CSE 473 – Introduction to Computer Networks

Studio 5

(*Adapted from Jon Turner's Studios*)

In this studio, you'll be implementing a simple overlay IP forwarder that routes packets randomly. Start by claiming one of the studio reservations by opening the provided configuration file and selecting "commit" using one of the studio accounts.

1. Start by reviewing the provided source code with your group. Make sure you understand what all the parts are doing and how they fit together. Note that the interface provided by the *Substrate* includes a link number. Make sure you understand what these link numbers signify and how they are used (you may want to do the second part first, then come back to this).

2. In the *net* sub-directory, you will find a set of configuration files that define an overlay IP network that runs on top of the ONL hosts. Draw a picture of the network topology on the white board, labeling each the nodes *r*1, *r*2, etc. Label the connecting links with their "costs". Next to each node, write it's IP address in the overlay network and the name of the ONL host on which it runs.

3. Complete the implementation of the *Forwarder* thread. When sending (or forwarding) a packet, just pick a link at random using the *Random.nextInt*() method. You may assume that every node has exactly three links, numbered 0, 1 and 2. Packets addressed to "this" node should have their payloads delivered to *SrcSnk*. When forwarding a packet decrement the TTL and discard if the TTL is zero. Whenever this happens, print a message along with a copy of the packet. When first creating a packet, initialize the TTL to 15.

4. Test your program using the provided *script1* (you may need to edit the script to set the "root" directory) by typing

   ```
   ./script1 .1 10 static debug
   ```

   Look at the statistics in the log file: for r_1. For each of the nodes it sends test packets to, what is the minimum delay reported? What is the maximum delay? What is the ratio between the two? Considering the link costs in your network graph, does this surprise you at all?

   Determine the number of times that a node forwarded a packet with a TTL of 2. An easy way to do this is to type

   ```
   grep -A1 sending log* | grep "ttl=2" | wc
   ```

   Here, *wc* (stands for "word count") is a simple program that counts the number of lines, words and characters in its input. Make sure you understand how this works. Repeat this for TTLs of 3, 4, 5, 6 and 7. See anything curious?

   How many times are packets discarded because their TTLs hit zero?

5. In this part, you will be using a "discard filter" that causes one of the ONL routers to discard all the packets it received on one of the inter-router links. First, run *script1* by typing

   ```
   ./script1 1 10 static
   ```

   Notice the values of the packet counts in the *SrcSnk* statistics. Now, click on port 1 of ONL router number 1 (the left-hand router) and select the "Filter Table" item. Examine the filter and note the check box at the right end. Clicking on this check box “turns on” the filter, causing the router to discard all packets received on this port. Do this, and select “Commit” from the File menu. Then re-run the script. Observe the difference in the *SrcSnk* statistics and in the monitoring display. To turn off the filter, uncheck the box and select Commit again.

6. Try running the script without the “static” argument. How do the delay statistics compare to the earlier run? What do you expect would be the average delay between a pair of adjacent routers? You’ll need to check out the *Substrate* code to answer this.