NS2 SIMULATION, PART 2

General

The objectives of this assignment (Part 2) are as follows:

[1] to continue acquaintance with simulation capabilities of the network simulator ns2;

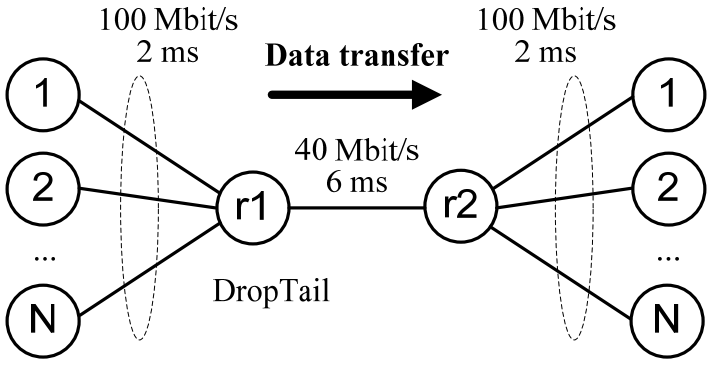
[2] to conduct case studies dealing with: a) TCP and UDP ability to share available bandwidth; b) how router buffer size affects TCP performance and network characteristics; c) effects of parallel TCP connections on total TCP goodput and link utilization.

Your task is to write a Tcl script according to these instructions and perform some simulations with it.

Since Part 2 should be done only after successful completion of Part 1, it is presumed that students have basic knowledge about ns2, such as creating simulation scenarios, tracing objects, and processing trace files.

The network to be simulated

Use the network topology shown below (also known as “dumbbell topology”). In the figure, every circle represents a node, while lines represent links.



All access links are full-duplex and have 100 Mbit/s of bandwidth and 2 ms of delay. The link between r1 and r2 is also full-duplex and has 40 Mbit/s of bandwidth and 6 ms of delay. From here it follows that the link r1-r2 is a bottleneck link.

The queuing discipline of all links is DropTail (FIFO).

Limit the maximum queue size at r1-r2 link to 67 packets.

Use the following TCP/UDP/CBR parameter:

```
#TCP Tahoe parameters
Agent/TCP set maxcwnd_ 200
Agent/TCP set window_ 200
Agent/TCP set windowInit_ 2
Agent/TCP set packetSize_ 1460
Agent/TCP set tcpTick_ 0.001

#TCP NewReno parameters (Slow-but-Steady)
Agent/TCP/Newreno set maxcwnd_ 300
Agent/TCP/Newreno set window_ 300
Agent/TCP/Newreno set windowInit_ 2
Agent/TCP/Newreno set packetSize_ 1460
Agent/TCP/Newreno set tcpTick_ 0.001
Agent/TCP/Newreno set newreno_changes_ 0
Agent/TCP/Newreno set newreno_changes1_ 0


#UDP and CBR parameters
$udp set packetSize_ 1500
$cbr set packetSize_ 1500
```

Use `Agent/TCPSink` as a TCP receiver and one-way TCP agent as a TCP sender.

Set different colours for data flows.

Generate TCP traffic using FTP and UDP traffic using CBR application.

Start sending data from all sources at the same time and stop each flow after 130 seconds. In order to remove transient phase at the beginning of simulation start to trace data after 10 seconds from the beginning of simulation (or analyse only the last 120 seconds of the trace).

Since the number of sources (and, consequently, receivers) will vary in different tasks, you are required to define network topology using loop statements.

Note that command `namtrace-all` may result in the creation of huge files. In order to save Lintula's processors and space use nam visualization only as a debugging tool at the script debug phase.

Tasks

[1] Use TCP Tahoe senders and run simulations varying the number of TCP senders from 1 to 10. At each step calculate the average goodput (receive rate) of each connection and its bottleneck bandwidth share, and summarize in a table. Plot a graph where the bottleneck bandwidth share of the first connection is on the y-axis and the total number of TCP connections is on the x-axis.

On a base of simulation results (provide the graph) explain TCP resource sharing policy while coexisting with identical TCP connections.

[2] Set the number of sources to be equal to 2, where one of them is TCP Tahoe sender and the other is UDP agent with a CBR application on top. Run 11 simulations varying CBR rate from 0 to 40 Mbit/s with a step of 4 Mbit/s. At each step calculate the average goodput of the TCP connection and its bottleneck bandwidth share. Plot a graph where the bottleneck bandwidth share of the TCP connection is on the y-axis and the CBR rate is on the x-axis.

On a base of simulation results (provide the graph) explain the difference between TCP and UDP flows in sharing resources. When does UDP flow behave itself non-aggressive and TCP-friendly?

[3] Set the number of sources to be equal to 1. Use TCP NewReno agent (Slow-but-Steady variant) as a TCP sender (use `maxcwnd_ 300` and `window_ 300`). Run simulations varying the maximum queue size at r1-r2 link from 10 to 130 with a step of 30 packets.

Let us define the ratio between queue size and bandwidth-delay product of the network as

$$X = \frac{queue\_size}{BDP},$$

where bandwidth-delay product (BDP) of the network can be found as

$$BDP = \frac{20\text{ms} \times 40\text{Mbit/s}}{8 \times 1500\text{Bytes}} \approx 67\text{pkts}.$$

Plot the following graphs:

- average TCP loss rate as a function of X;
- average queuing delay as a function of X;
- average bottleneck link utilization as a function of X.

Provide the graphs and give your explanation of the obtained results. How do you think what is the most appropriate queue size for the given topology (in terms of small loss rate, low delay, and high bottleneck link utilization)? Compare it with bandwidth-delay product of the network.

Note that $RTT_{\text{observed}} = RTT_{\min} + queuing\_delay$. Thus, the average queuing delay can be calculated as

$queuing\_delay_{\text{average}} = \overline{RTT} - RTT_{\min}$,

where $RTT_{\min} = 20\text{ms}$.

[4] Set the maximum queue size of r1-r2 link to be equal to 67 packets and the maximum congestion window size of TCP Tahoe agent to be equal to 20 packets (`maxcwnd_ 20`). Run simulations varying the number of TCP Tahoe senders as follows: 1, 2, 3, 4, 8, 16, and 32. Plot a graph where the utilized bottleneck bandwidth (or, what is the same thing, total goodput of the parallel TCP connections) is on the y-axis and the number of parallel TCP connections is on the x-axis.

Provide the graphs and give your explanation of the obtained results. What is the advantage of using parallel TCP connections? How many connections are required to fully utilize available bandwidth in the given conditions?

Report

Provide your report with a cover page containing your name, student number, e-mail, and name of your work directory. Also include script text into the report.

You should leave your report (hard copy only) in the report box 247: Tietotalo Building, 2nd floor, nearby the entrance to the G-wing. Also send it via e-mail to dunaytse@cs.tut.fi.

The report will be graded using pass/fail. The following issues will be evaluated: the correctness of the results, the written part of the report, the presentation, and appropriate commentary of the results.

In order to save disk space, please remove your work directory (at /export/tlt/250x-ns2/) after getting notification about successful pass. Note that only after that you will get your credit points.