

9 Mathematical Modelling of TCP Throughput Performance

9.1 Introduction

The average throughput performance of additive-increase multiplicative-decrease TCP congestion control algorithms have been studied in a variety of network scenarios. In the regime of large RTT, the average throughput performance of the TCP congestion control algorithms can be approximated by the ratio of the average congestion window *cwnd* and RTT.

9.1.1 Loss-less Network

In a loss-less network, we can expect the TCP congestion window *cwnd* to quickly increase to the maximum value of 64 KB (without TCP scaling). In such a case, the long-term average throughput of TCP can be approximated as

$$\text{Throughput} \approx \frac{64 \times 1024 \text{ (bits)}}{\text{RTT (in secs)}}$$

9.1.2 Lossy Network

We refer to an exercise in Chapter 3 of Computer Networking: A top-down approach, by Kurose and Ross for the setup. Consider a TCP connection over a lossy link with packet error rate *p*. In a period of time between two packet losses, the congestion window may be approximated to increase from an average value of *W/2* to *W* (see **Figure 9-8** for motivation). In such a scenario, the throughput can be approximated to vary from *W/2/RTT* to *W/RTT* (in the cycle between two packet losses). Under such assumptions, we can then show that the loss rate (fraction of packets lost) must be equal to

$$p = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

and the average throughput is then approximately,

$$\text{Throughput} \approx \sqrt{\frac{3}{2p}} \times \frac{\text{MSS (in bits)}}{\text{RTT (in secs)}}$$

9.2 Network Setup

We will seek a large file transfer with TCP over a loss-less and lossy link to study long-term average throughput performance of the TCP congestion control algorithm. We will simulate the network setup illustrated in **Figure 9-2** with the two (loss-less and lossy) configuration parameters listed in detail in Table 1 to study the throughput performance of TCP New Reno.

Open NetSim and click **Examples > Experiments > Mathematical-model-of-TCP-throughput-performance > Sample-1** as shown below **Figure 9-1**.

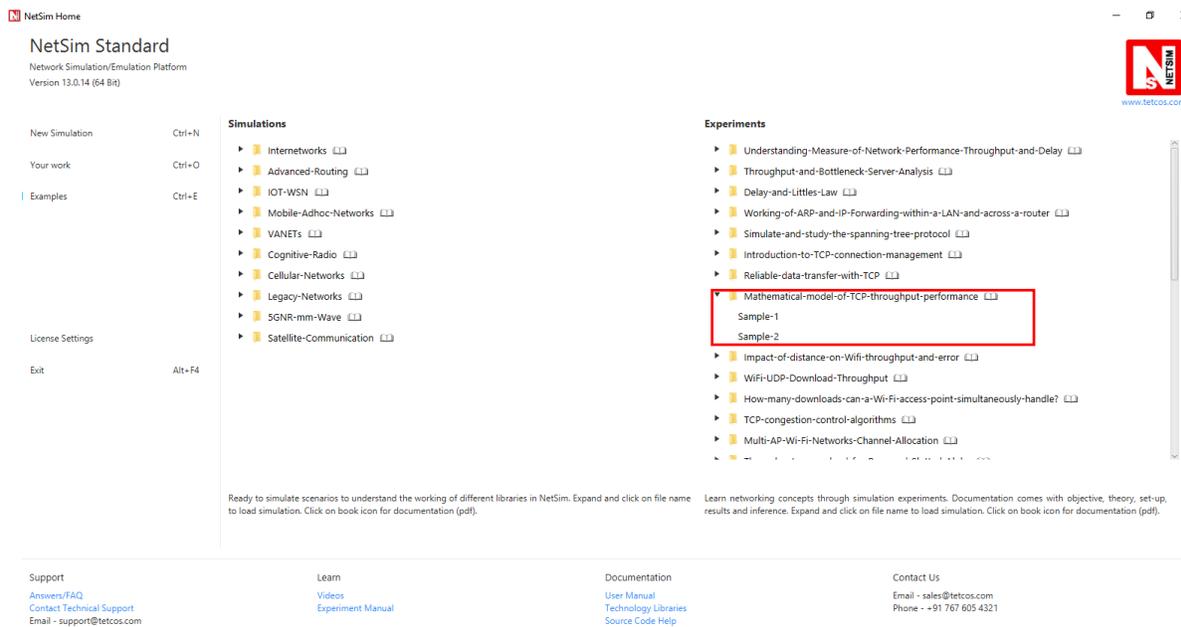


Figure 9-1: Experiment List

NetSim UI displays the configuration file corresponding to this experiment as shown below **Figure 9-2**.

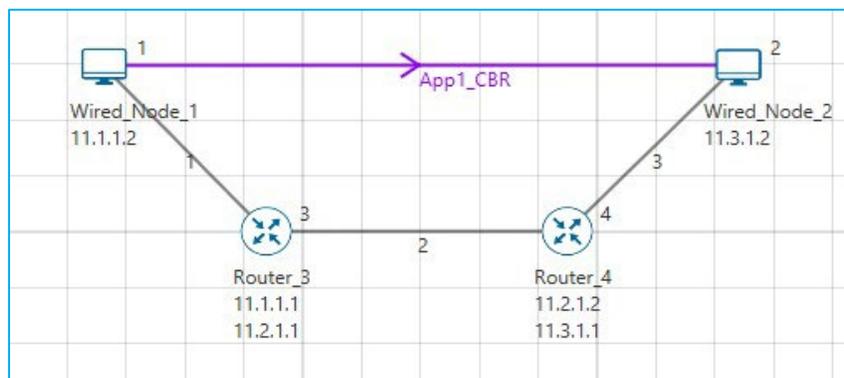


Figure 9-2: Network Topology

9.3 Procedure

Sample 1

The following set of procedures were done to generate this sample.

Step 1: A network scenario is designed in NetSim GUI comprising of 2 Wired Nodes and 2 Routers in the “**Internetworks**” Network Library.

Step 2: In the General Properties of Wired Node 1 i.e., Source, Wireshark Capture is set to Online. Transport Layer properties Congestion plot is set to true.

Note: *Accept default properties for Routers as well as the Links.*

Step 3: Right-click the link ID (of a wired link) and select Properties to access the link’s properties. Set Max Uplink Speed and Max Downlink Speed to **10** Mbps. Set Uplink BER and Downlink BER to **0**. Set Uplink Propagation Delay and Downlink Propagation Delay as **100** microseconds for the links 1 and 3 (between the Wired Node’s and the routers). Set Uplink Propagation Delay and Downlink Propagation Delay as **50000** microseconds for the backbone link connecting the routers, i.e., 2.

Step 4: Right click on the Application Flow **App1 CBR** and select Properties or click on the Application icon present in the top ribbon/toolbar.

An CBR Application is generated from Wired Node 1 i.e., Source to Wired Node 2 i.e. Destination with Packet Size set to 1460 Bytes and Inter Arrival Time set to 1168 microseconds.

Step 5: Click on Display Settings > Device IP check box in the NetSim GUI to view the network topology along with the IP address.

Step 6: Click on **Plots** icon and select the **Enable Plots** checkbox. This enables us to view the throughput plot of the application **App1 CBR**.

Step 7: Click on Run simulation. The **simulation time** is set to 100 seconds. In the “**Static ARP Configuration**” tab, Static ARP is set to **disable**.

Sample 2

Step 1: Right-click the link ID (of a wired link) and select Properties to access the link’s properties. Set Max Uplink Speed and Max Downlink Speed to **10** Mbps. Set Uplink BER and Downlink BER to **0**. Set Uplink Propagation Delay and Downlink Propagation Delay as **100** microseconds for the links 1 and 3 (between the Wired Node’s and the routers). Set Uplink Propagation Delay and Downlink Propagation Delay as **50000** microseconds and Uplink BER and Downlink BER to **0.0000001** for the backbone link connecting the routers, i.e., 2.

Step 2: Click on Run simulation. The **simulation time** is set to 100 seconds. In the “**Static ARP Configuration**” tab, Static ARP is set to **disable**.

9.4 Output

In **Figure 9-3**, we report the application metrics data for data transfer over a loss-less link (Sample-1).

Application_Metrics_Table							
Application_Metrics							<input type="checkbox"/> Detailed View
Application Id	Throughput Plot	Application Name	Packet generated	Packet received	Throughput (Mbps)	Delay(microsec)	Jitter(microsec)
1	Application Throughput plot	App1_CBR	85617	42055	4.912024	25667072.446895	1206.188143

Figure 9-3: Application Metrics when BER=0

In **Figure 9-4**, we report the plot of long-term average throughput of the TCP connection over the loss-less link.

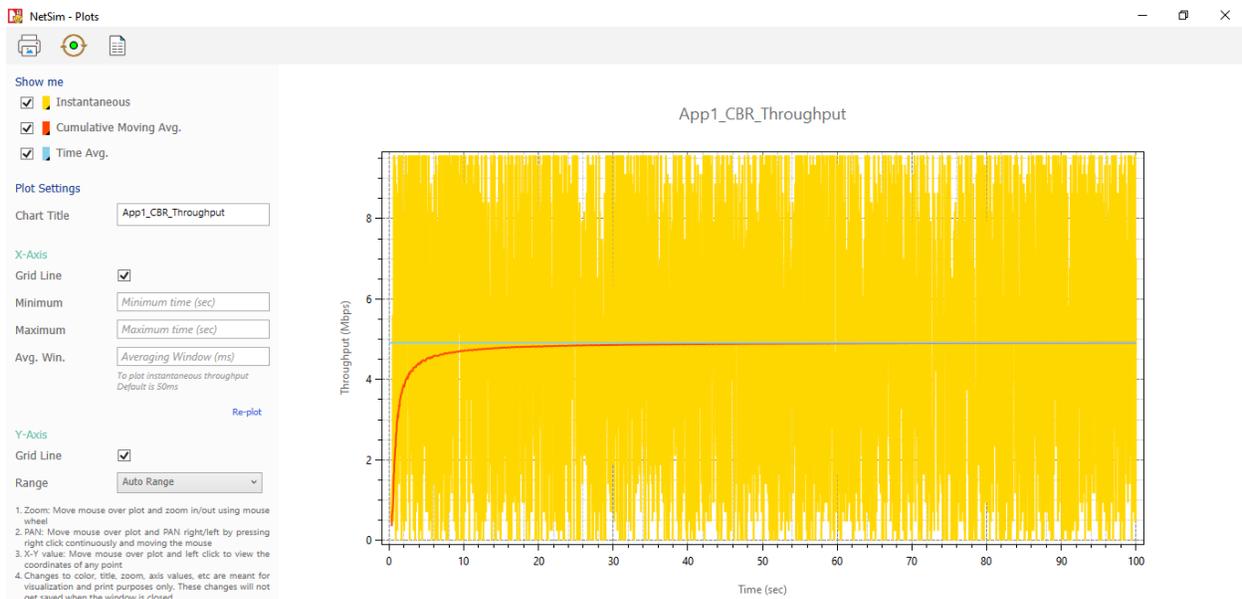


Figure 9-4 Long-term average throughput of TCP New Reno over a loss-less link

We have enabled Wireshark Capture in the Wired Node 1. The PCAP file is generated at the end of the simulation. From the PCAP file, the congestion window evolution graph can be obtained as follows. In Wireshark, select any data packet with a left click, then, go to **Statistics > TCP Stream Graphs > Window Scaling > Select Switch Direction**. In **Figure 9-5**, we report the congestion window evolution of TCP New Reno over the loss-less link.

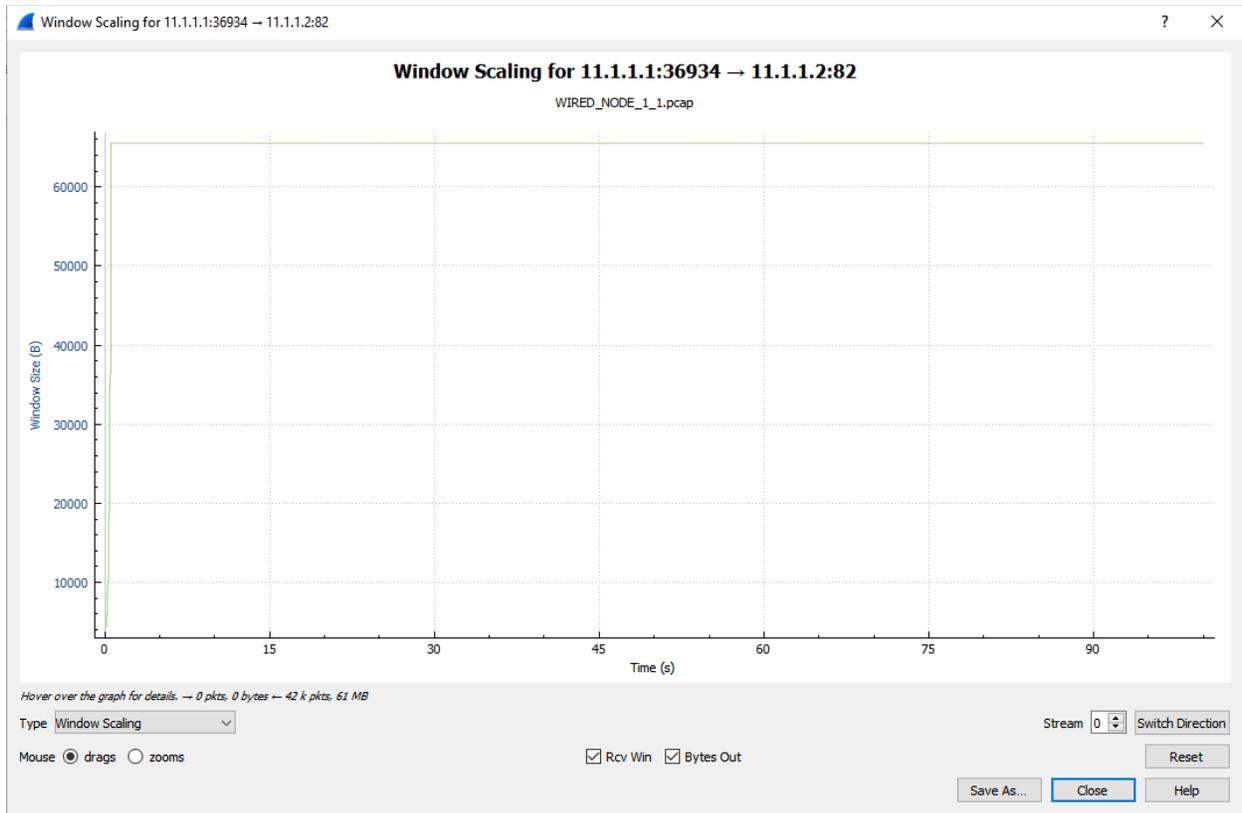


Figure 9-5: Congestion window evolution with TCP New Reno over a loss-less link.

In **Figure 9-6**, we report the application metrics data for data transfer over a lossy link (Sample-2).

Application Id	Throughput Plot	Application Name	Packet generated	Packet received	Throughput (Mbps)	Delay(microsec)	Jitter(microsec)
1	Application Throughput plot	App1_CBR	85617	32589	3.806395	31884960.015729	1976.200182

Figure 9-6: Application Metrics when $BER = 1 \times 10^{-7}$

In **Figure 9-7**, we report the plot of long-term average throughput of the TCP connection over the lossy link.

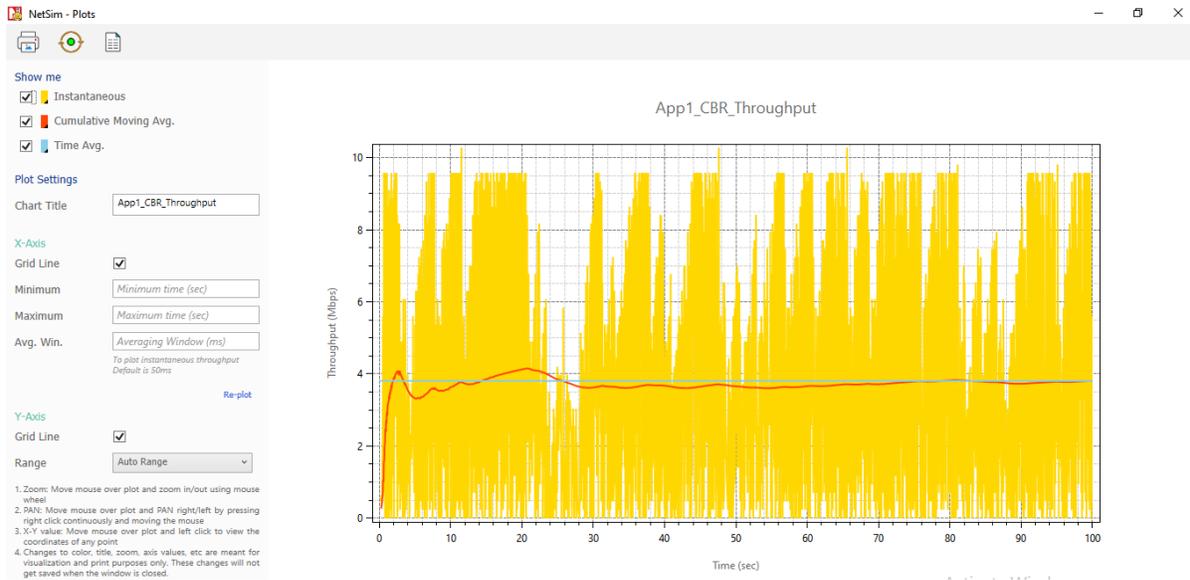


Figure 9-7: Throughput graph

In **Figure 9-8**, we report the congestion window evolution of TCP New Reno over the lossy link.



Figure 9-8 Congestion window evolution with TCP New Reno over a lossy link

9.5 Observations and Inference

1. In **Figure 9-5**, we notice that the congestion window of TCP (over the loss-less link) increases monotonically to 64 KB and remains there forever. So, a block of 64 KBs of data is transferred

over a round-trip time (RTT) of approximately 100 milliseconds. Hence, a good approximation of the TCP throughput over the loss-less link is

$$\begin{aligned} \text{Throughput} &\approx \frac{\text{Window Size (in bits)}}{\text{RTT (in secs)}} \\ &= \frac{65535 \times 8}{100 \times 10^{-3}} = 5.24 \text{ Mbps} \end{aligned}$$

We note that the observed long-term average throughput (see **Figure 9-3**) is approximately equal to the above computed value.

2. In **Figure 9-8**, for the lossy link with $BER = 1e^{-7}$, we report the congestion window evolution with New Reno congestion control algorithm. The approximate throughput of the TCP New Reno congestion control algorithm for a packet error rate p , TCP segment size MSS and round-trip time RTT

$$\begin{aligned} \text{Throughput} &\approx \sqrt{\frac{3}{2p}} \times \frac{MSS \text{ (in bits)}}{RTT \text{ (in secs)}} \\ &\approx \sqrt{\frac{3}{2 \times 1.2 \times 10^{-3}}} \times \frac{1460 \times 8}{100 \times 10^{-3}} \\ &= 4.12 \text{ Mbps} \end{aligned}$$

where the packet error rate p can be computed from the bit error rate ($BER = 1e^{-7}$) and PHY layer packet length (1500 bytes, see packet trace) as

$$p = 1 - (1 - BER)^{1500 \times 8} \approx 1.2e^{-3}$$

We note that the observed long-term average throughput (see **Figure 9-6**) is approximately equal to the above computed value.