IT 304: Computer Networks

Lab # 5: Bridging in Local Area Networks (LANs) *

For the Week of September 09, 2013

Pre-lab preparation:

- Read the relevant portions of the reference book on the basic operation and functions of a bridge and the spanning tree protocol (IEEE 802.1d) used to compute the minimum spanning tree of a given network.
- For this lab we shall configure the Linux machine as a bridge. To accomplish this you will need to log in using the root id and password.

1. **Aim:** The purpose of this lab is to understand, basic operation of a bridge and operation of spanning tree protocol (STP).
2. **Bridge operation and STP:** A bridge is a layer two device that facilitates communication across different LANs. In essence a bridge creates one big extended LAN that is comprised of numerous LANs. The bridge operates promiscuously and creates a forwarding table that is used to forward frames from one LAn to another. As discussed in the class the presence of loops can create problems for the normal operation of a bridge. To overcome this the bridges run STP that essentially computes the minimum spanning tree for the given network topology.

   In the following simple network scenarios we shall understand the basic operation of bridges and the STP that is run on them.

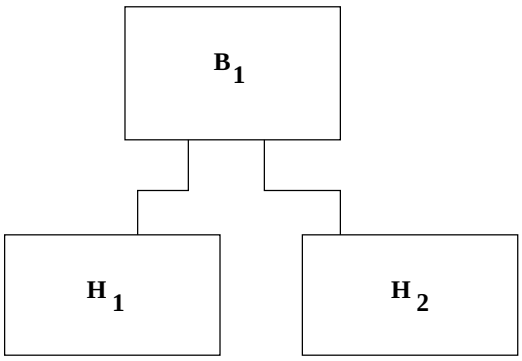

Figure 1: The figure above shows one of the network scenarios considered as part of this lab.

3. **Lab Scenarios:**

(a) Consider the lab scenario in Fig. 1 where we have two Linux machines that will act as hosts (denoted respectively as H_1 and H_2 respectively) connected to another Linux machine (denoted as B_1) that will be configured as a bridge. Configure ip addresses on one of the Ethernet interfaces available on the two hosts so that they are on two different networks (say `10.0.1.1` and `10.0.2.1` with network mask `255.255.255.0`). This can be accomplished on the Ethernet interface `eth0` as follows: `ifconfig eth0 10.0.1.1 netmask 255.255.255.0 up`. Ping each other to see if they can communicate.

(b) Configure the Ethernet interface `eth0` of B_1 in Fig. 1 with an ip address so that it belongs to the same network as that of H_1. Ping H_1 to see if it can be reached from B_1. We next configure it as a bridge and add the two Ethernet interfaces to it using the following commands:

`brctl addbr br0` (this command creates a bridge with id `br0`)

`brctl addif br0 eth0` (adds the Ethernet interface `eth0` to `br0`)

`brctl addif br0 eth1` (adds the Ethernet interface `eth1` to `br0`)

`ifconfig br0 up` (makes the bridge `br0` active)

`brctl show` (shows the details of bridge `br0` like bridge id, interfaces configured under the given bridge, etc.)

Try pinging H_1 from B_1 again and see if it can be reached. Explain why or why not?

`brctl showmacs br0` (can be used to inspect filter databases of `br0`; see if the bridge has learnt MAC addresses of the ethernet interfaces configured on the two hosts H_1 and H_2 respectively. Ensure that you even look at other fields of the various entries).

Try pinging H_2 from H_1 and vice-versa. Can the two hosts reach one another now? Explain why or why not?

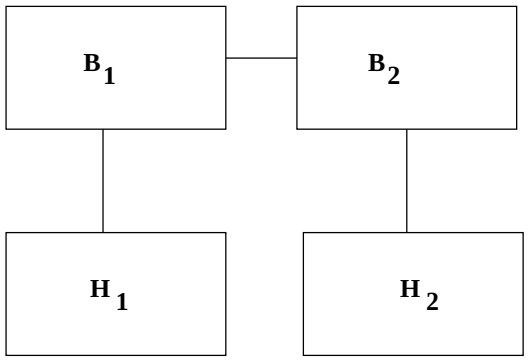

Figure 2: The figure above shows one of the network scenarios considered as part of this lab.

(c) Consider the second network scenario in Fig. 2 in which we have two Linux machines B_1 and B_2 configured as bridges using commands given above. Connect H_1 and H_2 to the two bridges as shown in Fig. 2. Ensure that H_1 and H_2 are in different networks. By default the STP is disabled on Linux bridges. Enable it on both the bridges and look for the STP details on each bridge using the following commands:

`brctl stp br0 on` (this enables STP on `br0`)

`brctl showstp br0` (shows STP details on `br0`)

What are the bridge ids of both the bridges. Look for the default priority values set on both bridges.

Which is the root bridge among the two bridges. Change the priority value on one of the bridges so that the other bridge becomes the root bridge.