# 18CS46, 2022, ASSSIGNMENT

**INSTRUCTIONS**

- Please read the assignment in full before you start out.
- The assignment Due date will be announced in the google classroom.
- Assignments have to be done individually and not in groups.
- You can take the help from the online sources as well as me but not your fellow students.
- If one or more codes and reports are similar the one who has given and one who has copied all will be awarded zero.
- This assignment will be added to your final total of your internal marks.
- You can use tools like UNIX, Bash, Perl, gnuplot, gimp etc
- You will submit this assignment electronically. Please submit a neatly typed out document answering the questions in the various parts. Please follow the format below for ease of correction.
  - Create a directory “assignmen1-xyz” where xyz should be your USN
  - Submit your main file as “answer.txt”
  - For each of the parts in the assignment, you will be submitting various files. You will have to put these files in a subdirectory “part-x” where x is the part number. The files to submit will be specified in each part.
  - Submit the whole thing as a single tar-gzipped file as “assignment1-zyx.tar.gz”
  - Please submit all your submissions to google classroom
  - And paste the link to your github repository in the same zip file.

**BEHAVIOUR**

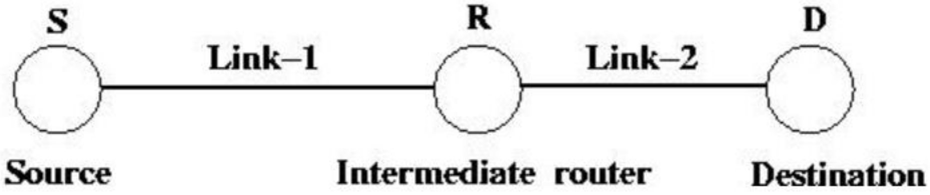
In this assignment you will be studying the performance behaviour of TCP, first on wired links, and then on lossy/wireless links. You will use the ns-2 Network Simulator. It has been installed in the cs625 account. To use ns, set your shell PATH variable to include "/3u3/course/cs625/ns-allinone-2.26/bin".

You can look at an NS tutorial on the web, such as this one here. You can download some example script, view some animations and have fun to begin with. I suggest you go through the explanation of the example script since it will help you in the exercise below.

The NS documentation is available at "/3u3/course/cs625/ns_doc.pdf".

**PART-1**

For your exercises, you will be creating the following simple topology, where you have a source node, a destination node, and an intermediate router (marked "S", "D", and "R" respectively).



The part between node "S" and node "R" is marked as link-1, and the part between node "R" and node "D" is marked as link-2. Link-1 is supposed to be the wired part of the network, with high bandwidth, and (almost) no errors. Link-2 is supposed to be the wireless hop of the network, with lower bandwidth, and high error probability (although you will introduce wireless errors only in a later part of the assignment).

**Here are the steps in this part:**

1. In this part, you are supposed to create in ns-2, the above topology, with the following parameters:
    - Link-1: 100Mbps, 50ms latency
    - Link-2: 100Kbps, 1ms latency
2. In addition, 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".
3. Make sure that the TCP source you create is the Reno variant. This you will do for example by having the line "set tcp [new Agent/TCP/Reno]" when you create the TCP source object in your TCL code.
4. Have the simulation run from time 0 to 51 seconds, and the FTP run from 0.5 to 50.5 seconds (for a duration of 50 seconds).
5. Have the ns-2 code generate a trace-file under the name "out.tr". You will use the command "$ns trace-all $f" for this.
6. Also animate the above using NAM (it should be easy to do this, if you had gone through the example in the ns-2 tutorial). You will use the command "$ns namtrace-all $nf" for this. Create the nam output under the name "out.nam".
7. You can refer to the ns_doc.pdf file, the introduction chapter, for information on trace-all and namtrace-all commands.
8. Answer the following **question** in your main file (answers.txt). Look at the last few lines of out.tr you generated. Look for a line containing the word "ack". The

last-but-one column gives the sequence number. As you can see from any of the lines containing the word "tcp", the packet size is "1040 bytes". The last (ack) sequence number should give you the number of packets successfully received in the 50 seconds of simulation (do not care about plus-minus-1 here). From the packet size information, what in the number of bytes transmitted in 50 seconds? What is the throughput achieved? How does this compare with the bottleneck bandwidth in your topology? (Beware of the 1024 vs. 1000 notation difference while checking your numbers here).

**Here's what to submit for this part (in sub-dir "part-1"):**

- The ns-2 TCL script for the above topology creation.
- A screen-shot of the NAM animation, somewhere in the middle of the FTP flow, after at least 5 seconds (you can fast-forward, pause NAM, and then use "gimp" to capture the window). The screen-shot may be in JPG or GIF format.

**Part-2 (TCP slow-start and congestion avoidance)**

From here on, you will just have to make minor modifications to the ns-2 TCL code you have written above. First, you are going to make the following modifications:

- First, you will have the TCP source collect trace information by using the following commands:

```
# Open the TCP trace file
set par [open param.tr w]

...

# Create the TCP source object
set tcp [new Agent/TCP/Reno]

...

#trace the tcp parameters
$tcp attach $par
$tcp trace cwnd_
$tcp trace maxseq_
$tcp trace rtt_
$tcp trace dupacks_
$tcp trace ack_
$tcp trace ndatabytes_
$tcp trace ndatapack_
$tcp trace nrexmit_
$tcp trace nrexmitpack_
```

Make sure to also close the "param.tr" trace file in the "finish" procedure, just like you close the "out.tr" and "out.nam" files.

- Next, you will limit the queue size in the bottleneck link (link-2) between node "R" and node "D" by using the command "queue-limit". (See section 7.4 of ns_doc.pdf). Set the limit to be 10 packets.
- Now you can run the simulation as earlier. You will now have an extra trace file "param.tr" with the TCP parameters, as seen at the sender "S". The file "param.tr" will have time in the first column, and the value of the parameter in the last column.
- 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. I will call these individual files as cwnd.tr, maxseq.tr, rtt.tr, etc.

- Use gnuplot to plot three different files:
  - The congestion window (cwnd) as a function time (can you identify the slow-start and congestion-avoidance phases)?
  - The maximum sequence number sent (maxseq) as a function of time
  - The round trip time (rtt) as a function of time
- Answer the following **questions** in your main file (answers.txt):
  - Just as in part-1, calculate the throughput achieved in this case? How does it compare with the throughput you saw in part-1?
  - Look for lines starting with a "d" in "out.tr". These represent packet drops. Can you find the first packet drop? What is its sequence number? (Look at the last-but-one column for the sequence number). What is the time when this drop occurs? (Time is given in the first column).
  - Look at the cwnd.tr file and determine the time when slow-start ends and congestion avoidance begins.
  - Explain in words, the shape of each of the three plots.

What to submit for this part (in sub-dir part-2):

1. The ns-2 TCL code
2. The three plots that you generated above

## Part 3 (TCP and wireless losses)

In this part, you will be studying the behaviour of TCP under wireless losses. See section 13.4 of ns_doc.pdf for this. In fact, you will use the last few lines of page 127, and the "link-lossmodel" command in the subsequent page.

- Introduce errors in the wireless link (link-2). Start with an error probability of 0.01 (1 percent), and increase in steps of 0.01, upto 0.10 (10 percent). Also include the 0 percent error case for comparison. In each case, run the simulation, and compute the throughput achieved. Plot a graph (probably using gnuplot) of the throughput achieved (in Kbps) versus the error probability.

### What to submit for this part (in sub-dir part-3):

1. Submit the plot that you generated above

### c**hecklist before submission**

In answers.txt, you should have answers for **one** question in part-1, and **four** questions in part-2.

The tree of files/directories should be:

```
  assignment1-xyzabc/
  answers.txt
part-1/
        code.tcl
        screenshot.jpg
 part-2/
        code.tcl
        cwnd.png
        maxseq.png
        rtt.png
 part-3/
        throughput-plot.png
```