Lab 02: NS2 Warm-up

OSL, Thu Jan 21, 2016

Objective:

1. Get familiarized with NS2 simulator
2. Understand popular performance metrics: throughput, delay and loss.

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 called <rollnumber1>_<rollnumber2>_lab02. As you proceed with the lab instructions below, for each exercise, note down the answers to the exercise along with any interesting observations in the file "lab02.txt". ***Take care to ensure that the content in the file is neatly organized***.
4. Also add to this directory any written code along with output files. You will find more details of this in the specific exercises.

Reference:

1. ns-tcp.tcl, ns-simple.tcl, (example commands)
2. NS tutorial slides
3. Goals and metric slides
4. sample.gnu (example Gnuplot file for plotting)

Lab Instructions:

Exercise 1: Warm Up **[10 Marks, 50 Min]**

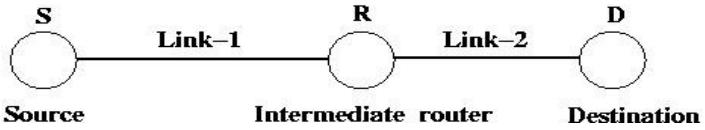
Two scripts ns-tcp.tcl and ns-simple.tcl have been provided. Run them and see the action in NAM. Also examine the output trace file (simple.tr, out.tr). Understand what is happening.

Report:

1. For ns-tcp.tcl:
   a) What is the size of the TCP packet, what is the size of the ack?
   b) What application feeds TCP?
   c) What is the queue-size limit at the intermediate node?
   d) What is the data rate and propogation delay of the two links?
   e) How many tcp data packets and how many ack packets were dropped?
2. For ns-simple.tcl
   a) How many flows and of what type (distinguish based on transport protocol) have been set-up as part of the simulation?
   b) What applicaton feeds the two flows?
   c) By looking at the 'tcl' file, can you figure out at what rate the two flows are injecting data (source data-rate) into the network? Specify the rate also.
   d) When do the flows end?
   e) Determine the throughput (as observed at the receiver) of the two flows? Hint: Filter the trace file using shell commands. Answer can be got at a glance or two.

Exercise 2: Metrics **[30 Marks, 120 Min]**

Goal: Metric Murugan wants to understand performance of protocols on networks. Since this is his first foray in this area, he decides to keep the topology and protocol simple. His smart friend Hari Puttar gave him the following topology to use and suggested that he use UDP since its much simpler than TCP. UDP just adds sequence numbers to packets and pretty much does nothing more (as far as this experiment is concerned).



There is a source node, a destination node, and an intermediate router (marked "S", "D", and "R" respectively). The link between nodes S and R (Link-1) has a bandwidth of 1Mbps and 50ms latency. The link between nodes R and D (Link-2) has a bandwidth of 100kbps and 5ms latency.

Murugan wants to understand the following:
If the source data rate (rate at which source injects data into the network) varies, what happens to the packets in the network? At what point does it get congested? How do the throughput, delay and loss vary as a function of the source data rate. Help him write a ns2 tcl script (name it ns-udp.tcl) that helps him conduct this experiment. Help him also interpret the results of the experiment.

Guidance:

1. Vary the source data rate (termed Offered Load hence forth) to take on values of 40kbps, 80 kbps, 120kbps and 160kbps. Note each value will result in one run. Thus you have 4 runs and hence 4 trace files. Ensure proper naming of the trace files (Eg: udp-40k.tr, udp-80k.tr, udp-120k.tr and udp-160k.tr). Include a tcl file of one of the runs as ns-udp.tcl in the directory.
2. Routers have limited buffer space and drop packets during congestion. Use the "queue-limit" command to model this. Limit the queue at the bottleneck link (link-2) between node "R" and node "D" to 10 packets. Its fun to monitor the queue at Link-2 (between R and D). You can do so using the *duplex-link-op* command.
3. Let each experiment run for atleast 10sec.
4. **Metrics:** For each run, calculate/measure the following metrics by processing the output trace file. You can use bash commands (grep, awk, pipe etc) to process this information.
   a) Offered Load: The rate at which source traffic is being injected into the network. (You set this value in the tcl script but confirm via trace that you got it correct.)
   b) Packet Loss: The number of packets that were sent by the sender, but didn't reach the destination. Express it in percentage.
   c) Throughput: The rate at which bits are being received at the destination? Eg: If 100 packets of size 50 bytes were received in a duration of 10 seconds, we say the throughput is 100 * 50 * 8/10 = 4000bps or 4kbps. Express it in kbps.
   d) Average end-to-end delay: If t1 is the time the packet was generated at the source and t2 was the time the packet was received at the destination. Delay of a packet is t2-t1. Calculate average of these values for packets that reached the destination. Express it in ms.

Report:

Plot the following graphs. Ensure the axis are properly labelled, with proper legend and correct units.

1. Offered Load vs percentage packet loss
2. Offered load vs throughput
3. Offered Load vs Average end-to-end delay

In the report, comment on what you observe in each graph and the reasons for the same.

Exercise 3: Earn Credit **[10 Marks, 20 Min]**

This exercise is not compulsory but you can earn extra credits by completing it.

For an offered load of 120kbps, measure queuing delay experienced by the packets at the intermediate router (ignore dropped packets) and plot it as a function of the packet number.

Calculate the average queuing delay. How does it compare to the end-to-end delay (of the same experiment)?

Submission instructions

The directory named <rollnumber1>_<rollnumber2>_lab02 that you will submit should contain the following files:

1. lab02.txt
2. ns-udp.tcl
3. 4 trace files
4. bash scripts (if used)
5. 3 plots (in eps format, properly named)
6. Optional queuing delay plot

Now tar it as follows:
tar -zcvf <rollnumber1>_<rollnumber2>_lab2.tgz <rollnumber1>_<rollnumber2>_lab02/
Submit the file <rollnumber1>_<rollnumber2>_lab02.gz via bodhitree1 for grading.