Brief Overview and Background

In this assignment you will be studying the performance behavior of TCP, using ns-2. At the end of this exercise, you should be able to write simple scripts in ns-2 as well as understand the operation of TCP.

A simple sample script that uses UDP (note that you will be using TCP) is available in your directory. The script is heavily commented, so you should be able to follow it easily. Please use this script as a reference for the sequence of commands. You can even run it and check out in **nam** the flow of packets. Any additional commands needed for the exercises over what has been explained in the script are given below. **The variables used in these commands may be different from what you may have defined, so be careful.** All exercises will need the generation of the output trace file (out.tr) and NAM trace file (out.nam). The trace format of the out.tr is as under:

<event ><time><from><to><type><size> ---- <flowid><src><dst><seqno><aseqno>
Event: Can be enque (+) at buffer, deque(-) at buffer, receive(r) at node or drop(d)
Time: Time at which the event occured
From/To: Between which two nodes tracing is happening
Type: Type of packet (tcp, ack, cbr etc)
Size: Packet Size
flowid: IPv6 flow identifier
src/dst: The source and destination node addresses
Seqno: Packet Seqno
aseqno: Unique packet identifier

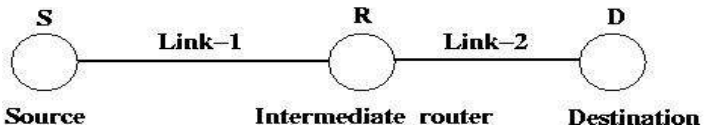
#Setup a TCP connection set tcp [new Agent/TCP/Reno] $ns attach-agent $n0 $tcp set sink [new Agent/TCPSink] $ns attach-agent $n1 $sink $ns connect $tcp $sink	#Trace the TCP parameters set par [open output/param.tr w] $tcp attach $par $tcp trace cwnd_ $tcp trace maxseq_ $tcp trace rtt_

#Setup a FTP over TCP connection set ftp [new Application/FTP] $ftp attach-agent $tcp	#Start and stop FTP connection $ns at 0.5 "$ftp start" $ns at 50.5 "$ftp stop"

#Set Queue Size of link (n1-n2) to 10 $ns queue-limit $n1 $n2 10	#Monitor queue of link (n1-n2) (for NAM) $ns duplex-link-op $n1 $n2 queuePos 0.5

Exercise 1:

Create the following topology where you have 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 10ms latency. Run the simulation and verify in NAM that the following topology is indeed generated.



Exercise 2:

Let us now build on top of the above setup. Create a TCP connection with an FTP flow running on top of it, with the TCP source at node "S", and the TCP destination at node "D" (look at the commands given above). Ensure that the TCP variant you are using is Reno. Also, monitor the queue at Link-2 (between R and D) (use the *duplex-link-op*

command). Have the simulation run from 0 to 51 sec and the FTP run from 0.5 to 50.5 seconds (for a duration of 50 seconds, look at the commands given above). Verify in NAM the flow of packets. Look at the "out.tr" you generated. You should be able to observe the following:

NAM:

1. At time 0.5 sec, you should see a packet traversing along S to R to D and an acknowledgment in the reverse direction i.e. D to R to S. At about 0.627 sec the acknowledgment reaches S (You can verify this in out.tr as well).
2. After the receipt of this acknowledgment, you should see two packets leaving S.
3. As packets flow across, can you notice the build up of queue at Link-2?
4. Notice that after 2.338s, instead of sending 2 packets in receipt of acknowledgment, only one packet gets send out. Why?
5. Can you guess what the receive window size is by looking at the queue size?

Out.tr:

1. Can you relate the packet flow observed in NAM to that in out.tr?
2. What is the packet size of the TCP connection as shown in out.tr?
3. What is the total time this connection was running (look at the first and last line of out.tr)?
4. Look at the very last line in "out.tr" that has a field "ack" (type) in it. The last-but-one column of this specific row gives the sequence number (seqno) successfully received. This essentially means these many packets have been successfully transferred. What is this number?
5. From the packet size information, duration of connection, number of packets transferrred, can you calculate the throughput achieved? How does this compare with the bottleneck bandwidth in your topology?

Exercise3:

Let us now trace the various variables associated with TCP (cwnd, maxseq, rtt etc) to get a better understanding of the operation of TCP. Trace these parameters in a file "param.tr" (as outlined above). Next, separate out each of the parameters into separate files, for instance by "*grep cwnd_ param.tr > cwnd.tr*"

The first set of values in "param.tr" will have invalid values for the various parameters, at time 0 -- this should not really matter, but be aware of this while looking at any traces. We are mainly interested in variation of congestion window (cwnd), sequence number

(maxseq) and round-trip time (rtt) with time. Use the individual files (cwnd.tr, maxseq.tr, rtt.tr) to plot the variation of these parameters with time. Use the file “plot.gnu” provided in the directory to plot. Look inside this file to ensure that the path to the individual trace files is valid. Command to use is “ gnuplot < plot.gnu”. This should generate the necessary png files.

1. In the plot of cwnd with time, can you identify the slow start and the congestion avoidance phase?
2. Do you see a change of slope in the maxseq plot at the transition point between slow start and the congestion avoidance phase?
3. When RTT reaches a steady state value, can you explain the individual components that make up for this value of RTT?
4. Can you explain why towards the end when cwnd is close to 40, the buffer at Link-2 as seen in nam is still under 20 packets? Note that default value of buffer at Link-2 is large enough to accommodate more than 40 packets.

Exercise 4:

Let us now restrict the buffer size of Link-2 to introduce packet losses (congestion) and study TCP performance. Use the “queue-limit” command to limit the bottleneck link (link-2) between node "R" and node "D" to 10 packets. Run the simulation and plot cwnd, maxseq and rtt vs time.

1. Run nam. When does the buffer starts discarding packets? Look for lines starting with a "d" in "out.tr". These represent packet drops. Can you find the first packet drop? Does it correspond to the same time as in nam? What is its sequence number?
2. Calculate the throughput in this experiment and compare it with that you obtained in exercise2.
3. Notice the famous “saw-tooth” shaped behavior of TCP. Can you identify the slow-start and congestion-avoidance pahses?
4. Can you explain why the shape of the three plots are so? To understand the behavior better, try to plot the graphs over smaller time intervals.

Exercise 5:

Change the variant of TCP from Reno to Newreno i.e. use [new Agent/TCP/Newreno]. Run the simulation again.

1. Compare the cwnd plot with that obtained in exercise4. Can you see the improvement Newreno offers over Reno?
2. What is the throughput improvement of TCP Newreno over Reno?