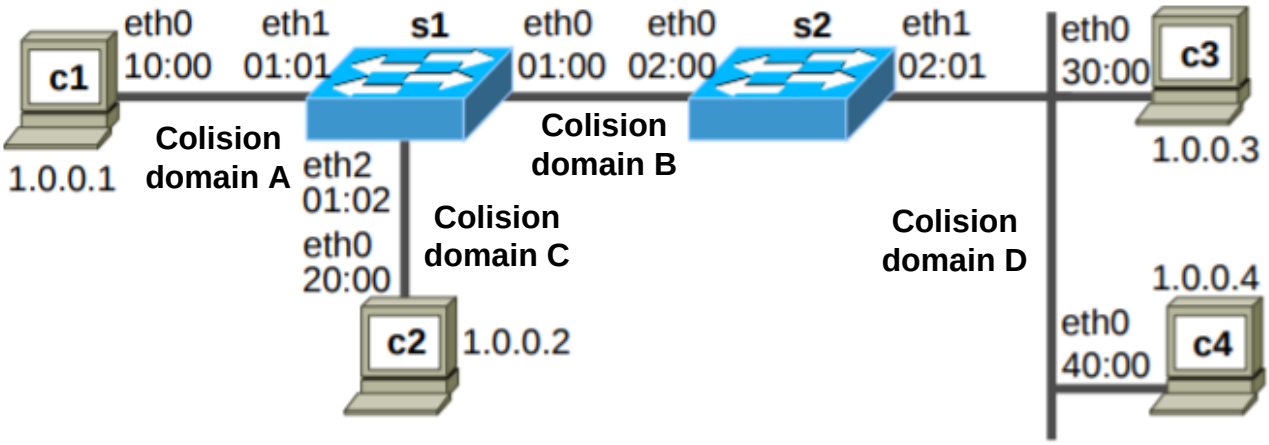

Figure 1: Network topology

For the computers **`c1`, …, `c4`,** the `.startup` file should contain the configuration of the IP and MAC addresses:

 # computer c1

- `ifconfig eth0 1.0.0.1 up`
- `ifconfig eth0 hw ether 00:00:00:00:10:00`

For the switches `s1, s2`, the `.startup` file should contain the configuration of the MAC addresses:

 # switch s1

- `ifconfig eth0 up # s1 port 0`
- `ifconfig eth0 hw ether 00:00:00:00:01:00`
- `ifconfig eth1 up # s1 port 1`
- `ifconfig eth1 hw ether 00:00:00:00:01:01`
- `ifconfig eth2 up # s1 port 2`
- `ifconfig eth2 hw ether 00:00:00:00:01:02`

2. Network initialization

Start the Netkit lab.

Check the initial configuration of the network devices using the commands:

- `ifconfig # to list the state of the interfaces`
- `brctl show #to list the state of the bridges`

Check that the IP and MAC addresses are correctly configured. Is the bridge function activated on `s1` and `s2` devices?

3. Bridge initialization

Configure the bridges s1 and s2. The commands to activate bridge function on s1 are:

- `brctl addbr br0 #add bridge br0`
- `brctl addif br0 eth0 #attaches interface eth0 to br0`
- `brctl addif br0 eth1 #attaches interface eth1 to br0`
- `ifconfig br0 up #activates bridge br0`

Examine the state of the s1 and s2 bridges with the commands:

- `brctl show`
- `brctl showstp br0 # shows the state of the br0 interfaces according to STP`
- `brctl showmacs br0 # shows the MAC address table`

***Q1:** Is the bridge activated on `s1` and `s2`?*

***Q2:** Are the interfaces attached to the bridge?*

Depending on the configuration, a machine traffic even if not solicited (e.g., broadcast packets)

- the source address tables of switch1 and switch2 may already contain non-local entries
- hard to prevent

Ports (interfaces) are numbered according to the 802.1d standard (STP)

- the correspondence between kernel interface numbering (ethX) and 802.1d numbering can be obtained by using `brctl showstp`

4. Packet forwarding

Capture the traffic on eth1 interfaces for `s1` and `s2` with the command:

- `s1# tcpdump -i eth1 -e -q`
- `s2# tcpdump -i eth1 -e -q`

Initiate the communication between `c3` and `c4` using `ping` command:

- `c3# ping -c 3 1.0.0.4`

Q3: *Explain why for the first pair of messages ICMP Echo Request/Reply the delay is bigger than the delay for the next ICMP Echo Request/Reply messages.*

Repeat the experiment for the communication between `c1` and `c4`.

Study of the STP protocol

In this section, the Spanning Tree Protocol, which is used to prevent loops in a network topology consisting of bridges with redundant links, will be analyzed.

1. Network topology

The network topology is presented in Figure 2.

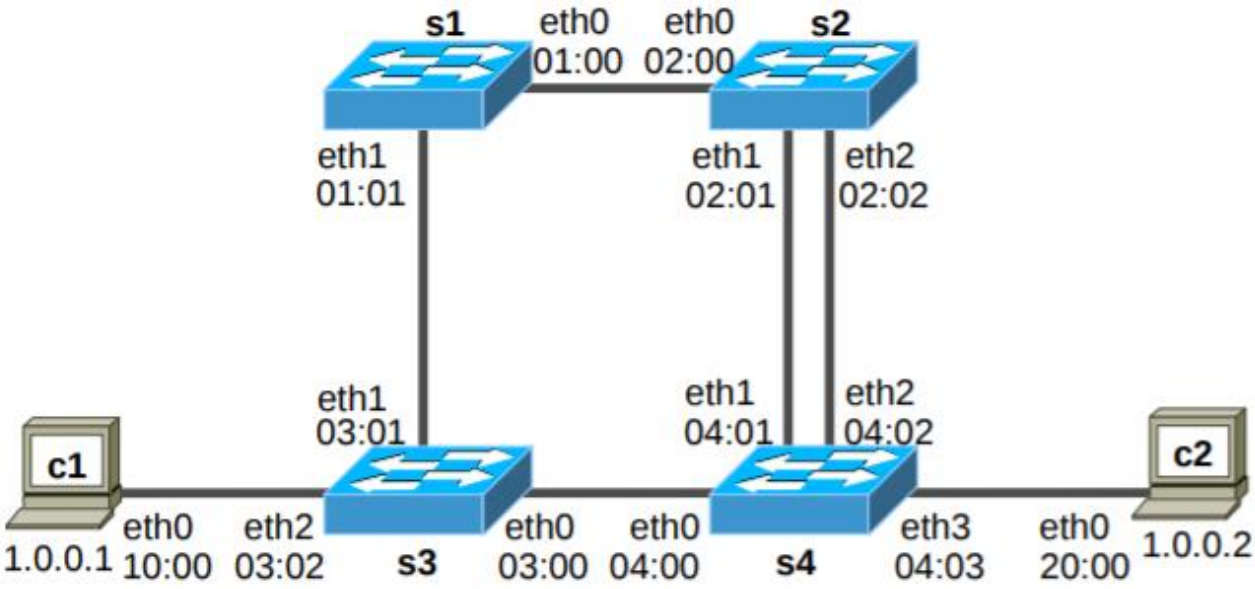

Figure 2: Network topology for STP study

Configure the hosts and switches as in the first part of the lab. On each switch activate the STP protocol with the command:

- `brctl stp br0 on`

2. Network initialization

Examine the initial configuration using commands:

- `ifconfig # to list the state of the interfaces`
- `brctl show #to list the state of the bridges`

***Q4**: Check if the interfaces are up, the IP and MAC addresses are correct, the bridge function is activated on all switches, and the interfaces are attached to the bridges.*

Examine the state of bridges and identify the spanning tree:

- `brctl showstp br0`

***Q5**: Which is the state of each port for every switch?*

3. STP protocol

The Root Bridge is sending STP Configuration BPDU (Bridge Protocol Data Unit) packets periodically. The other bridges resend these packets on the Designated Ports such as they are propagated to all the bridges. Thus, the bridges can check that the spanning tree is functional and to detect link failures.

Capture the frames sent by `s1` on interface `eth0` and by bridge `s2` on the interface `eth1`:

- `s1# tcpdump –i eth0 –s 0 –nev`
- `s2# tcpdump –i eth1 –s 0 –nev`

***Q6**: Analyze the frames captured and identify the differences between those sent by `s1` and those sent by s2. Repeat the measurements for other bridges and ports*.

***Remark:** The MAC address of the STP BPDU frames is 01:80:C2:00:00:00. This is a multicast address reserved for STP.*

5. Packet forwarding

Capture the traffic for `eth1` interface on `s3` and for `eth0` interface on `s2` with the command:

- `s3# tcpdump –i eth1 –s 0 –nev arp or ip`
- `s4# tcpdump –i eth0 –s 0 –nev arp or ip`

Initiate the communication between `c1` and `c2` using `ping` command:

- `c1# ping –c 3 1.0.0.2`

Stop capturing the frames and display the address table on each bridge with the command:

- `brctl showmacs br0`

***Q7**: Analyze the captured traffic. Identify the ports used to send the ARP and ICMP packets. Explain the changes in the bridges' address tables (initially the address tables contain only the MAC addresses of the bridges).*

6. Spanning tree reconfiguration as a result of a port failure

eth1: 10

Capture the frames on `s1` interface `eth1`, on `s2` interface `eth1`, and on `s4` interface `eth0`.

- `s1# tcpdump -i eth1 -s 0 -w /hostlab/s1e1.cap`
- `s2# tcpdump -i eth1 -s 0 -w /hostlab/s2e1.cap`
- `s4# tcpdump -i eth0 -s 0 -w /hostlab/s4e0.cap`

Initiate the communication between `c1` and `c2` using `ping` command:

- `c1# ping 1.0.0.2`

Shut down the `eth0` interface on `s2`:

- `s2# ifconfig eth0 down`

***Q8**: After the interface has been closed the communication between `c1` and `c2` is interrupted. Wait until the communication resumes. Stop `ping` command and packet capturing.*

***Q9**: Examine the state of bridges and identify the spanning tree:*

- `brctl showstp br0`

Compare the new spanning tree with the initial one.

Vizualize the captured traffic with Wireshark using the command on the terminal:

- `wireshark – r s1e1.cap`
- `wireshark -r s2e1.cap`
- `wireshark -r s4e0.cap`

***Q10**: What type of BPDU frames `s2` starts transmitting after `eth0` is shutted down?*

***Q11:** What is the meaning of the BPDU TCN frames sent by `s4` and `s3`?*

***Q12:** How does `s3` react on receiving the BPDU TCN frames?*

***Q13:** How are the ICMP packets send through the network?*

***Q13:** How long the communication is interrupted?*

Study of the virtual private LANs (VLAN)

VLANs are used to divide the network of an organization in multiple subnets. VLAN are implemented on Ethernet bridges running the IEEE 802.1Q protocols. VLANS can be interconnected using routers.

1. Network topology

The network topology is presented in Figure 3. The local network consists of two VLANs, `vlan1` (red) and `vlan 2` (blue), which correspond to the subnets 1.0.1.0/24 and 1.0.2.0/24. These subnets will be interconnected through `r1` router.

Configure the hosts and switches as in the first part of the lab.

2. Network initialization

Examine the initial configuration using commands:

- `ifconfig # to list the state of the interfaces`
- `route #to list the routing tables on c1, …, c5, r1`

***Q14**: Check if the interfaces are up, the IP and MAC addresses are correct, the bridge function is activated on all switches, and the interfaces are attached to the bridges.*

Examine the state of bridges:

- `brctl show`

***Q15**: Which is the state of each port for every switch?*

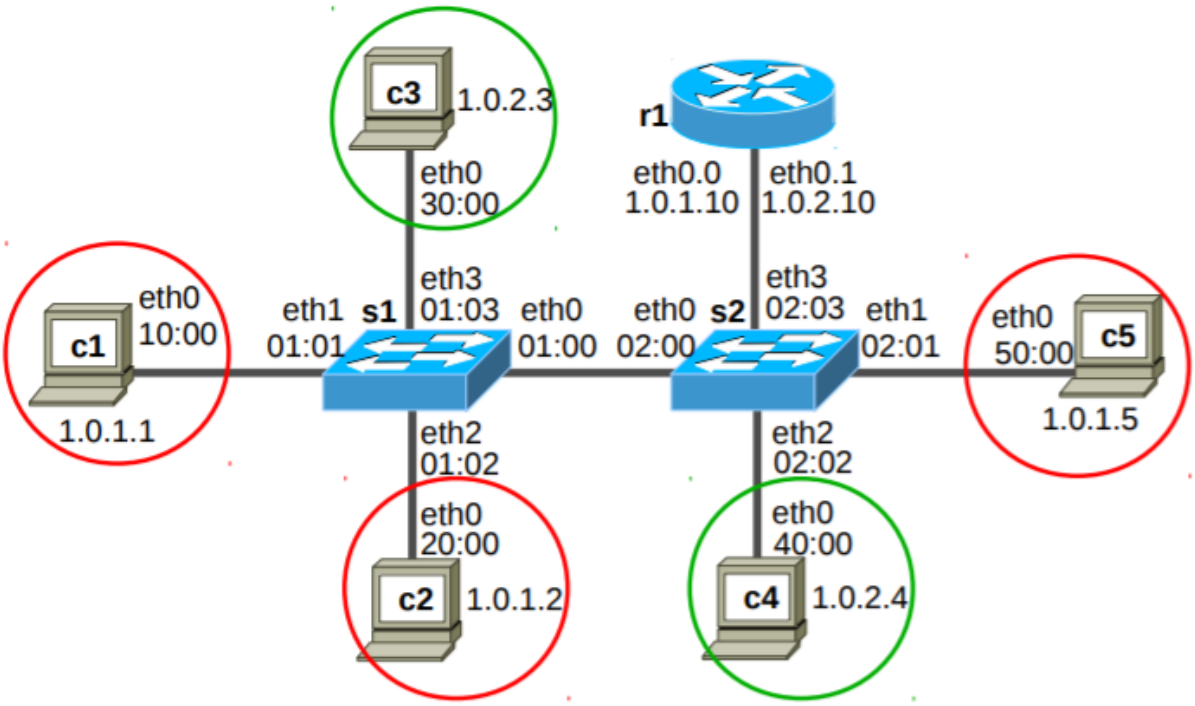

Figure 3: Network topology of a local network with two VLANs

3. VLAN Configuration

Configure `vlan1` and `vlan2` on each of `s1` and `s2` bridges. The commands for `s1` are presented below. Similar commands will be used on `s2`.

For the beginning, two bridges, `vlan1` and `vlan2`, will be created on `s1`, and the interfaces are attached to the corresponding VLAN.

- `s1# brctl addbr vlan1`
- `s1# brctl addif vlan1 eth1`
- `s1# brctl addif vlan1 eth2`
- `s1# ifconfig vlan1 up`
- `s1# brctl addbr vlan2`

- `s1# brctl addif vlan2 eth3`
- `s1# ifconfig vlan1 up`

The above commands are enough to configure VLAN on a single switch. Because the topology under study consists of two switches, and the VLANs extends on both, a connection between the VLANS on `s1` and `s2` must be configured. The commands for `s1` are presented below. Similar commands will be used on `s2`.

- `vconfig add eth0 1 # add virtual interface eth0.1`
- `vconfig add eth0 2 # add virtual interface eth0.2`
- `brctl addif vlan1 eth0.1 # add eth0.1 at vlan1`
- `brctl addif vlan2 eth0.2 # add eth0.2 at vlan2`
- `ifconfig eth0.1 up`
- `ifconfig eth0.2 up`

Because of these commands, on the link between `s1` and `s2`, the header of the Ethernet frames is extended with the 802.1Q header, which contains the label (VLAN ID - VID) of the VLAN the frame belongs.

Q16: *Examine the states of* `s1` and `s2` with the commands:

- `brctl show`
- `brctl showstp vlan1`
- `brctl showstp vlan2`

Are the bridges actives? Are all the interfaces connected to the bridges? Are all the ports in forwarding state?

4. Packet forwarding between hosts on the same VLAN

Capture the traffic on the link between `c1` and `s1` and on the link between `s1` and `s2` with the command:

- `s1# tcpdump -i eth1 -s 0 -w /host/s1e1vlan.cap`
- `s2# tcpdump -i eth0 -s 0 -w /hostlab/s2e0vlan.cap`

Initiate the communication between `c1` and `c5` using `ping` command:

- `c1# ping -c 3 1.0.1.5`

Initiate the communication between `c3` and `c4` using `ping` command:

- `c3# ping -c 3 1.0.2.4`

To illustrate that the computers located in different VLANs cannot communicate, temporarily modify the IP on `c3`, such as `c3` to be in the same subnet with `c1`. Test the communication between `c1` and `c3`.

- `c3# ifconfig eth0 1.0.1.3 netmask 255.255.255.0`
- `c3# ping 1.0.1.1`

Q17: *Does ping work? Why?*

Modify the IP address on `c3` at its initial value.

- `c3# ifconfig eth0 1.0.2.3 netmask 255.255.255.0`

End the capture of frames on the switches. Examine the traffic with Wireshark.

- `wireshark -r s1e1vlan.cap &`
- `wireshark -r s2e0vlan.cap &`

Q18: *Examine the encapsulation of frames on both links. Explain the role the transmission of VID plays.*

5. Packet forwarding between hosts on different VLANs

To interconnect `vlan1` and `vlan2`, router `r1` must be configured to route the packets between the two subnets. Also, static routes must be configured on c1, …, c5.

On router `r1` enter the following commands:

- `r1# ifconfig eth0 up`
- `vconfig add eth0 1`
- `vconfig add eth0 2`
- `ifconfig eth0.1 1.0.1.10 netmask 255.255.255.0 up`
- `ifconfig eth0.2 1.0.2.10 netmask 255.255.255.0 up`

On the switch `s2` configure the link with `r1` with the commands:

- `vconfig add eth3 1 # add virtual interface eth3.1`
- `vconfig add eth3 2 # add virtual interface eth3.2`
- `brctl addif vlan1 eth3.1 # add eth0.1 at vlan1`
- `brctl addif vlan2 eth3.2 # add eth0.2 at vlan2`
- `ifconfig eth3.1 up`
- `ifconfig eth3.2 up`

Configure the default gateway on the computers `c1, …, c5`:

- `c1:# route add default gw 1.0.1.10`

Remark: *Take care to configure the right gateway for each computer.*

Examine the router and the computers' configuration using commands:

- `ifconfig`
- `route #to list the routing tables on c1, …, c5, r1`

Initiate a communication between `c1` and `c3`, which are in different VLANs.

Q19: *Does ping work? Which is the route followed by the packets?*

Close the lab with `lcrash` command. Remove the lab folders created during the lab activity.

References:

1. https://www.techopedia.com/definition/3179/transparent-bridge
2. https://en.wikipedia.org/wiki/Spanning_Tree_Protocol
3. https://en.wikipedia.org/wiki/Virtual_LAN
4. http://wiki.netkit.org/netkit-labs/netkit-labs_advanced-topics/netkit-labs_bridging/netkit-lab_two-switches.pdf
5. http://discipline.elcom.pub.ro/apc/Lab/apc-l6-lan_v0_9.pdf