Lab 07: Spanning Tree Protocol and Static Routing

OSL, 12 Mar Thu, 2015

Objective:

1. Use VNUML to understand spanning tree protocol
2. Use VNUML to learn how to route packets in Linux.

General instructions:

1. This lab is to be done in **groups of two students**
2. Read the exercise fully before starting to experiment.
3. Create a directory <rollnumber1>_<rollnumber2>_lab07. As you proceed with the lab instructions below, for each exercise, note down the answers to the exercise along with any interesting observations in a file lab07.txt created within. You may have to add other files. Details below.
4. **For the exercise 4 in this lab, you will need to show the output to the TAs to get credit.**
5. **This is a slightly lengthy lab** *(lengthy since exercises are correlated, didn't want to split)***. So focus.**

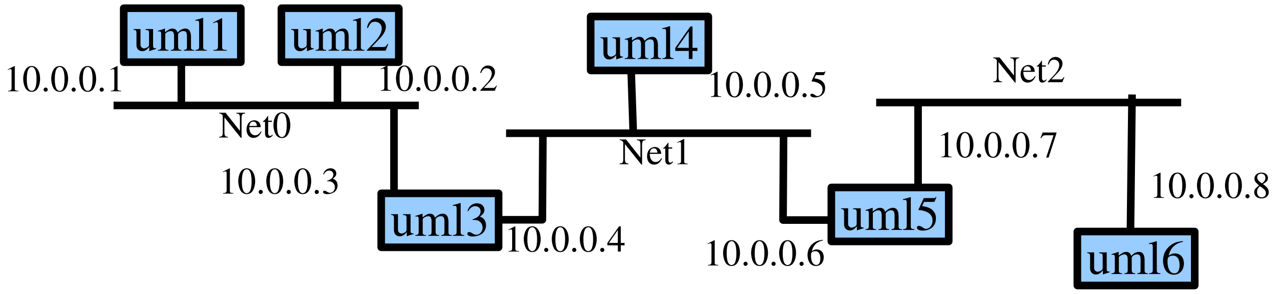
Reference:

1. Man pages of relevant commands and a **new command 'route'.**
2. Video/slides of 'Spanning Tree Protocol' under link layer.
3. Bridging commands reference: http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge (Look at the bridge commands under Module loading. Note not all commands listed in the above are relevant. Relevant portions are sections 5.2 till 5.6.1 and 5.8)

Lab Instructions:

Exercise 1: Play time with VNUML

Playtime: I have provided a VNUML configuration file corresponding to the below topology. Browse it. Bring up the virtual network using the command: " vnumlparser.pl -Z -t lab07_bridging.xml -v". ***In case you want to release the simulator (do it for a good reason since setting up the machines takes time), you can do so using the command "vnumlparser.pl -d lab06_bridging.xml -v"***



When you run the above command (it may take some time to execute), you should see six xterm windows popping up; these are the six virtual machines running Linux corresponding to the umls. You can **root** login to each by using the password "xxxx". What you do on these machines will have no effect whatsoever on the physical machine's Linux or the filesystem! So the system administrators are happy!

Checkout the xterms corresponding to the umls. They are much like your regular terminals. You can use ifconfig, route, arp, vim, cat, bash scripts etc much like on our own laptops/desktops where you have root permissions.

Play around with the terminals and check using networking commands that indeed the above topology has been created.

Repeat the last exercise of last lab and see if the spanning tree works properly. This will not be graded, its for completeness and for your understanding.
After you are done with the simulator, do not forget to release it using the "-d" option.

Exercise 2: Bridging with loops

When hooking up the three networks, Ms. Bree Edge wants fault tolerance. So some one suggested that she introduce redundancy via a loop in the topology. She wants to know how the looping issue is addressed in bridging and how bridge failures are handled by the software. Help her understand the same.

Guidance:

1. Now that you have warmed up, create a new VNUML configuration file (title the file "lab07-stp-loop.xml") connecting three networks but introduce a cycle (or even two) in the topology. Use the earlier configuration file as a reference. Apart from requisite bridges, let each network also contain a plain host that doesn't act as a bridge
2. Draw the topology in any drawing tool and export as a jpg (lab07-stp-loop.jpg), this will be part of your submission. If time is short, draw it on paper (specify roll numbers) and hand the sheet to the TA.
3. Play around with this topology to see how things work. Finally enable STP in one bridge after another and examine the output (there are specific commands that show the status of stp) after each step to see how looping is avoided.
4. After you answer some of the questions below, you will also have to bring down a bridge to test fault-tolerance.

Questions: Answer/do the following.
Eliminating loop via STP

1. For each bridge in your network, cut & paste the status of the stp prototol by running the appropriate command. Name the file "noloop-uml" and within also specify the uml the status corresponds to.
2. What is the id of the root bridge? Who is the designated bridge for each of the LAN segments in your network?
3. Based on the above information, copy and modify the figure in lab07-stp-loop to show the spanning tree. Essentially you need to remove all ports/interfaces of a bridge that are not active. Specify this file as "lab07-stp-noloop.jpg" (Draw on paper neatly if time is short)

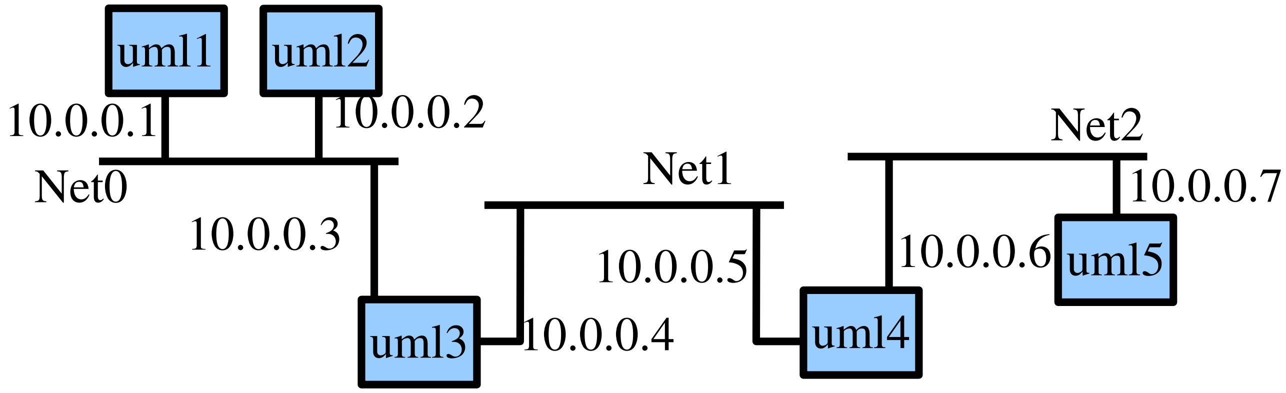
Root Bridge Failure

4. Now disable the root bridge in your topology (by using brctl again). Observe how the spanning tree changes. Make sure that it matches what you expect according to what you learnt in class. Check the connectivity between some machines again.
5. What is the id of the new root bridge? Who is the designated bridge for each of the LAN segments in your network?
6. As part of root bridge failure (rbf) exercise, for each bridge in your network, also cut & paste the status of the stp protocol to a file "rbf-uml" and within also specify the uml the status corresponds to.
7. Again create a figure showing the new spanning tree and name this file as "lab07-stp-rbf.jpg". Draw on paper if short of time. If done, do not forget to release the simulator instance.

After you are done with the simulator, do not forget to release it using the "-d" option.

Exercise 3: Manual Routing **[9 Marks]**

Mr. Rao Ting (Indian-Chinese ancestry) is studying networks and has recently taken up a job as a network administrator. All this is new to him. His first job is to interconnect the hosts belonging to different networks (see figure below). He is able to ping some hosts, while not others. Some times ping works in one direction, but not in the other. This is rather perplexing to him. Help him understand what is happening. Not only that, help him fix the problem. Specifically help him solve the below posed problems. As you solve the problem also record observations made as well as how you solved the problem in the report.



Guidance:

1. Use tcpdump to debug. To run tcpdump in the background and to free up the console for other commands, use "tcpdump -enx -w trace.out &".
2. If you wish to terminate tcpdump when running it in background, you will need to kill the tcpdump process. Use "ps ax | grep tcpdump" to get the process number and "kill" command to kill it.
3. You can view the output of tcpdump by executing the following command "tcpdump -r trace.out | less". If you wish to view addition information like the link layer headers, you can use -e option.
4. Go through the man pages of the "route" command to fix the problem using just the "route" command.

Questions:

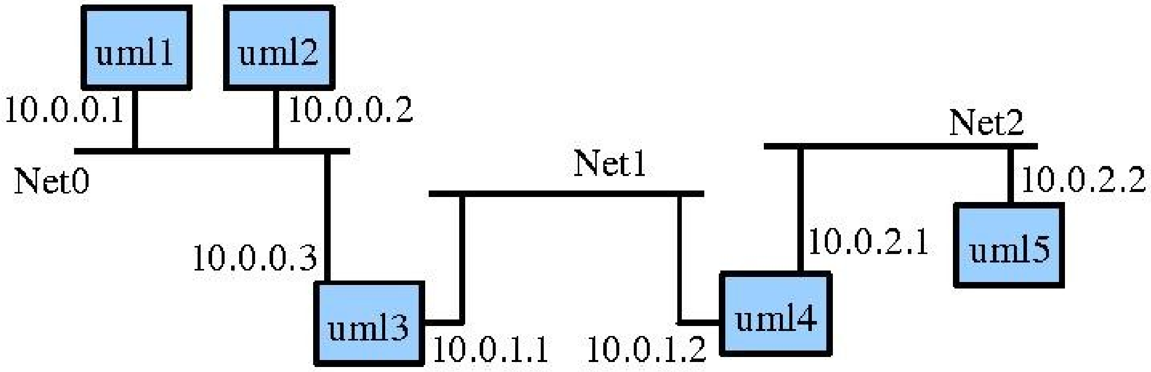
1. From uml3, ping to 10.0.0.5 does not work. Why?
2. However, ping with the "-I" option works. Why does this work?
3. Ping to 10.0.0.4 from uml4 does not work with or without -I option. Why?

4. Fix the specific problem where ping to 10.0.0.5 from uml3 and ping to 10.0.0.4 from uml4 works with or without -I option? Specify the commands you used.

Some other pings like from 10.0.0.1 to 10.0.0.7 still don't work. We will address a more sophisticated version of the problem in the exercise below.

Exercise 4: More Manual Routing **[16 Marks]**

Mr. Rao Ting's boss is fed up with pings not working the way they should and asked him to do it the clean/right way by splitting the entire network into subnets as shown in the figure. And finally to configure the nodes so that hosts across the networks can communicate with each other. This is turning out to be rather tricky. Help him solve the problem in steps. As you solve each problem record observations made as well as how you solved the problem.



Guidance:

1. Use the same vnuml configuration as above. You don't have to release the simulator. Just change the IP address of the different machines as shown in the figure using the "ifconfig" command. Ensure that the subnet mask for all the interfaces corresponds to netmask 255.255.255.0 (i.e. an 8-bit network size). Now we have three subnets corresponding to the three networks Net0, Net1 and Net2.
2. Apart from tcpdump, "route -n" is a useful tool to debug what is happening. This basically prints the forwarding table of a given terminal.
3. As part of the exercise, you will have to set default routes at a node using "route" command. Figure out how to set a default route.
4. You will also need to add routes specific to networks using "route add -net" option. Explore this option as well.
5. To enable communication across nodes, some of the nodes would have to be converted into routers. To convert a node into a router and to enable IP forwarding within, use the command "echo 1 > /proc/sys/net/ipv4/ip_forward".
6. If anything goes wrong in any of the exercises below, you can just reboot the virtual machines; or better still, try to restore order via ifconfig up/down commands. You can also learn how to delete routing entries, by using "route del" with the appropriate arguments.

Problems/Questions:

1. From uml1, ping to 10.0.1.1 does not work. Why? Also answer what all hosts received any of these ping packets?
2. Make the above ping work by appropriately configuring the routing table at uml1 (don't make any changes to any other nodes yet). Use tcpdump to figure out why it works now. Specify the commands used to make the ping work and also explain why it works now.
3. Ping 10.0.1.2 from uml1. This will not work. Why? Note down which all hosts receive these ping packets i.e the path taken by the ping packets and the host at which ping fails.
4. Fix the above ping via the following steps.
   a) To fix above ping, you will need to convert a node to a router. Figure out which node to convert and enable forwarding on that node. Alas this doesn't work. Debug via tcpdump as to why.
   b) You will need to add routing entries (hint: -net option) at one of the nodes to fix this. Figure out which node and what entries to add. After this ping should work fine. Specify the set of commands you used and at what nodes to make ping to 10.0.1.1 work from uml1.
5. Now you probably got the hang of things. Add appropriate routing entries, enable forwarding where required to make ping work from uml1 to 10.0.2.2. And from uml5 to 10.0.0.1. Show the TAs that you are able to ping in either direction to get credit for this exercise.

After you are done with the simulator, do not forget to release it using the “-d” option.

Submission instructions

The directory named <rollnumber1>_<rollnumber2>_lab07 that you will submit should contain the following files:

1. lab07.txt
2. lab07-stp-loop.xml
3. lab07-stp-loop.jpg (or paper)
4. lab07-stp-noloop.jpg (or paper)
5. File noloop-uml
6. lab07-stp-rbf.jpg (or paper)
7. File rbf-uml

Tar the directory and upload it on moodle