

2 Understand Measures of Network Performance: Throughput and Delay

2.1 Introduction

The two main performance measures of a network are:

- **Throughput:** how many bits per second are going through the network
- **Delay:** how long does it take a bit from one end to the other

These are two orthogonal concepts, and one could think of it as width of a pipe and length of a pipe through with data flows.

2.1.1 Throughput

In general terms, throughput is the rate of production or the rate at which something is processed. When used in the context of communication networks, such as Ethernet or packet radio, throughput or network throughput is the rate of successful message delivery over a communication channel.

Throughput is related to other quantities like bandwidth or data-rate of a link. A link can have a certain "nominal" bandwidth or data-rate to send data at, however, all of it may not be used all the time to send useful bits. You may also have packet losses and retransmissions. Throughput measures the number of useful bits delivered at the receiver and is different from but related to the individual link data rates.

The throughput of a network is limited by the link with the slowest throughput along the path, the bottleneck link. You cannot pump data faster than the rate of the slowest link. Note that the bottleneck link need not always be the link with the slowest nominal data-rate. Sometimes a high-speed link may be shared by several flows, causing each flow to receive a small share, thus becoming the bottleneck. In other cases, you may not always be able to send at the bottleneck rate, because your protocol may have other delays, like waiting for ACKs. So, while instantaneous throughput can be the bottleneck link rate, average throughput may be lower. The way to compute average throughput is always: see the data sent over a period of time and get the ratio. A file of size F takes T units of time to be transferred. Average throughput is F/T .

2.1.2 Delay

The end-to-end delay in a path is sum of delays on all links and intermediate nodes. There are components to delay.

When a packet leaves a node, it first experiences transmission delay. That is, all the bits of a packet that have to be put out on the link. If a link can transmit data at R bits/s, a packet of size B bits will require B/R seconds to be just put out there.

Next is propagation delay. That is, the bits have to propagate at the speed of waves in the transmission medium to reach the other end. This delay depends on the length of the wire and is usually only significant for long distance links. If d is the distance the wave has to travel is s is the speed in the medium, the propagation delay is d/s . The speed of light is 3×10^8 m/s in free space and hence a radio wave takes 1 microsec to traverse a distance of 300 metres. The speed of light in copper is around 2×10^8 m/s, and it would take about 10 ns to travel a 2-meter-long wire.

If propagation delay is less than the transmission delay, then the first bit of the packet would have reached the other end point before the sender finishes putting all bits on the wire. Hence the limiting factor is really how fast the link is. On the other hand, if propagation delay is greater than transmission delay, as is the case for long distance links, then the first bit reaches the other end point much after the last bit has been sent.

Next, once the packet arrives at the other end point, it must be processed by the switch or router. This processing delay could involve looking up routing tables, computations of header checksums etc. Again, this is usually not a significant component with today's high-speed hardware.

Once an intermediate point processes the packet and decides which link to send it on, the packet may potentially be queued until the next link becomes free. This delay is called the queueing delay. This is the most unpredictable part of the delay, as it depends on traffic sent by other nodes. A large branch of study called *Queueing theory* is devoted to modelling and understanding this delay under various conditions. Internet traffic is often bursty, and hence queueing delays occur even if the aggregate traffic is less than the capacity of the links on an average. That is, suppose incoming packets arrive at an aggregate rate of L bits/s and link rate is R bits/s, then as long as $L < R$, it appears that there should be no queueing. However, packets don't arrive in an equally spaced fashion, and the arrival pattern is often random. In such cases, the queueing delay maybe high even if $\frac{L}{R} < .1$. In fact, queueing delay increases quite steeply as L/R approaches 1. It is approximately equal to $\frac{1}{(R-L)}$. Usually, network designers try to keep this ratio well below 1.

Once the packet gets out of the queue and gets ready for transmission, the cycle begins again with the transmission delay on the next link. So we add one of each of the 4 delays for every link traversed. Some switches can also start transmission even before reception fully completes. But most often, switches today are store-and-forward. That is, they wait for entire packet to arrive, then start forwarding. Once a queue is full, it may also drop packets, leading to losses. Losses can also occur due to transmission errors on the wire. This is more common in wireless links; wired links are pretty reliable.

2.2 NetSim Simulation Setup

Open NetSim and click **Examples > Experiments > Understanding-Measure-of-Network-Performance-Throughput-and-Delay**

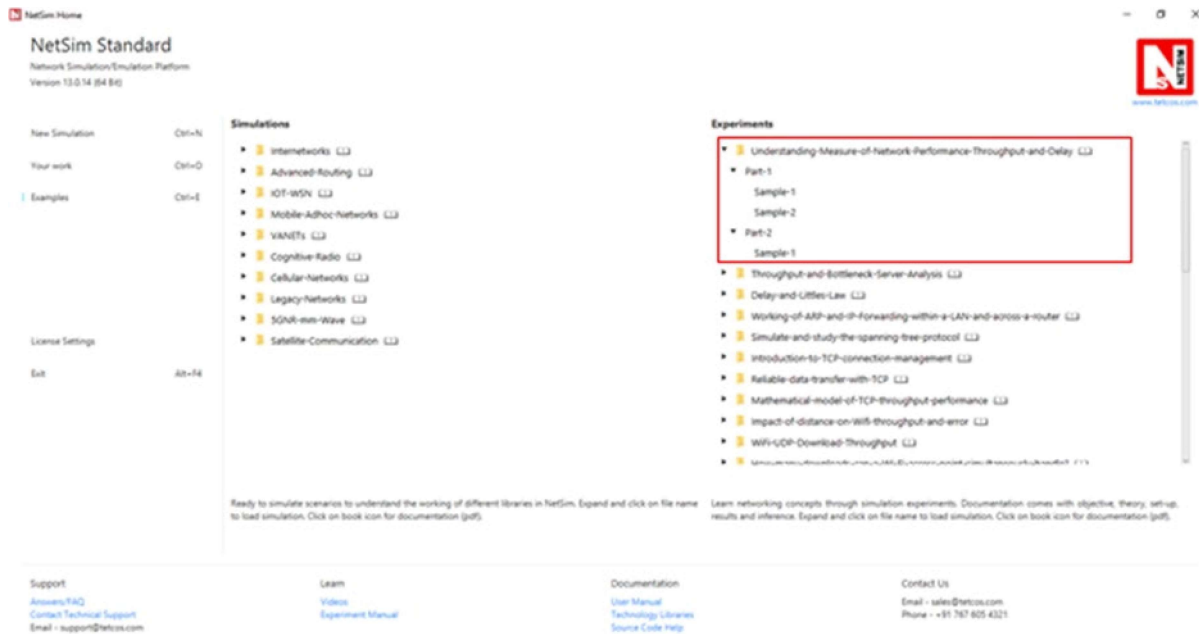


Figure 2-1: Experiments List

2.3 Part-1: Throughput Analysis

2.3.1 Without packet acknowledgement (UDP)

Sample 1: Consider a 125 KB file that needs to be sent through a network path. To explore the case where there are no packet acknowledgements we use the UDP transport protocol. The bottleneck bandwidth of the path is 1 Mbps. The one-way delay between sender and receiver is 20 μ s. Suppose the sender continuously sends data at the bottleneck rate, and no packets are lost and there are no retransmissions.

$$TransmissionTime = \frac{125KB}{1\text{ Mbps}} = \frac{125 \times 1000 \times 8}{1000 \times 1000\text{ bps}} = 1s$$

It will take 1 second to send the file and average throughput is 1 Mbps, which is the bottleneck bandwidth.

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

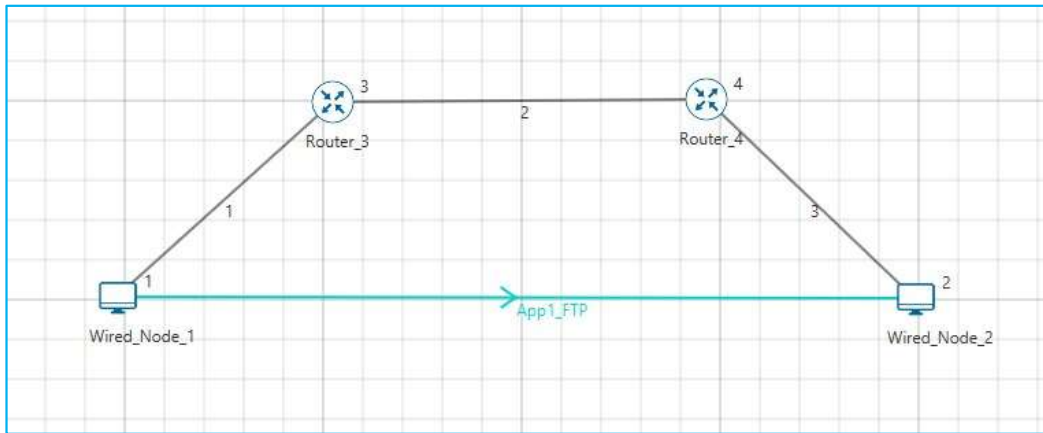


Figure 2-2: A WAN network architecture

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

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

Step 2: Right click on Wired link and select Properties, BER is set to 0, and Propagation Delay is set to 20μs. For link id 2 Link Speed is set to 1 Mbps.

Step 3: Right click on the Application Flow **App1 FTP** and select Properties or click on the Application icon present in the top ribbon/toolbar.

A FTP Application is generated from Wired Node 1 i.e. Source to Wired Node 2 i.e. Destination with File Size remaining 125000Bytes and Inter Arrival Time remaining 1s.

Transport Protocol is set to **UDP** instead of TCP.

Step 4: Enable the plots and click on Run simulation. The simulation time is set to 100 seconds

2.3.2 With Packet Acknowledgement (TCP)

Sample 2: Suppose the sender needs to wait for an ACK after sending every TCP traffic of 1 KB packet. Assume ACK also takes 20 ms to come back. Now, the sender can send 1 KB in 20 + 20 = 40 ms. Thus, the average throughput (θ) is

$$\theta = \frac{1 \times 8 \times 1000 \text{ bits}}{40 \text{ ms}} = 200 \text{ kbps}$$

Notice that the average throughput is one-fifth of what it was before, with the new ACK requirement. And the time taken to send the file will be 5 times larger, i.e 5 seconds. You can also compute 5 seconds as follows: 1 KB takes 40 ms, so 125 KB takes

$$= 125 \times 40 \text{ ms} = 5 \text{ s}$$

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

Step 1: Right click on Wired link and select Properties, BER is set to 0, and Propagation Delay is set to 40μs. For link id 2 Link Speed is set to 1 Mbps.

Step 2: Right click on the Application Flow **App1 FTP** and select Properties or click on the Application icon present in the top ribbon/toolbar.

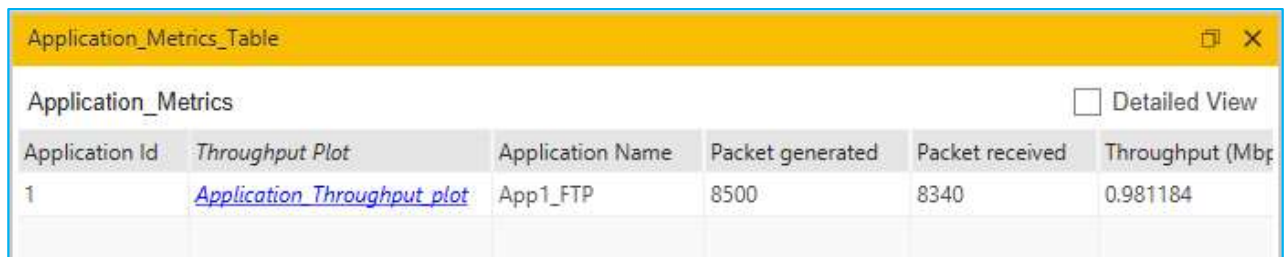
A FTP Application is generated from Wired Node 1 i.e. Source to Wired Node 2 i.e. Destination with File Size remaining 125000Bytes and Inter Arrival Time remaining 5s.

Transport Protocol is set to **TCP**.

Step 3: Enable the plots and click on Run simulation. The simulation time is set to 100 seconds.

2.3.3 Output

Sample 1:

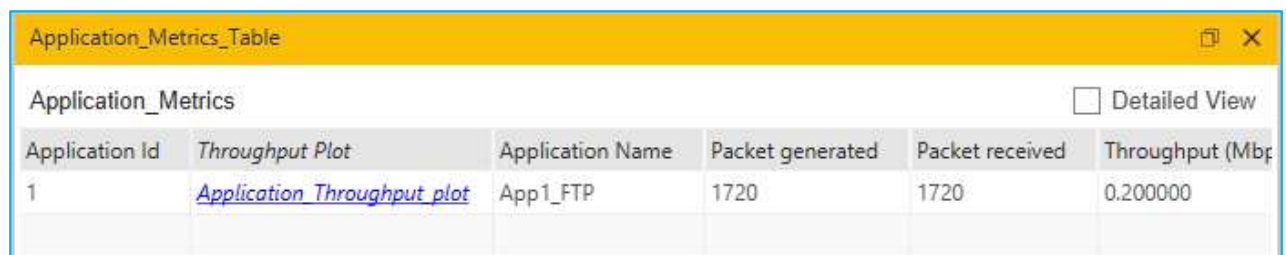


The screenshot shows a window titled 'Application_Metrics_Table' with a yellow header bar. Below the header is a table with the following data:

Application Id	Throughput Plot	Application Name	Packet generated	Packet received	Throughput (Mbps)
1	Application Throughput plot	App1_FTP	8500	8340	0.981184

Figure 2-3: Application Throughput for Sample 1

Sample 2:



The screenshot shows a window titled 'Application_Metrics_Table' with a yellow header bar. Below the header is a table with the following data:

Application Id	Throughput Plot	Application Name	Packet generated	Packet received	Throughput (Mbps)
1	Application Throughput plot	App1_FTP	1720	1720	0.200000

Figure 2-4: Application Throughput for Sample 2

2.4 Part - 2: Delay Analysis

2.4.1 Procedure

Sample 1: Consider the above A--S--B problem. Suppose A wants to send a 1MB file to B. A will divide the 1MB file into 1480-byte (standard UDP packet size) packets.

$$\text{Number of packets} = \frac{1000000}{1480} = 675 \text{ packets of size 1480} + \text{last packet of size 1054}$$

To these packets a 54-byte header is added. This makes the total packet size as 1534B or $1534 \times 8 = 12,272 \text{ bits}$. A packet of size 12,272 bits would take 12,272 μs of time to be transmitted over a 1Mbps (mega bit per second) link. Next, let us compute end-to-end delays. For now, let's ignore propagation and processing delays, as they are small.

A sends first 1534-byte packet in 12.27 ms and 1054-bytes packet in 8.43ms. While S forwards this packet to B, A can send the next packet to S (switches can send packets on one port while receiving them on another port.).

$$\text{File transmission time per link} = 675 \times 12272 + 1 \times 1054 \times 8 = 8.29 \text{ sec}$$

Thus, A takes 8.29 second to send all the packets to S. And a similar amount of time is taken by S to send the file to B. Therefore the total time to send the file from A to B would be

$$T_{total} = \text{Transmission Time per link} \times 2 = 8.29 \times 2 = 16.58 \mu s$$

This is now simulated in NetSim. The GUI open the configuration file corresponding to this experiment as shown below **Figure 2-5**

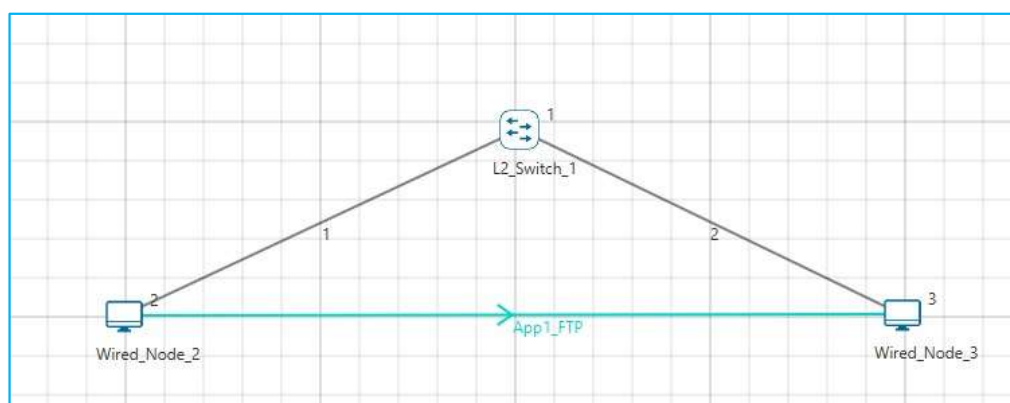


Figure 2-5: A LAN network

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

Step 1: A network scenario is designed in NetSim GUI comprising of 1 L2 Switch, and 2 Wired Node in the “**Internetworks**” Network Library.

Step 2: Right click on Wired link and select Properties, Link Speed is set to 1 Mbps, BER is set to 0, and Propagation Delay is set to 0μs.

Step 3: Right click on the Application Flow **App1 FTP** and select Properties or click on the Application icon present in the top ribbon/toolbar.

A FTP Application is generated from Wired Node 1 i.e. Source to Wired Node 2 i.e. Destination with File Size remaining 1000000Bytes and Inter Arrival Time remaining 100s.

Transport Protocol is set to **UDP** instead of TCP.

Step 4: Enable the packet trace and plots. Click on Run simulation. The simulation time is set to 100 seconds.

2.4.2 Output

Sample 1: In packet trace we can see only one file is generated from source to Destination, the file is divided into packets. Filter the packet type as FTP to calculate

End to end delay = PHY_LAYER_END_TIME - PHY_LAYER_ARRIVAL_TIME

Sending 1 MB file on 1 Mbps link should take 8.29s and the same is seen in the packet trace. Then it takes another 8.29s to go from the switch to then node, or 16.58s total see **Figure 2-6**.

	TX_LAYER_ARRIVAL_TIME[US]	PHY_LAYER_ARRIVAL_TIME[US]	MAC_LAYER_ARRIVAL_TIME[US]	PHY_LAYER_ARRIVAL_TIME[US]	PHY_LAYER_START_TIME[US]	PHY_LAYER_END_TIME[US]	End to End Delay	APP_LAYER_PAY
1	0	0	0	0	0.96	12272.96	12272.96	12273
2	0	0	0	0	12273.92	24545.92	24546.92	12273
3	0	0	0	0	12273.96	24545.96	24546.96	12273
4	0	0	0	0	24546.88	36818.88	36818.88	12273
5	0	0	0	0	24546.92	36818.92	36818.92	12273
6	0	0	0	0	36819.84	49091.84	49091.84	12273
7	0	0	0	0	36819.88	49091.88	49091.88	12273
8	0	0	0	0	49092.8	61364.8	61365.8	12273
9	0	0	0	0	49092.84	61364.84	61365.84	12273
10	0	0	0	0	61365.76	73637.76	73638.76	12273
11	0	0	0	0	61365.8	73637.8	73638.8	12273
12	0	0	0	0	73638.72	85910.72	85911.72	12273
13	0	0	0	0	73638.76	85910.76	85911.76	12273
14	0	0	0	0	85911.68	98183.68	98184.68	12273
15	0	0	0	0	85911.72	98183.72	98184.72	12273
16	0	0	0	0	98184.64	110456.64	110457.64	12273
17	0	0	0	0	98184.68	110456.68	110457.68	12273
18	0	0	0	0	110457.6	122729.6	122730.6	12273
19	0	0	0	0	110457.64	122729.64	122730.64	12273
20	0	0	0	0	122730.56	135002.56	135003.56	12273
21	0	0	0	0	122730.6	135002.6	135003.6	12273
22	0	0	0	0	135003.52	147275.52	147276.52	12273
23	0	0	0	0	135003.56	147275.56	147276.56	12273
24	0	0	0	0	147276.48	159548.48	159549.48	12273
25	0	0	0	0	147276.52	159548.52	159549.52	12273
26	0	0	0	0				

Packet Trace Print Table(Tx-800) Print Table(Custom) X

Ready Average: 11267.31953 Count: 1350 Sum: 16555416 100%

Figure 2-6: End to End Delay from Packet Trace