Albert Shevchuk

4/28/19

CSC 645 Computer Networks

Jung-Yup Kang

**Network Simulations Using NS2**

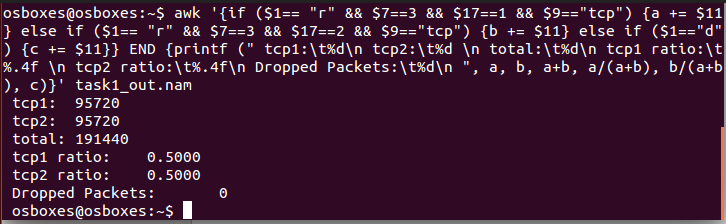
**Introduction**:

This write up was based on the Linux based Network Simulator called NS2. The instructions given were to install a Linux operating system Ubuntu version 14.04, install a modified version of NS2, install related supporting packages and set up the NS2 environment. After the ns command was recognized by the kernel, a provided sample file task1\_sample.tcl was used to run on NS2’s GUI interface. From then on, the simulation ran for approximately 6 seconds and an output trace file was generated from where command line scrips were used to examine and analyze: TCP traffic, ratios, packet loss and etcetera. The Analysis questioner will be answered as followed:

1. Based on the execution results of the Awk commands, what is the total traffic of each flows? What is the overall packet loss rate of the two TCP flows? What is the ratio of throughput between the two TCP flows?

**Command Used:**

awk '{if ($1== "r" && $7==3 && $17==1 && $9=="tcp") {a += $11} else if ($1== "r" && $7==3 && $17==2 && $9=="tcp") {b += $11} else if ($1=="d") {c += $11}} END {printf (" tcp1:\t%d\n tcp2:\t%d \n total:\t%d\n tcp1 ratio:\t%.4f \n tcp2 ratio:\t%.4f\n Dropped Packets:\t%d\n ", a, b, a+b, a/(a+b), b/(a+b), c)}' task1\_out.nam

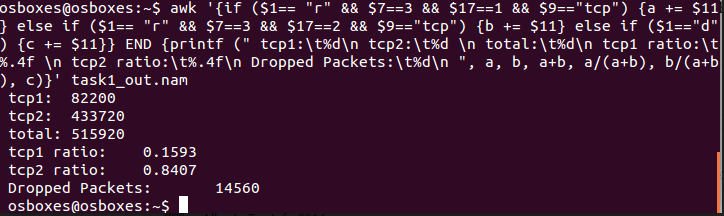
**Output**:

1. Revise the sample code "task1\_sample.tcl" so that the window size of the second TCP flow (tcp2) changes from 1 to 25, and save it a new file “task1\_b.tcl”. Use the Awk commands to compute the total traffic of each flow. How does the packet loss rate change? Which flow uses more bandwidth in the bottleneck from node 3 to node 4? What is the reason?

**Command Used:**

awk '{if ($1== "r" && $7==3 && $17==1 && $9=="tcp") {a += $11} else if ($1== "r" && $7==3 && $17==2 && $9=="tcp") {b += $11} else if ($1=="d") {c += $11}} END {printf (" tcp1:\t%d\n tcp2:\t%d \n total:\t%d\n tcp1 ratio:\t%.4f \n tcp2 ratio:\t%.4f\n Dropped Packets:\t%d\n ", a, b, a+b, a/(a+b), b/(a+b), c)}' task1\_out.nam

**Output:**

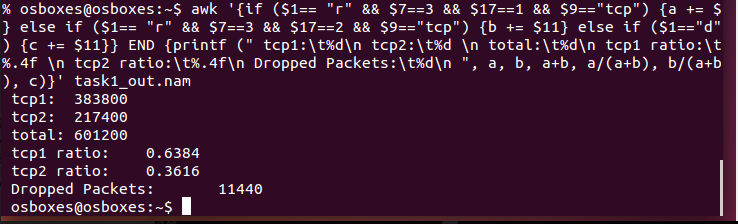


The difference here is that there is packet loss because the tcp2’s window size has been increased and is able to send more packets therefor hogging the bandwidth. Together both tcp’s exceed the buffer size and packet loss occurs.

1. Revise the sample code "task1\_sample.tcl" so that the window size of both the TCP flows are set to 5, and save it as a new file “task1\_c.tcl”. Use the Awk commands to compute the total traffic of each flows. How does the packet loss rate change? Do both flows receive a "fair share" of the available bandwidth of the bottleneck link?

**Command Used:**

awk '{if ($1== "r" && $7==3 && $17==1 && $9=="tcp") {a += $11} else if ($1== "r" && $7==3 && $17==2 && $9=="tcp") {b += $11} else if ($1=="d") {c += $11}} END {printf (" tcp1:\t%d\n tcp2:\t%d \n total:\t%d\n tcp1 ratio:\t%.4f \n tcp2 ratio:\t%.4f\n Dropped Packets:\t%d\n ", a, b, a+b, a/(a+b), b/(a+b), c)}' task1\_out.nam

**Output:**

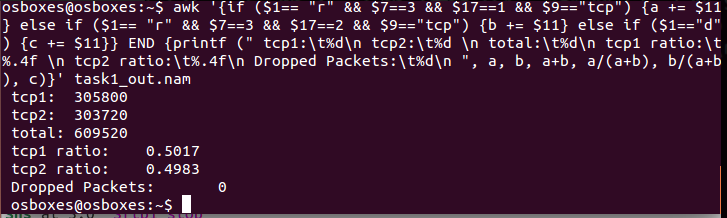
The packet loss rate has gone down, but tcp2 still holds most of the bandwidth so it’s not a fair share.

1. Revise the sample code "task1\_sample.tcl" so that the window size of both the TCP flows are set to 5 and the queue buffer size for the link from node 3 to node 4 is set to 40 packets. Save the new file as “task1\_d.tcl”. Use the Awk commands to compute the total traffic of each flows. How does the packet loss rate change? Do both flows receive a "fair share" of the available bandwidth of the bottleneck link?

**Command Used:**

awk '{if ($1== "r" && $7==3 && $17==1 && $9=="tcp") {a += $11} else if ($1== "r" && $7==3 && $17==2 && $9=="tcp") {b += $11} else if ($1=="d") {c += $11}} END {printf (" tcp1:\t%d\n tcp2:\t%d \n total:\t%d\n tcp1 ratio:\t%.4f \n tcp2 ratio:\t%.4f\n Dropped Packets:\t%d\n ", a, b, a+b, a/(a+b), b/(a+b), c)}' task1\_out.nam

**Output**:

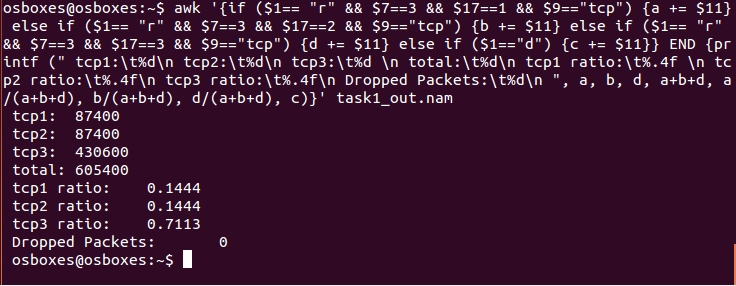
****

The packet loss have been eliminated because the queue buffer was increased and thus managing the flow coming from both ends. Helping the ratios equalize. This scenario would be considered fair share.

1. Revise the sample code “task1\_sample.tcl” and add node 5 to the network configuration. It is connected to node 3. The link between node 5 and node 3 has a speed of 2 Mbps and a propagation delay of 10 ms. The new network topology is shown in Figure 2. One more TCP/FTP flow with a window size of 5 is generated from node 5 to node 4. Save the new file as “task1\_e.tcl”. Run the simulation. Revise the sample Awk commands to compute the total bytes of TCP for each flow. What are the Awk commands? Do all the TCP flows receive a fair "share" of the bandwidth? Why or why not?

**Command Used:**

awk '{if ($1== "r" && $7==3 && $17==1 && $9=="tcp") {a += $11} else if ($1== "r" && $7==3 && $17==2 && $9=="tcp") {b += $11} else if ($1== "r" && $7==3 && $17==3 && $9=="tcp") {d += $11} else if ($1=="d") {c += $11}} END {printf (" tcp1:\t%d\n tcp2:\t%d \n tcp3:\t%d \n total:\t%d\n tcp1 ratio:\t%.4f \n tcp2 ratio:\t%.4f\n tcp3 ratio:\t%.4f\n Dropped Packets:\t%d\n ", a, b, d, a+b+d, a/(a+b+d), b/(a+b+d), d/(a+b+d), c)}' task1\_out.nam

**Output**:

Tcp1 & tcp2 have fair share, the 5th node (tcp3) dominates because the window size is increased. This combination balances out the flow of the packets thus eliminating the packet loss.

**Conclusion**:

With the 5 exercises above we can conclude that the experiment went as expected. With the efforts of modifying the original tcl file we have demonstrated a simulation of networks with packet loss and ways we can eliminate the loss.