1 Throughput and Bottleneck Server Analysis (Level 2)

1.1 Introduction

An important measure of quality of a network is the maximum throughput available to an application process (we will also call it a flow) in the network. **Throughput** is commonly defined as the rate of transfer of application payload through the network, and is often computed as

$$\text{Throughput} = \frac{\text{application bytes transferred}}{\text{Transferred duration}}\text{bps}$$

1.1.1 A Single Flow Scenario

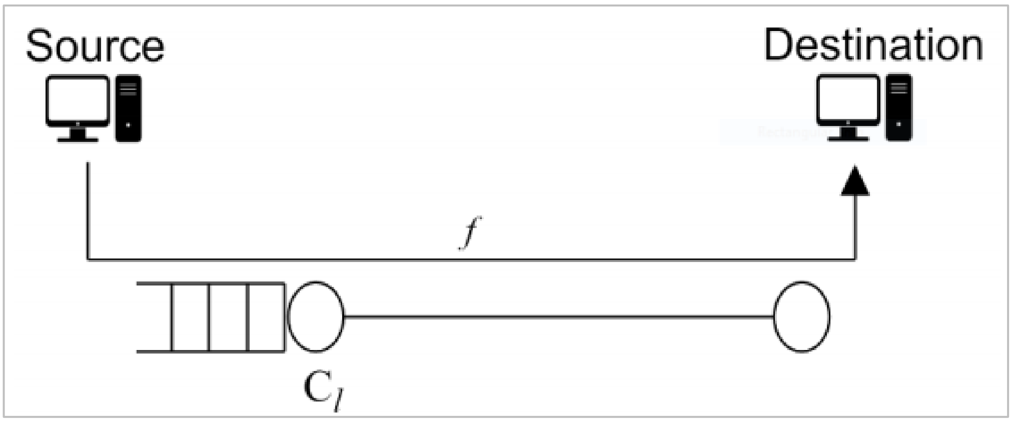


Figure 1-1: A flow $\boldsymbol{f}$ passing through a link $\boldsymbol{l}$ of fixed capacity $\boldsymbol{C_l}$.

Application throughput depends on a lot of factors including the nature of the application, transport protocol, queueing and scheduling policies at the intermediate routers, MAC protocol and PHY parameters of the links along the route, as well as the dynamic link and traffic profile in the network. A key and a fundamental aspect of the network that limits or determines application throughput is the capacity of the constituent links (capacity may be defined at MAC/PHY layer). Consider a flow f passing through a link l with fixed capacity C_l bps. Trivially, the amount of application bytes transferred via the link over a duration of T seconds is upper bounded by $C_l \times T$ bits. Hence,

$$\text{Throughput} = \frac{\text{application bytes transferred}}{\text{Transferred duration}} \le C_l\ bps$$

The upper bound is nearly achievable if the flow can generate sufficient input traffic to the link. Here, we would like to note that the actual throughput may be slightly less than the link capacity due to overheads in the communication protocols.

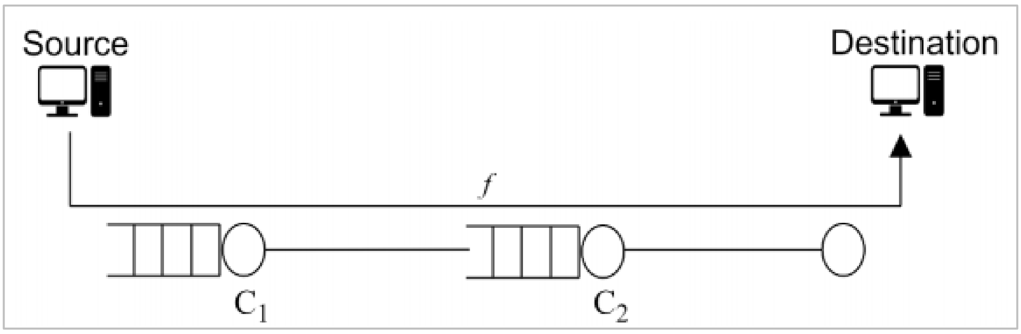


Figure 1-2: A single flow $\boldsymbol{f}$ passing through a series of links. The link with the least capacity will be identified as the bottleneck link for the flow $\boldsymbol{f}$

If a flow f passes through multiple links $l \in L_f$ (in series), then, the application throughput will be limited by the link with the least capacity among them, i.e.,

$$\text{throughput} \leq \{\min_{l \in L_f} C_l\} \text{ bps}$$

The link $l_f^* = \arg min_{l \in \mathcal{L}_f} C_l$ may be identified as the bottleneck link for the flow f. Typically, a server or a link that determines the performance of a flow is called as the bottleneck server or bottleneck link for the flow. In the case where a single flow f passes through multiple links $(\mathcal{L}_f)$ in series, the link l_f^* will limit the maximum throughput achievable and is the bottleneck link for the flow f. A noticeable characteristic of the bottleneck link is queue (of packets of the flow) build-up at the bottleneck server. The queue tends to increase with the input flow rate and is known to grow unbounded as the input flow rate matches or exceeds the bottleneck link capacity.

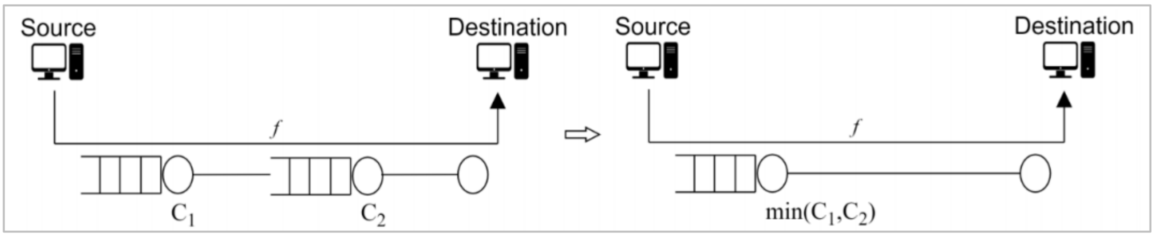


Figure 1-3: Approximation of a network using bottleneck server technique

It is a common and a useful technique to reduce a network into a bottleneck link (from the perspective of a flow(s)) to study throughput and queue buildup. For example, a network with two links (in series) can be approximated by a single link of capacity $\min(C1, C2)$ as illustrated in Figure 1-3. Such analysis is commonly known as bottleneck server analysis. Single server queueing models such as M/M/1, M/G/1, etc. can provide tremendous insights on the flow and network performance with the bottleneck server analysis.

1.1.2 Multiple Flow Scenario

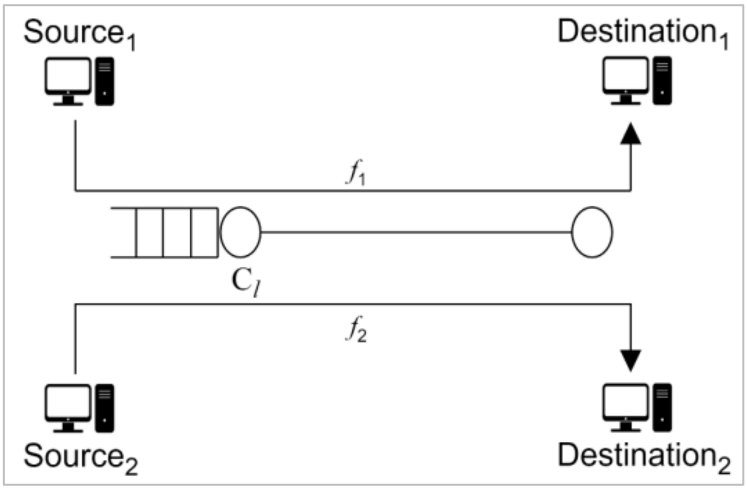


Figure 1-4: Two flows f_1 and f_2 passing through a link l of capacity C_l

Consider a scenario where multiple flows compete for the network resources. Suppose that the flows interact at some link buffer/server, say $\hat{l}$, and compete for capacity. In such scenarios, the link capacity $C_{\hat{l}}$ is shared among the competing flows and it is quite possible that the link can become the bottleneck link for the flows (limiting throughput). Here again, the queue tends to increase with the combined input flow rate and will grow unbounded as the combined input flow rate matches or exceeds the bottleneck link capacity. A plausible bound of throughput in this case is (under nicer assumptions on the competing flows)

$$\text{throughput} = \frac{C_{\hat{l}}}{\text{number of flows competing for capacity at link } \hat{l}} \; bps$$

1.2 NetSim Simulation Setup

Open NetSim and click on **Experiments> Internetworks> Network Performance> Throughput and Bottleneck Server Analysis** then click on the tile in the middle panel to load the example as shown in below Figure 1-5.

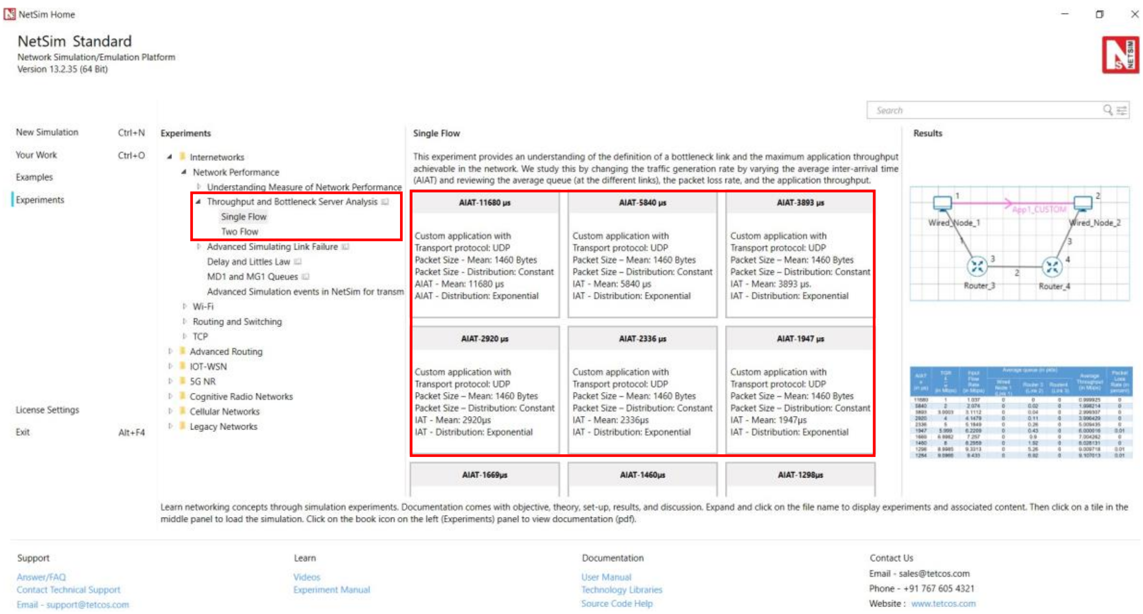


Figure 1-5: List of scenarios for the example of Throughput and Bottleneck Server Analysis

1.3 Part-1: A Single Flow Scenarios

We will study a simple network setup with a single flow illustrated in Figure 1-6 to review the definition of a bottleneck link and the maximum application throughput achievable in the network. An application process at Wired_Node_1 seeks to transfer data to an application process at Wired_Node_2. We consider a custom traffic generation process (at the application) that generates data packets of constant length (say, L bits) with i,i,d. inter-arrival times (say, with average inter-arrival time v seconds). The application traffic generation rate in this setup is $\frac{L}{v}$ bits per second. We prefer to minimize the communication overheads and hence, will use UDP for data transfer between the application processes.

In this setup, we will vary the traffic generation rate by varying the average inter-arrival time v and review the average queue at the different links, packet loss rate and the application throughput.

1.3.1 Procedure

We will simulate the network setup illustrated in Figure 1-6 with the configuration parameters listed in detail in Table 1-1 to study the single flow scenario.

NetSim UI displays the configuration file corresponding to this experiment as shown below:

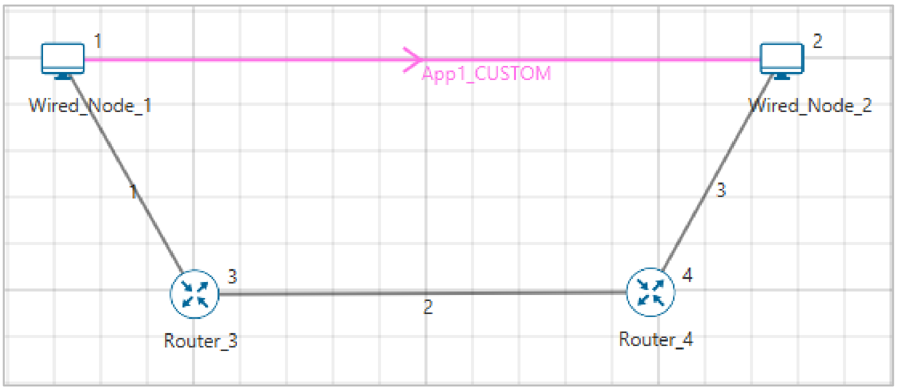


Figure 1-6: Network set up for studying a single flow

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

Step 1: Drop two wired nodes and two routers onto the simulation environment. The wired nodes and the routers are connected with wired links as shown in (See Figure 1-6).

Step 2: Click the Application icon to configure a custom application between the two wired nodes. In the Application configuration dialog box (see Figure 1-7), select Application Type as **CUSTOM**, **Source ID** as **1** (to indicate Wired_Node_1), **Destination ID** as **2** (to indicate Wired_Node_2) and **Transport Protocol** as **UDP**. In the PACKET SIZE tab, select **Distribution** as **CONSTANT** and Value as **1460 bytes**. In the INTER ARRIVAL TIME tab, select **Distribution** as **EXPONENTIAL** and **Mean** as **11680** microseconds.

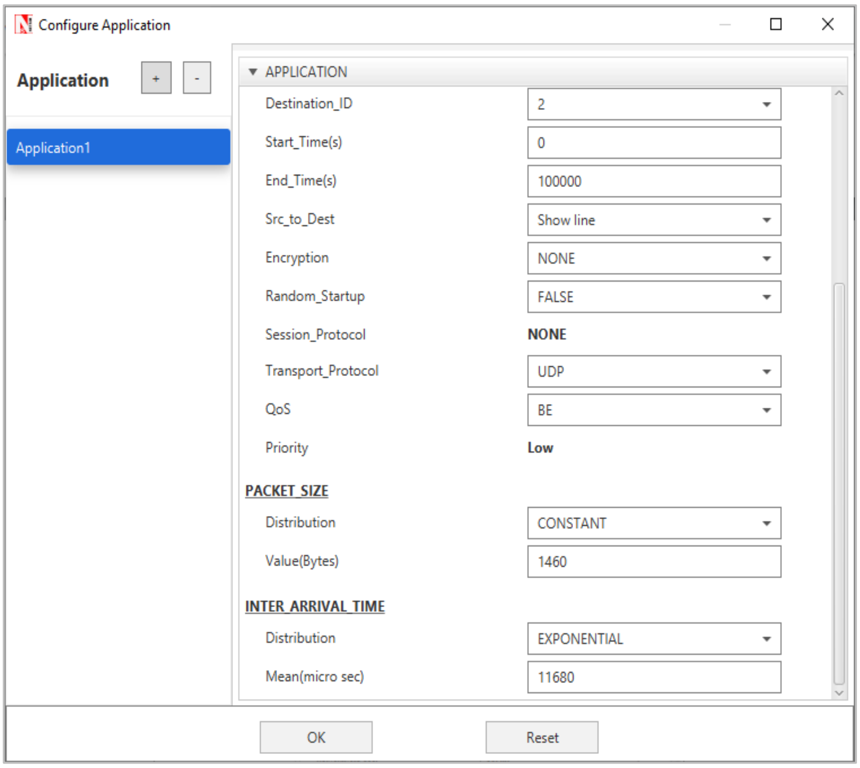


Figure 1-7: Application configuration dialog box

Step 3: The properties of the wired nodes are left to the default values.

Step 4: Right-click the link ID (of a wired link) and select **Properties** to access the link's properties dialog box (see Figure 1-8). Set **Max Uplink Speed** and **Max Downlink Speed** to 10 Mbps for link 2 (the backbone link connecting the routers) and 1000 Mbps for links 1 and 3 (the access link connecting the Wired_Nodes and the routers).

Set **Uplink BER** and **Downlink BER** as 0 for links 1, 2 and 3. Set **Uplink_Propagation Delay** and **Downlink_Propagation_Delay** as 0 microseconds for the two-access links 1 and 3 and 100 microseconds for the backbone link 2.

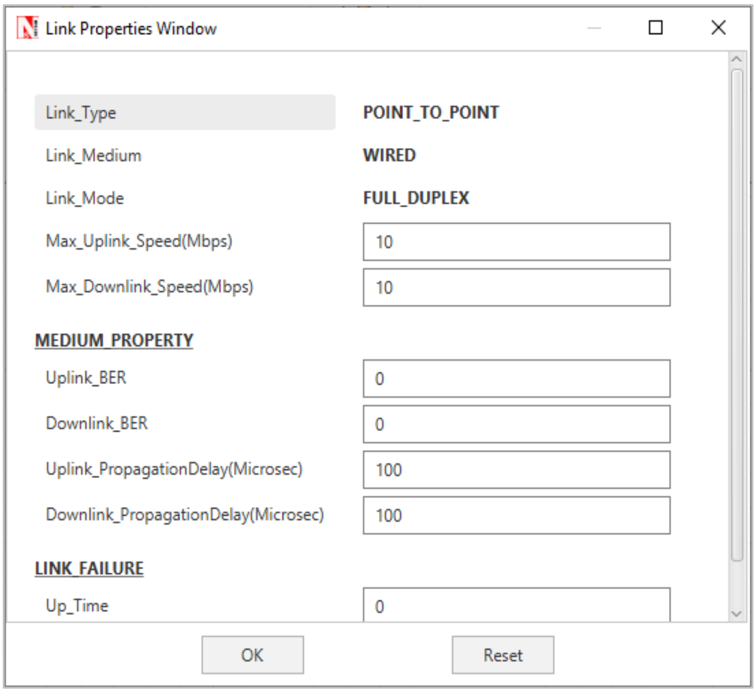


Figure 1-8: Link Properties dialog box

Step 5: Right-click **Router 3** icon and select **Properties** to access the link's properties dialog box (see Figure 1-9). In the **INTERFACE 2 (WAN)** tab, select the **NETWORK LAYER** properties, set **Buffer size (MB)** to **8.**

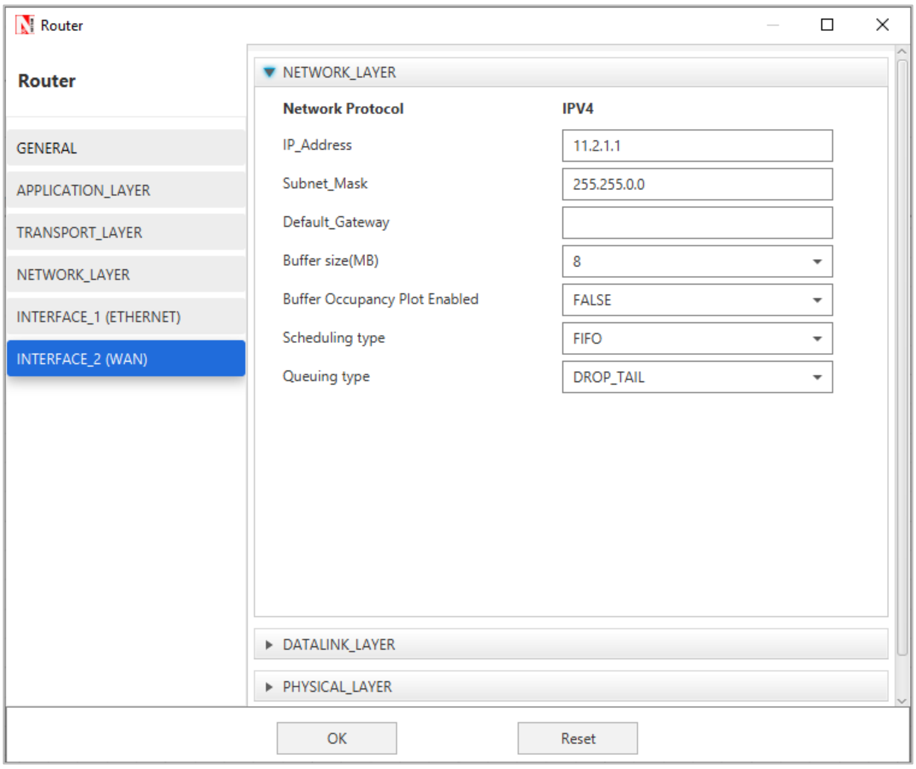


Figure 1-9: Router Properties dialog box

Step 6: Click on Packet Trace option and select the **Enable Packet Trace** check box. Packet Trace can be used for packet level analysis and Enable Plots in GUI.

Step 7: Click on **Run** icon to access the Run Simulation dialog box (see Figure 1-10) and set the **Simulation Time** to 100 seconds in the **Simulation Configuration** tab. Now, run the simulation.

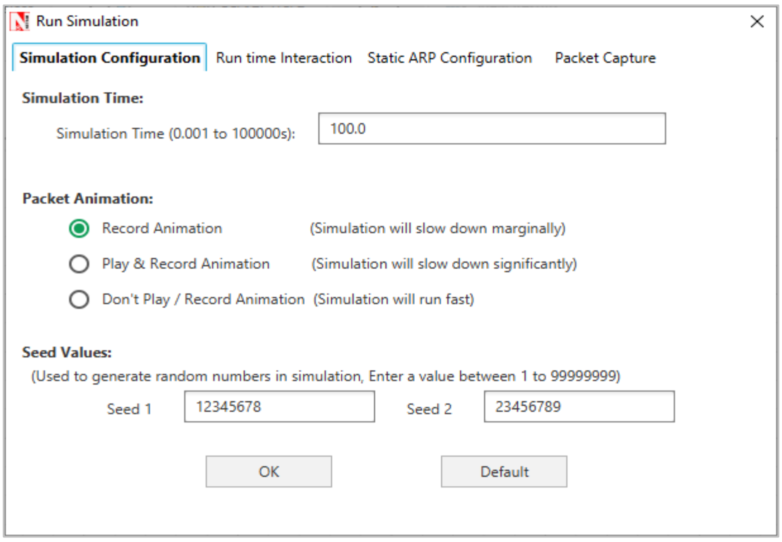


Figure 1-10: Run Simulation dialog box

Step 8: Now, repeat the simulation with different average inter-arrival times (such as 5840 µs, 3893 µs, 2920 µs, 2336 µs and so on). We vary the input flow rate by varying the average inter-arrival time. This should permit us to identify the bottleneck link and the maximum achievable throughput.

The detailed list of network configuration parameters is presented in (See Table 1-1).

Parameter	Value
LINK PARAMETERS	
Wired Link Speed (access link)	1000 Mbps
Wired Link Speed (backbone link)	10 Mbps
Wired Link BER	0
Wired Link Propagation Delay (access link)	0
Wired Link Propagation Delay (backbone link)	100 µs
APPLICATION PARAMETERS	
Application	Custom
Source ID	1
Destination ID	2
Transport Protocol	UDP
Packet Size – Value	1460 bytes
Packet Size – Distribution	Constant
Inter Arrival Time – Mean	AIAT (µs) Table 1-2
Inter Arrival Time – Distribution	Exponential
ROUTER PARAMETERS	
Buffer Size	8
MISCELLANEOUS	
Simulation Time	100 Sec
Packet Trace	Enabled
Plots	Enabled

Table 1-1: Detailed Network Parameters

1.3.2 Performance Measure

In Table 1-2, we report the flow average inter-arrival time v and the corresponding application traffic generation rate, input flow rate (at the physical layer), average queue at the three buffers (of Wired_Node_1, Router_3 and Router_4), average throughput (over the simulation time) and packet loss rate (computed at the destination).

Given the average inter-arrival time v and the application payload size L bits (here, 1460×8 = 11680 bits), we have,

$$\text{Traffic generation rate} = \frac{L}{v} = \frac{11680}{v} bps$$

$$\text{input flow rate} = \frac{11680 + 54 * 8}{v} = \frac{12112}{v} bps$$

where the packet overheads of 54 bytes is computed as $54 = 8(UDP\ header) + 20(IP\ header) + 26(MAC + PHY\ header)\ bytes$. Let $Q_l(u)$ as denote the instantaneous queue at link l at time u . Then, the average queue at link l is computed as

$$\text{average queue at link } l = \frac{1}{T}\int_0^T Q_l\ (u)\ \ du\ bits$$

where, T is the simulation time. The average throughput of the flow is computed as

$$\text{throughput} = \frac{\text{application byte transferred}}{T}\ bps$$

The packet loss rate is defined as the fraction of application data lost (here, due to buffer overflow at the bottleneck server).

$$\text{packet loss rate} = \frac{\text{application bytes not received at destination}}{\text{application bytes transmitted at source}}$$

1.3.2.1 Average Queue Computation from Packet Trace

- Open Packet Trace file using the **Open Packet Trace** option available in the Simulation Results window.
- Click on below highlighted icon to create new Pivot Table.

1	PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID
2	1	0	Custom	App1_CUSTOM	NODE-1	NODE-2	NODE-1
3	0	0	Control_Packet	OSPF_HELLO	ROUTER-3	Broadcast-0	ROUTER-3
4	0	0	Control_Packet	OSPF_HELLO	ROUTER-4	Broadcast-0	ROUTER-4
5	1	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-3
6	1	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-4
7	2	0	Custom	App1_CUSTOM	NODE-1	NODE-2	NODE-1
8	2	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-3
9	2	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-4
10	3	0	Custom	App1_CUSTOM	NODE-1	NODE-2	NODE-1
11	3	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-3
12	3	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-4
13	4	0	Custom	App1_CUSTOM	NODE-1	NODE-2	NODE-1
14	4	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-3
15	4	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-4
16	5	0	Custom	App1_CUSTOM	NODE-1	NODE-2	NODE-1
17	6	0	Custom	App1_CUSTOM	NODE-1	NODE-2	NODE-1
18	5	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-3
19	5	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-4
20	6	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-3
21	6	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-4
22	7	0	Custom	App1_CUSTOM	NODE-1	NODE-2	NODE-1
23	7	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-3
24	7	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-4
25	8	0	Custom	App1_CUSTOM	NODE-1	NODE-2	NODE-1
26	9	0	Custom	App1_CUSTOM	NODE-1	NODE-2	NODE-1
27	8	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-3
28	8	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-4
29	9	0	Custom	App1_CUSTOM	NODE-1	NODE-2	ROUTER-3

Packet Trace | Pivot Table(TX-RX) | Pivot Table(Custom)

Figure 1-11: Packet Trace

- Click on Insert on Top ribbon → Select Pivot Table.

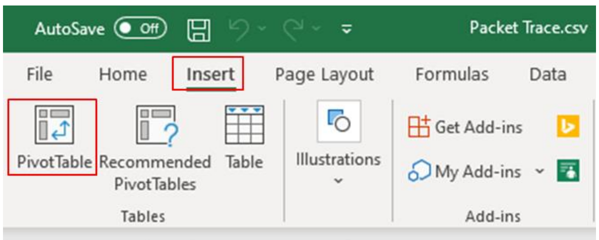


Figure 1-12: Top Ribbon

- Then select packet trace and press Ctrl + A → Select ok

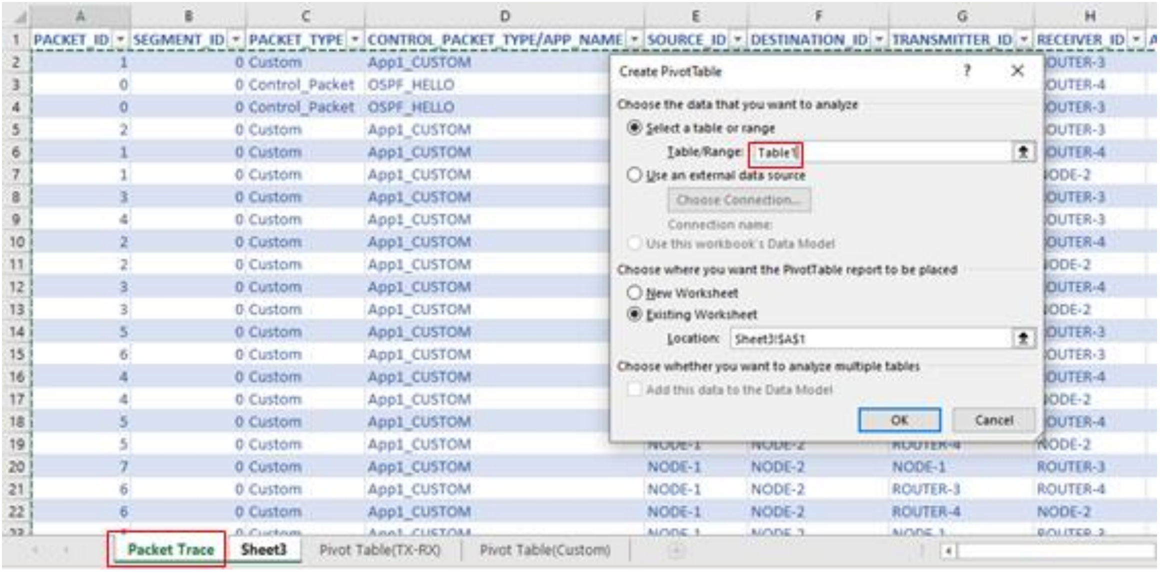


Figure 1-13: Packet Trace Pivot Table

- Then we will get blank Pivot table.

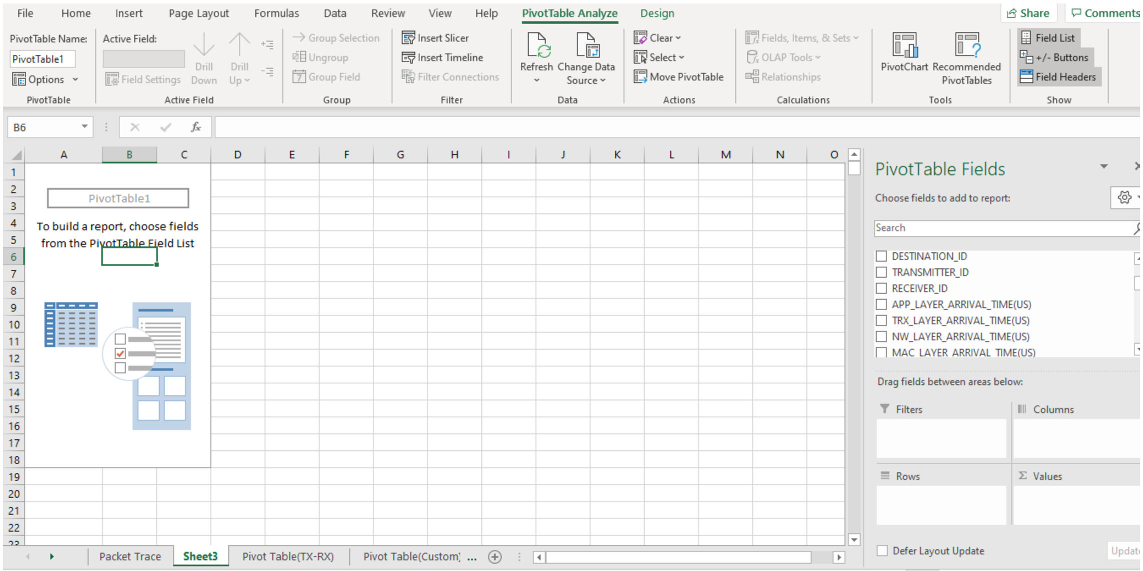


Figure 1-14: Blank Pivot Table

- **Packet ID** drag and drop into **Values field** for 2 times, **CONTROL PACKET TYPE/APP NAME, TRANSMITTER ID, RECEIVER ID** into **Filter field, NW_LAYER_ARRIVAL_TIME (US)** to **Rows field** see Figure 1-15.
- Change **Sum of PACKET ID** -> Values Field Settings **->**Select **Count -> ok** for both Values field**, CONTROL PACKET TYPE** to **APP1 CUSTOM, TRANSMITTER ID** to **Router_3 and RECEIVER ID to Router 4**

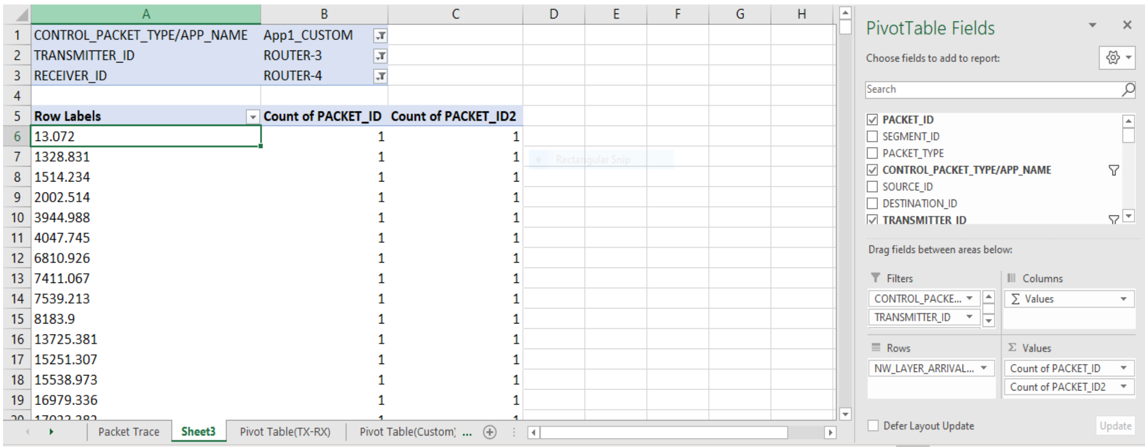


Figure 1-15: Adding fields into Filter, Columns, Rows and Values

- Right click on first value of Row Labels ->Group ->Select **By** value as 1000000.
- Go to Values field under left click on Count of PACKET ID2 ->Values Field Settings-> click on show values as -> **Running total in**-> click on OK.
- Again, create one more Pivot Table, Click on Insert on Top ribbon → Select Pivot Table.
- Then select packet trace and press Ctrl + A → Select ok
- Then we will get blank Pivot table see Figure 1-16.
- **Packet ID drag and drop into Values field for 2 times, CONTROL PACKET TYPE/APP NAME, TRANSMITTER ID, RECEIVER ID into Filter field, PHY_LAYER_ARRIVAL_TIME (US)** to **Rows field** see Figure 1-16,
- Change **Sum of PACKET ID** -> Values Field Settings **->**Select **Count -> ok** for both Values field**, CONTROL PACKET TYPE** to **APP1 CUSTOM, TRANSMITTER ID** to **Router_3 and RECEIVER ID to Router 4**
- Right click on first value of Row Labels for second Pivot Table->Group ->Select **by** value as 1000000.

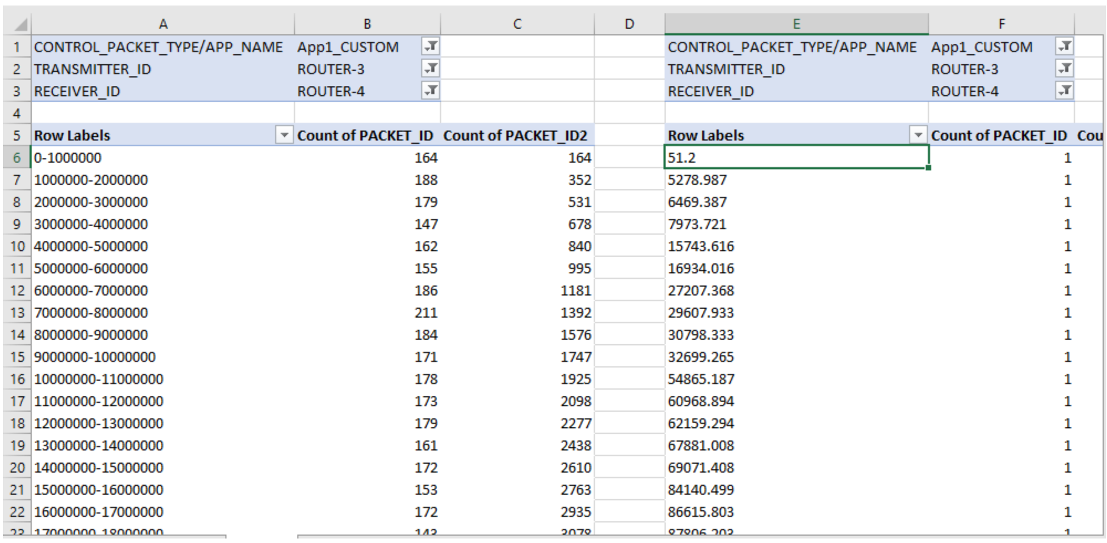
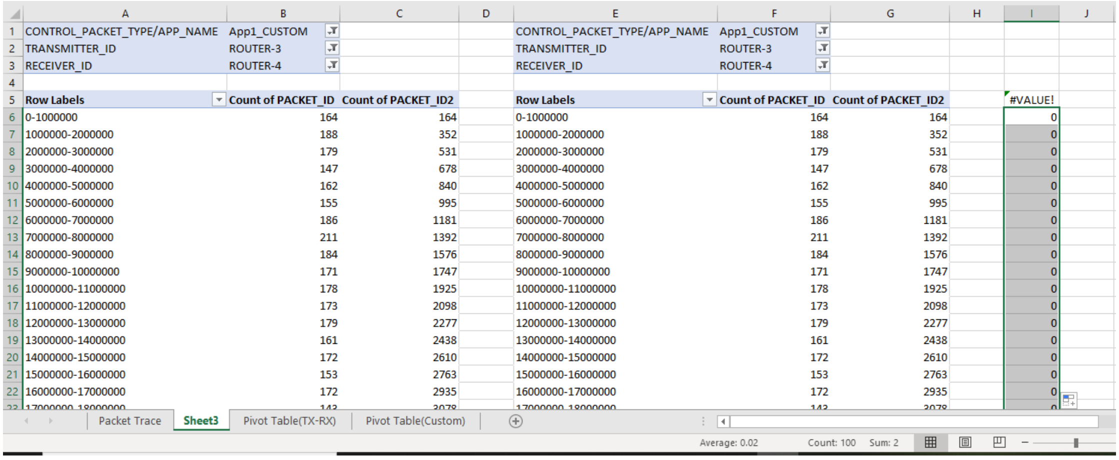


Figure 1-16: Create one more Pivot Table and Add All Fields

- Go to Values field under left click on Count of PACKET ID ->Values Field Settings-> click on show values as -> **Running total in**-> click on OK.
- Calculate the average queue by taking the mean of the number of packets in queue at every time interval during the simulation.
- The difference between the **count of PACKET ID2 (Column C)** and **count of PACKET ID2 (Column G),** Note down the average value for difference see Figure 1-17



	A	B	C	D	E	F	G	H	I
1	CONTROL_PACKET_TYPE/APP_NAME	App1_CUSTOM			CONTROL_PACKET_TYPE/APP_NAME	App1_CUSTOM			
2	TRANSMITTER_ID	ROUTER-3			TRANSMITTER_ID	ROUTER-3			
3	RECEIVER_ID	ROUTER-4			RECEIVER_ID	ROUTER-4			
4									
5	Row Labels	Count of PACKET_ID	Count of PACKET_ID2		Row Labels	Count of PACKET_ID	Count of PACKET_ID2		#VALUE!
6	0-1000000	164	164		0-1000000	164	164		0
7	1000000-2000000	188	352		1000000-2000000	188	352		0
8	2000000-3000000	179	531		2000000-3000000	179	531		0
9	3000000-4000000	147	678		3000000-4000000	147	678		0
10	4000000-5000000	162	840		4000000-5000000	162	840		0
11	5000000-6000000	155	995		5000000-6000000	155	995		0
12	6000000-7000000	186	1181		6000000-7000000	186	1181		0
13	7000000-8000000	211	1392		7000000-8000000	211	1392		0
14	8000000-9000000	184	1576		8000000-9000000	184	1576		0
15	9000000-10000000	171	1747		9000000-10000000	171	1747		0
16	10000000-11000000	178	1925		10000000-11000000	178	1925		0
17	11000000-12000000	173	2098		11000000-12000000	173	2098		0
18	12000000-13000000	179	2277		12000000-13000000	179	2277		0
19	13000000-14000000	161	2438		13000000-14000000	161	2438		0
20	14000000-15000000	172	2610		14000000-15000000	172	2610		0
21	15000000-16000000	153	2763		15000000-16000000	153	2763		0
22	16000000-17000000	172	2935		16000000-17000000	172	2935		0

Figure 1-17: Average Packets in Queue

$$\textbf{Packet Loss Rate (in percent)} = \frac{\textbf{Packet Generated} - \textbf{Packet Received}}{\textbf{Packet Generated}} \times \textbf{100}$$

1.3.3 Results

In Table 1-2, we report the flow average inter-arrival time (AIAT) and the corresponding application traffic generation rate (TGR), input flow rate, average queue at the three buffers (of Wired_Node_1, Router_3 and Router_4), average throughput and packet loss rate.

AIAT v (in µs)	TGR $\frac{L}{v}$ (in Mbps)	Input Flow Rate (in Mbps)	Average queue (in pkts) Wired Node 1 (Link 1)	Router 3 (Link 2)	Router4 (Link 3)	Average Throughput (in Mbps)	Packet Loss Rate (in percent)
11680	1	1.037	0	0	0	0.999925	0
5840	2	2.074	0	0.02	0	1.998214	0
3893	3.0003	3.1112	0	0.04	0	2.999307	0
2920	4	4.1479	0	0.11	0	3.996429	0
2336	5	5.1849	0	0.26	0	5.009435	0
1947	5.999	6.2209	0	0.43	0	6.000016	0.01
1669	6.9982	7.257	0	0.9	0	7.004262	0
1460	8	8.2959	0	1.92	0	8.028131	0
1298	8.9985	9.3313	0	5.26	0	9.009718	0.01
1284	9.0966	9.433	0	6.92	0	9.107013	0.01
1270	9.1969	9.537	0	7.98	0	9.209563	0.01
1256	9.2994	9.6433	0	7.88	0	9.314683	0
1243	9.3966	9.7442	0	11.48	0	9.416182	0.01
1229	9.5037	9.8552	0	16.26	0	9.520718	0.02
1217	9.5974	9.9523	0	25.64	0	9.616027	0.01
1204	9.701	10.0598	0	42.88	0	9.717994	0.05
1192	9.7987	10.1611	0	90.86	0	9.796133	0.26
1180	9.8983	10.2644	0	436.41	0	9.807696	1.15
1168	10	10.3699	0	847.65	0	9.808981	2.09

1062	10.9981	11.4049	0	3876.87	0	9.811667	11.00
973	12.0041	12.4481	0	4593.67	0	9.811667	18.53
898	13.0067	13.4878	0	4859.68	0	9.811667	24.75
834	14.0048	14.5228	0	5000.57	0	9.811667	30.09
779	14.9936	15.5481	0	5085.05	0	9.811667	34.75

Table 1-2: Average queue, throughput and loss rate as a function of traffic generation rate

We can infer the following from Table 1-2.

- The input flow rate is slightly larger than the application traffic generation rate. This is due to the overheads in communication.
- There is queue buildup at Router 3 (Link 2) as the input flow rate increases. So, Link 2 is the bottleneck link for the flow.
- As the input flow rate increases, the average queue increases at the (bottleneck) server at Router 3. The traffic generation rate matches the application throughput (with nearly zero packet loss rate) when the input flow rate is less than the capacity of the link.
- As the input flow rate reaches or exceeds the link capacity, the average queue at the (bottleneck) server at Router 3 increases unbounded (limited by the buffer size) and the packet loss rate increases as well.

For the sake of the readers, we have made the following plots for clarity. In Figure 1-18, we plot application throughput as a function of the traffic generation rate. We note that the application throughput saturates as the traffic generate rate (in fact, the input flow rate) gets closer to the link capacity. The maximum application throughput achievable in the setup is 9.81 Mbps (for a bottleneck link with capacity 10 Mbps).

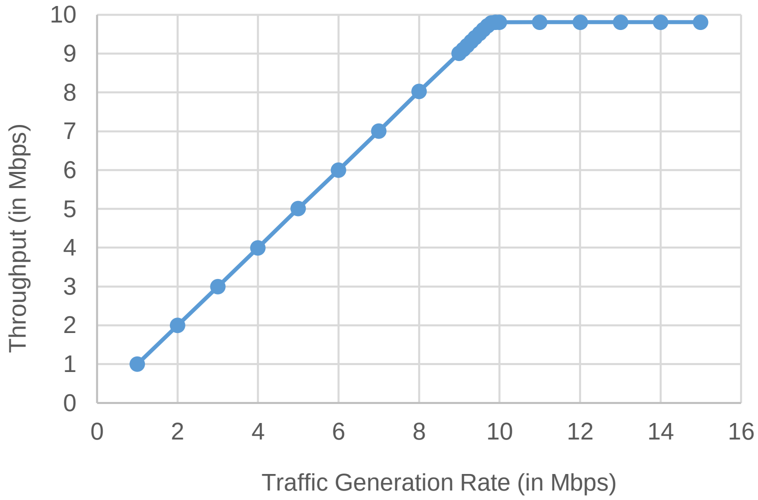
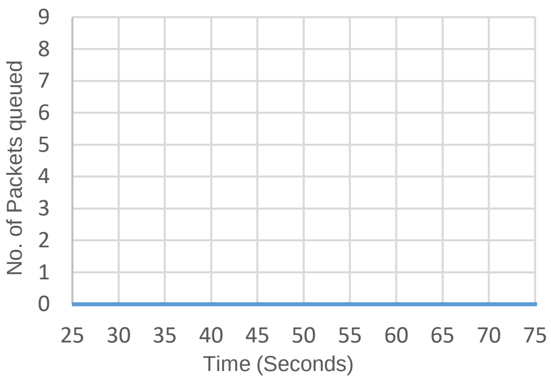
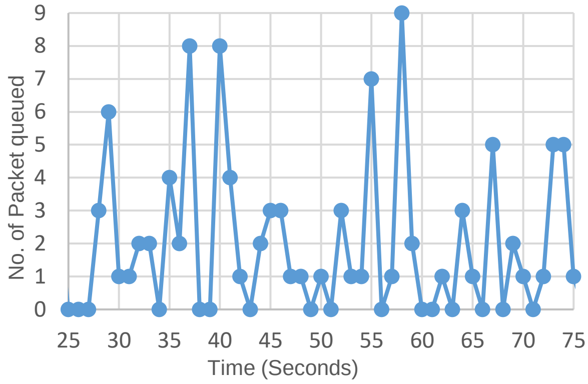


Figure 1-18: Application throughput as a function of the traffic generation rate

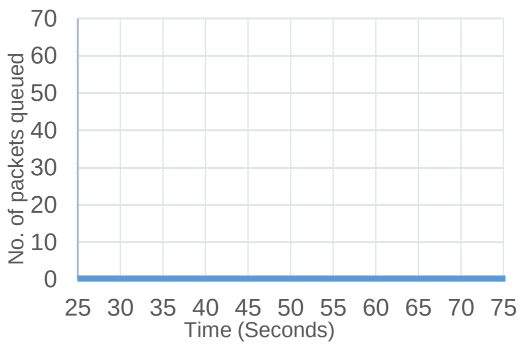
Figure 1-19, we plot the queue evolution at the buffers of Links 1 and 2 for two different input flow rates. We note that the buffer occupancy is a stochastic process and is a function of the input flow rate and the link capacity as well.



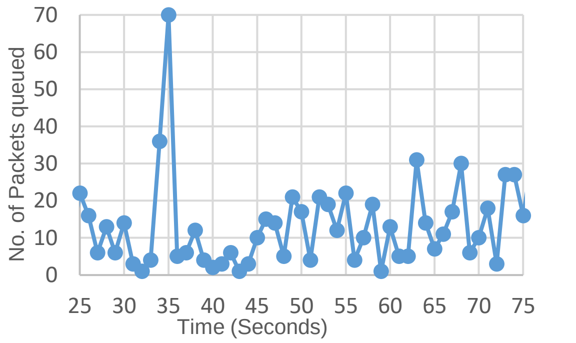
a) At Wired Node 1 for TGR = 8 Mbps



b) At Router 3 for TGR = 8 Mbps



c) At Wired Node 1 for TGR = 9.5037 Mbps



d) At Router 3 for TGR = 9.5037 Mbps

Figure 1-19: Queue evolution at Wired Node 1 (Link 1) and Router 3 (Link 2) for two different traffic generation rates

In Figure 1-20, we plot the average queue at the bottleneck link 2 (at Router 3) as a function of the traffic generation rate. We note that the average queue increases gradually before it increases unboundedly near the link capacity.

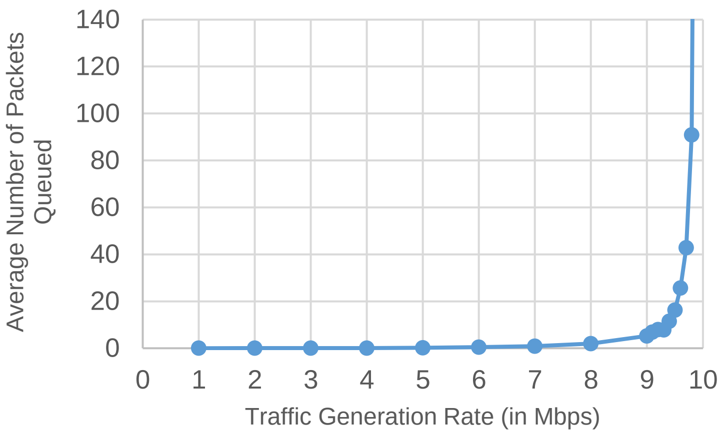


Figure 1-20: Average queue (in packets) at the bottleneck link 2 (at Router 3) as a function of the traffic generation rate

1.3.3.1 Bottleneck Server Analysis as M/G/1 Queue

Let us now analyze the network by focusing on the flow and the bottleneck link (Link 2). Consider a single flow (with average inter-arrival time v) into a bottleneck link (with capacity C). Let us the denote the input flow rate in packet arrivals per second as λ , where $\lambda = 1/v$. Let us also denote the service rate of the bottleneck server in packets served per second as μ, where $\mu = \frac{C}{L+54\times 8}$. Then,

$$\rho = \lambda \times \frac{1}{\mu} = \frac{\lambda}{\mu}$$

denotes the offered load to the server. When $\rho < 1$, ρ also denotes (approximately) the fraction of time the server is busy serving packets (i.e., ρ denotes link utilization). When $\rho \ll 1$, then the link is barely utilized. When $\rho > 1$, then the link is said to be overloaded or saturated (and the buffer will grow unbounded). The interesting regime is when $0 < \rho < 1$.

Suppose that the application packet inter-arrival time is i.i.d. with exponential distribution. From the M/G/1 queue analysis (in fact, M/D/1 queue analysis), we know that the average queue at the link buffer (assuming large buffer size) must be.

$$\text{average queue} = \rho \times \frac{1}{2}\left(\frac{\rho^2}{1-\rho}\right), \qquad 0 < \rho < 1$$

where, ρ is the offered load. In Figure 1-20, we also plot the average queue from (1) (from the bottleneck analysis) and compare it with the average queue from the simulation. You will notice that the bottleneck link analysis predicts the average queue (from simulation) very well.

An interesting fact is that the average queue depends on λ and μ only as $\rho = \frac{\lambda}{\mu}$.

1.4 Part - 2: Two Flow Scenario

We will consider a simple network setup with two flows illustrated in Figure 1-21 to review the definition of a bottleneck link and the maximum application throughput achievable in the network. An application process at Wired_Node_1 seeks to transfer data to an application process at Wired_Node_2. Also, an application process at Wired_Node_3 seeks to transfer data to an application process at Wired_Node_4. The two flows interact at the buffer of Router_ 5 (Link 3) and compete for link capacity. We will again consider custom traffic generation process (at the application processes) that generates data packets of constant length (L bits) with i.i.d. inter-arrival times (with average inter-arrival time v seconds) with a common distribution. The application traffic generation rate in this setup is $\frac{L}{v}$ bits per second (for either application).

In this setup, we will vary the traffic generation rate of the two sources (by identically varying the average inter-arrival time v) and review the average queue at the different links, application throughput (s) and packet loss rate (s).

1.4.1 Procedure

We will simulate the network setup illustrated in Figure 1-21 with the configuration parameters listed in detail in Table 1-1 to study the two-flow scenario. We will assume identical configuration parameters for the access links and the two application processes.

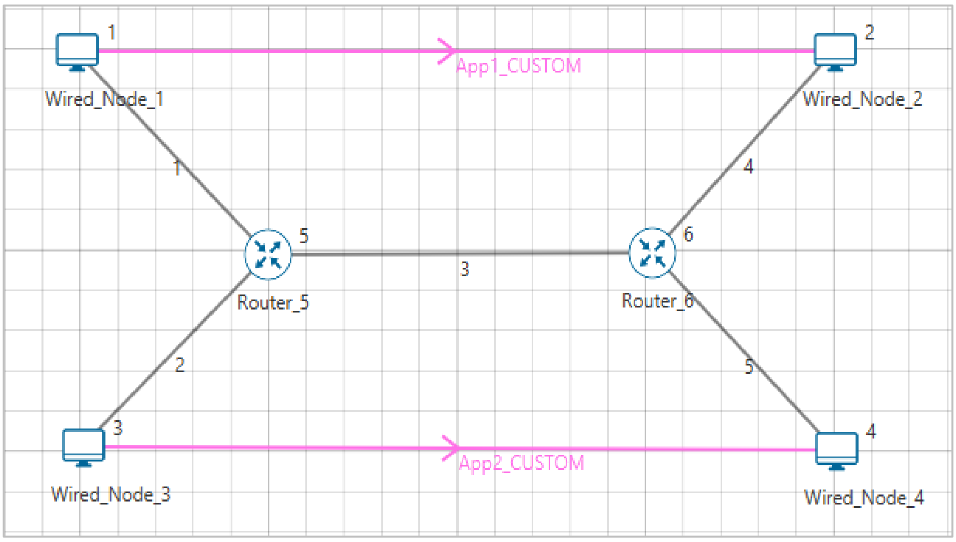


Figure 1-21: Network set up for studying two flows

Step 1: Right-click the link ID (of a wired link) and select Properties to access the link's properties dialog box. Set Max Uplink Speed and Max Downlink Speed to 10 Mbps for link 3 (the backbone link connecting the routers) and 1000 Mbps for links 1,2,4, 5 (the access link connecting the Wired Nodes and the routers). Set Uplink BER and Downlink BER as 0 for all links. Set Uplink Propagation Delay and Downlink Propagation Delay as 0 microseconds for links 1,2,4 and 5 and 100 microseconds for the backbone link 3.

Step 2: Enable Plots and Packet trace in NetSim GUI.

Step 3: Simulation time is 100 sec for all samples.

1.4.2 Results

In Table 1-3, we report the common flow average inter-arrival time (AIAT) and the corresponding application traffic generation rate (TGR), input flow rate, combined input flow rate, average queue at the buffers (of Wired_Node_1, Wired_Node_3 and Router_5), average throughput(s) and packet loss rate(s).

AIAT v (in µs)	TGR $\frac{L}{v}$ (in Mbps)	Input Flow Rate (in Mbps)	Combined Input Flow Rate (in Mbps)	Average queue (In pkts)			Average Throughput (in Mbps)		Packet Loss Rate (In percent)	
				Wired_ Node 1	Wired_ Node 2	Router	App1 Custom	App2 Custom	App1 Custom	App2 Custom
11680	1	1.037	2.074	0	0	0.03	0.999925	1.002728	0	0
5840	2	2.074	4.148	0	0	0.16	1.998214	2.006624	0	0
3893	3.0003	3.1112	6.2224	0	0	0.32	2.999307	3.001410	0	0
2920	4	4.1479	8.2958	0	0	1.99	3.996312	4.018504	0	0
2336	5	5.1849	10.3698	0	0	847.19	4.903614	4.907469	2.12	2.10
1947	5.999	6.2209	12.4418	0	0	4607.12	4.896606	4.915061	18.38	18.38
1669	6.9982	7.257	14.514	0	0	5009.33	4.896373	4.915294	30.10	30.00
1460	8	8.2959	16.5918	0	0	5150.91	4.906418	4.905250	38.88	38.78
1298	8.9985	9.3313	18.6626	0	0	5222.86	4.904782	4.906885	45.56	45.52

Table 1-3: Average queue, throughput(s) packet loss rate(s) as a function of the traffic generation

We can infer the following from

1. There is queue buildup at Router_5 (Link 3) as the combined input flow rate increases. So, link 3 is the bottleneck link for the two flows.
2. The traffic generation rate matches the application throughput(s) (with nearly zero packet loss rate) when the combined input flow rate is less than the capacity of the bottleneck link.
3. As the combined input flow rate reaches or exceeds the bottleneck link capacity, the average queue at the (bottleneck) server at Router 5 increases unbounded (limited by the buffer size) and the packet loss rate increases as well.
4. The two flows share the available link capacity and see a maximum application throughput of 4.9 Mbps (half of bottleneck link capacity 10 Mbps).

1.5 Useful Exercises

1. Redo the single flow experiment with constant inter-arrival time for the application process. Comment on average queue evolution and maximum throughput at the links.
2. Redo the single flow experiment with small buffer size (8 KBytes) at the bottleneck link 2. Compute the average queue evolution at the bottleneck link and the average throughput of the flow as a function of traffic generation rate. Compare it with the case when the buffer size in 8 MB.

3. Redo the single flow experiment with a bottleneck link capacity of 100 Mbps. Evaluate the average queue as a function of the traffic generation rate. Now, plot the average queue as a function of the offered load and compare it with the case of bottleneck link with 10 Mbps capacity (studied in the report). Comment.