

S-38.180: Quality of Service in Internet

Exercise 1: Packet latency and throughput

Deadline: October 29th, 4 pm

Grading: Pass / Fail

Return: PDF by e-mail OR paper to course locker in the G-wing

Introduction

Packet latency caused by the network has a significant effect on the throughput of a TCP-connection. TCP-protocol relies on the acknowledgements sent by the client-side of the connection in deciding the optimal send rate which is controlled by altering the size of the congestion window. The congestion window represents the maximum number of unacknowledged packets sent to the network. Initially the window has a low value but during the communication it is increased or decreased depending on the arrivals or lack of arrivals of acknowledgement packets. The more it takes for an acknowledgement to arrive the more time it takes for the send rate to be increased. In this exercise you will measure the effect of packet latency on the throughput of multiple TCP-connections competing of the same network resources.

Network Topology

The topology of the network to be simulated is shown in illustration 1. The link between nodes 0 and 1 has a bandwidth of 5 Mbps and a fixed packet delay of 20 ms. The rest of the links have a bandwidth of 10 Mbps and varying delays which will be altered in the simulations.

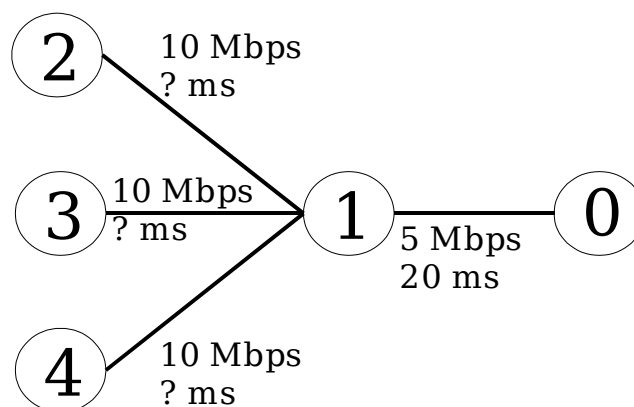


Illustration 1 Topology of the simulated network

Traffic Generation

The traffic in the network will be created by defining three TCP-connections controlled by FTP-application. The corresponding classes in NS-2 are:

- Agent/TCP/FullTcp
- Application/FTP

The connection pairs are as follows:

1. server node 2 <-> client node 0
2. server node 3 <-> client node 0
3. server node 4 <-> client node 0

All connections use the link between node 0 and 1 and it will be the bottle-neck link in the network.

Queueing

Define RED-queueing (Random Early Detection) in the link between nodes 0 and 1 (bottle-neck). Packet drops will occur only in the bottle-neck link in this network, so the queueing discipline of the other links is not relevant. The easiest way is to define the other links as DropTail-links.

Use the following code to create the RED-queue and to set the parameters:

```
$ns duplex-link $n0 $n1 5Mb 20ms RED
set bn_queue [[$ns link $n1 $n0] queue]
$bn_queue set limit_ 35      ;# Queue length (packets)
$bn_queue set thresh_ 8     ;# Lower RED threshold (packets)
$bn_queue set maxthresh_ 24 ;# Upper RED threshold (packets)
$bn_queue set q_weight_ 0.3 ;# Weight-parameter used in the EMA queue size
                          ;# calculation (EMA=Exponential Moving Average)
$bn_queue set linterm_ 1    ;# Inverse of the drop probability in maxthresh_
```

See <http://www.isi.edu/nsnam/ns/doc/node69.html> for more clarification on the RED-parameters.

Monitoring

Collecting the simulation results is possible by tracing all the links or just the bottle-neck link. Traces can then be post-processed by using tools like 'awk' or 'grep' to extract the needed information. However, flow monitoring is a much more efficient method of collecting traffic statistics from a single link - in this case the bottle-neck link. Flow monitors are used to collect TCP- and UDP-flow information, such as the number of transmitted and dropped packets. The distinction between different flows can be defined by the user. The simplest way is to set a unique flow-ID for each connection and configure the flow monitor to use them. The following is an example of attaching a flow-monitor on a link:

```
set fmon_file [open output_file.fmon w]
set flow_mon [[$ns makeflowmon Fid]
$ns attach-fmon [[$ns link $n1 $n0] $flow_mon 0
$flow_mon attach $fmon_file
```

The monitor is called each time the statistics need to be printed on the output file. For example writing the statistics after 30 seconds of simulation is done with command:

```
$ns at 30 "$flow_mon dump"
```

An example line of the output file is shown below:

```
65.000 1 0 0 1 2 0 24078 36109700 0 0 49350 74008940 0 0 306 457540 77 114040
```

The five highlighted elements are: the time the statistics were gathered, flow-ID, the number of arrived packets in this flow, the number of arrived packets on the whole link and the number of dropped packets in this flow.

Task

1. Create the simulation described above
2. Simulate scenarios where the changing parameter is the delay on the three client side links. For example in one scenario the link delays could be 10, 20 and 30 ms.
3. Measure the throughputs received by the TCP-connections in each scenario. Run several times (> 3) each scenario and use the average throughputs in your analysis. Present your results and explain how much the packet latency effects the throughputs.
4. How does the utilization of the bottle-neck link change when scaling up the delay on all three links? For example, what is the effect when changing the delays from 10, 20 and 30 ms to 20, 40 and 60 ms respectively? Use at least 4 measurements. Note: Utilization = Throughput / Bandwidth.

Hints

1. The first source of information is the NS-2 home page:

<http://www.isi.edu/nsnam/ns/>

Especially the manual will probably prove useful. Use the PDF-version of it for it is more complete than the HTML-version. You can download the PDF-version also from here:

http://www.netlab.hut.fi/opetus/s38180/s03/exercises/ns_doc.pdf

2. You can use the example script as the basis of your simulation so you don't have to start from scratch:

http://www.netlab.hut.fi/opetus/s38180/s03/exercises/ex1/example_script.tar.gz

3. To use NS-2 in the Computing Center's computers set the environment by typing:

```
"source /p/edu/s-38.180/usens2.csh"
```

You can then start a simulation by typing:

```
"ns your_script.tcl"
```