NetSim®

Accelerate Network R & D

Experiments Manual

A Network Simulation & Emulation Software

By

TETCOS LLP

Contact us at

TETCOS LLP

214, 39th A Cross, 7th Main, 5th Block Jayanagar,

Bangalore - 560 041, Karnataka, INDIA. Phone: +91 80 26630624

E-Mail: sales@tetcos.com

Visit: www.tetcos.com

TABLE OF CONTENTS

1 Introduction to Network simulation and NetSim

1.1 Introduction to NetSim (Level 1)

1.1.1 Simulation environment and workflow

NetSim is a network simulation tool that allows you to create network scenarios, model traffic, design protocols and analyze network performance. Users can study the behavior of a network by testing combinations of network parameters. The various network technologies covered in NetSim include:

- Internetworks - Ethernet, WLAN, IP, TCP
- Legacy Networks - Aloha, Slotted Aloha
- Cellular Networks - GSM, CDMA
- Mobile Ad hoc Networks - DSR, AODV, OLSR, ZRP
- Wireless Sensor Networks - 802.15.4
- Internet of Things - 6LoWPAN gateway, 802.15.4 MAC / PHY, RPL
- Cognitive Radio Networks - 802.22
- Long-Term Evolution Networks – LTE
- VANETs – IEEE 1609
- 5G NR - LTE NR
- Satellite Communication Networks - TDMA
- Software Defined Networking – Open flow protocol
- Advanced Routing and Switching - VLAN, PIM, L3 Switch, ACL and NAT
- UWAN – Slotted Aloha
- 5G NTN - LEO/MEO/GEO satellite networks , Downlink transmission

The NetSim home screen is as shown below see Figure 1-1. Click on the network type you wish to simulate.

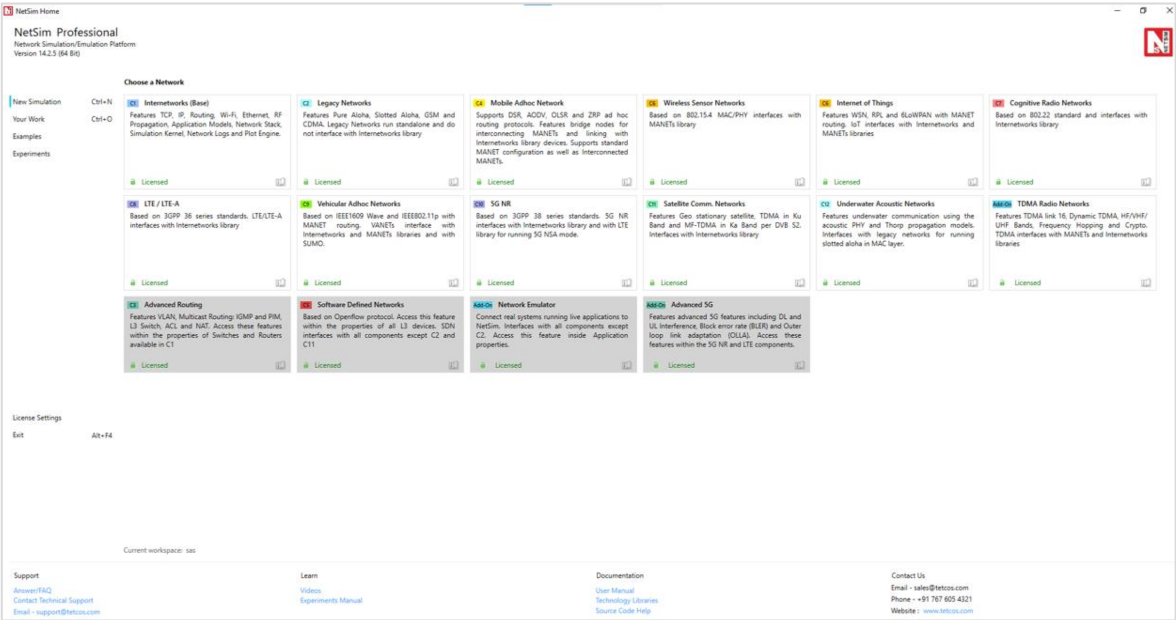


Figure 1-1: NetSim Home Screen

Network Design Window: A user would enter the design window upon selecting a network type in the home screen. The NetSim design window GUI see Figure 1-2. It enables users to model a network comprising of network devices like switches, routers, nodes, etc., connect them through links, and model application traffic to flow through the network. The network devices shown in the palette are specific to the network technologies chosen by the user.

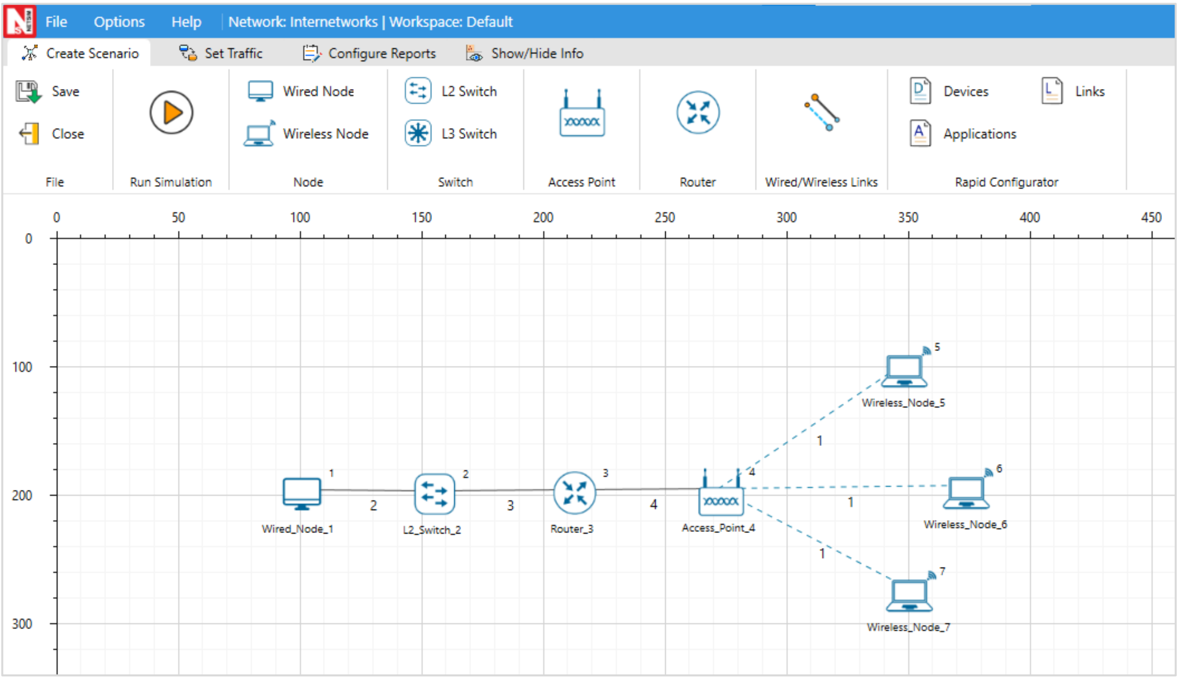


Figure 1-2: Network Design Window

Description:

1. **File -** In order to save the network scenario before or after running the simulation into the current workspace,
    - Click on File → Save to save the simulation inside the current workspace. Users can specify their own Experiment Name and Description (Optional).
    - Click on File → Save As to save an already saved simulation in a different name after performing required modifications to it.
    - Click on Close, to close the design window or GUI. It will take you to the home screen of NetSim.
2. **Help** - Help option allows the users to access all the help features.
    - **Video Tutorials** – Assists the users by directing them to our dedicated YouTube Channel **"TETCOS"**, where we have lots of video presentations ranging from short to long, covering different versions of NetSim up to the latest release.
    - **Answers/FAQ** – Assists the user by directing them to our **"NetSim Support Portal"**, where one can find a well-structured **"Knowledge Base"**, consisting of answers or solutions to all the commonest queries which a new user can go through.
    - **Raise a Support Ticket** – Assists the user by directing them to our **"NetSim Support Portal"**, where one can **"Submit a ticket"** or in other words raise his/her query, which reaches our dedicated Helpdesk and due support will be provided to the user.
    - **User Manual** – Assists the user with the usability of the entire tool and its features. It highly facilitates a new user with lots of key information about NetSim.
    - **Source Code Help** – Assists the user with a structured documentation for **"NetSim Source Code Help"**, which helps the users who are doing their R&D using NetSim with a structured code documentation consisting of more than 5000 pages with very much ease of navigation from one part of the document to another.
    - **Open-Source Code** – Assists the user to open the entire source codes of NetSim protocol libraries in Visual Studio, where one can start initiating the debugging process or performing modifications to existing code or adding new lines of code. Visual Studio Community Edition is a highly recommended IDE to our users who are using the R&D Version of NetSim.
    - **Experiments** – Assists the user with separate links provided for 30+ different experiments covering almost all the network technologies present in NetSim.
    - **Technology Libraries** – Assists the user by directing them to a folder comprising of individual technology library files comprising all the components present in NetSim.

Below the menu options, the entire region constitutes the Ribbon/Toolbar using which the following actions can be performed:

- Click and drop network devices and right click to edit properties.
- Click on Wired/Wireless links to connect the devices to one another. It automatically detects whether to use a Wired/Wireless link based on the devices we are trying to connect.
- Click on any Application from Set Traffic tab to configure different types of applications and generate traffic.
- Click on Plots, Packet Trace, and Event Trace from Configure reports tab to generate metrics to further analyze the network performance.
- Click on Run to perform the simulation and specify the simulation time in seconds.
- The Show/hide tab is mainly used to display various parameters like Device Name, IP, etc., to provide a better understanding especially during the designing of the scenario.

Results Window: Upon completion of simulation, Network statistics or network performance metrics reported in the form of graphs and tables. The report includes metrics like throughput, simulation time, packets generated, packets dropped, collision counts etc. see Figure 1-3 and Figure 1-4.

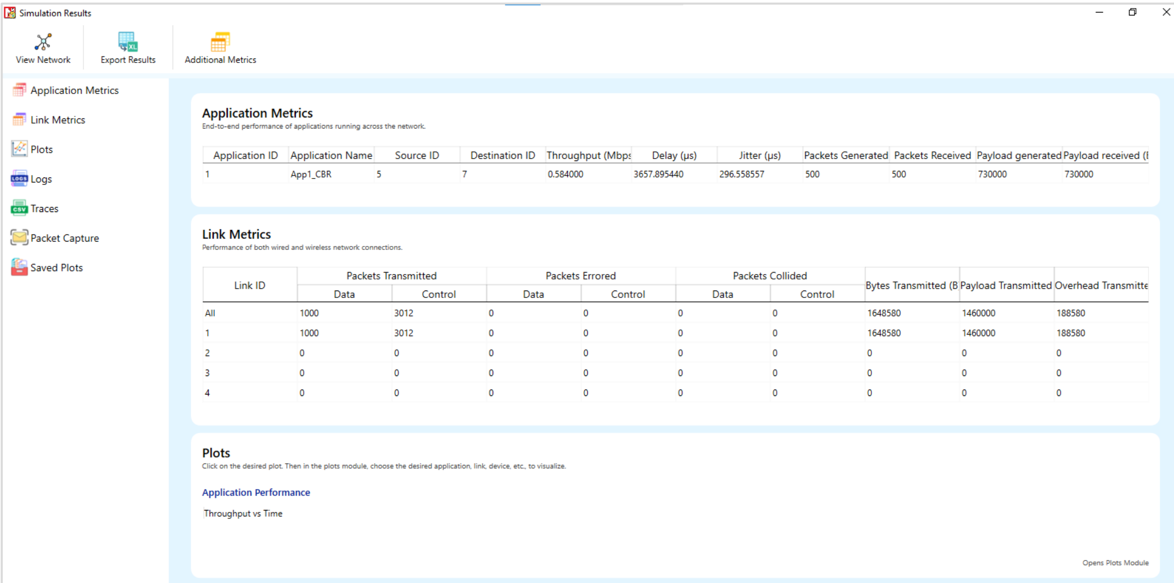


Figure 1-3: Results Window

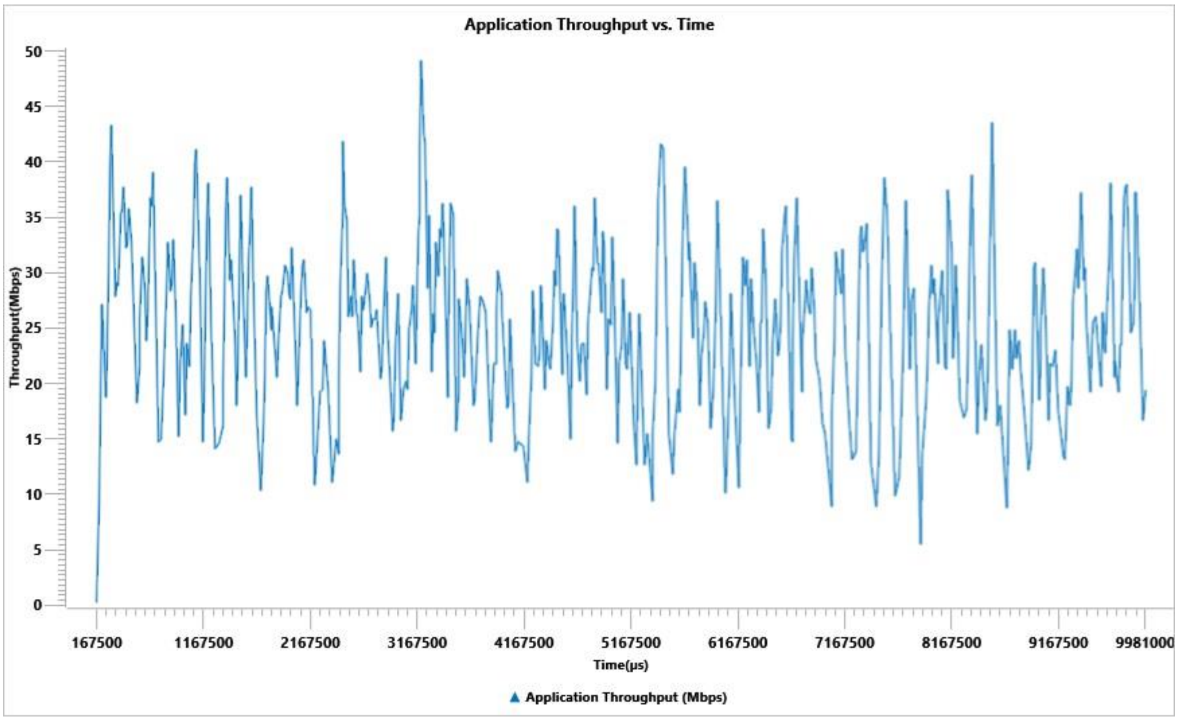


Figure 1-4: Application Throughput Plot

Description:

1. In the Simulation Results, clicking on a particular metrics will highlight the respective metrics window in the Centre.
2. Clicking on links in a particular metric link like Plots, Logs, Traces will open the respective files in a separate window.
3. Click on Packet trace / Event trace to open the csv file which will provide in depth analysis on each Packets / Events.
4. Users can click on Additional metrics on the top ribbon to view the additional metrics table.

1.1.2 How does a user create and save an experiment in workspace?

To create an experiment, select New simulation-> <Any Network> in the NetSim home screen Figure 1-5.

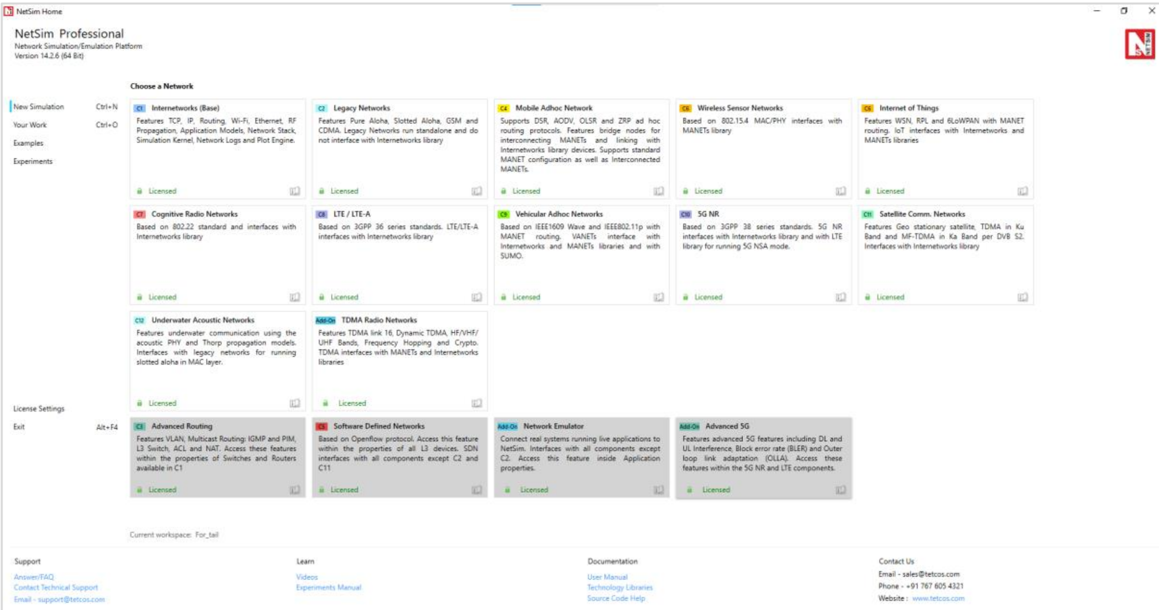


Figure 1-5: NetSim Home Screen

Create a network and save the experiment by clicking on File->Save button on the top left.

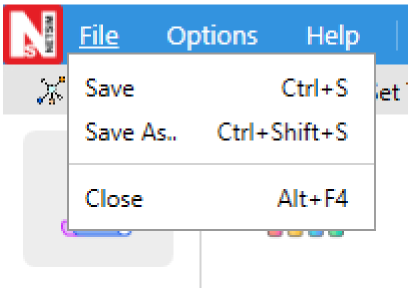


Figure 1-6: Save the network using file option

A save popup window appears which contains experiment name, workspace path and description see Figure 1-7.

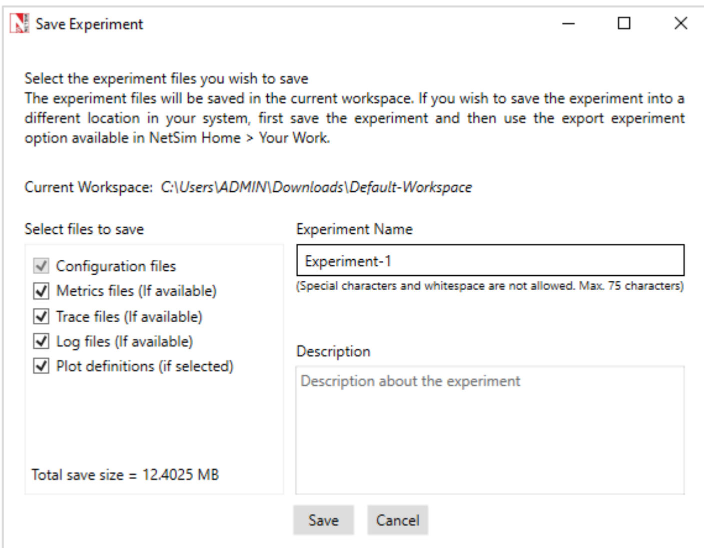


Figure 1-7: NetSim Save Window

Specify the Experiment name and description (optional) and then click on save. The workspace path is non-editable. Hence all the experiments will be saved in the default workspace path. After specifying the experiment name click on save.
In our example we saved with the name MANET and this experiment can be found in the default workspace path see below Figure 1-8.

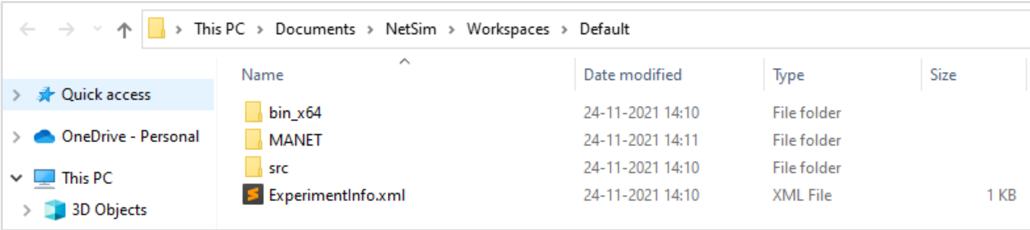


Figure 1-8: NetSim Default Workspace Path

Users can also see the saved experiments in "Your work" menu shown below Figure 1-9.

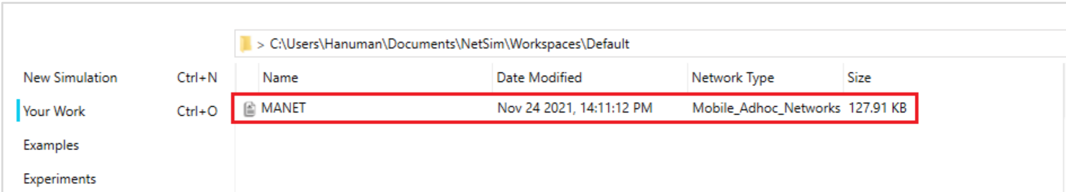


Figure 1-9: Your Work Menu

"Save As" option is also available to save the current experiment with a different name.

1.1.3 Typical sequence of steps to perform the experiments provided in this manual

The typical steps involved in doing experiments in NetSim are:

- **Network Set up:** Drag and drop devices and connect them using wired or wireless links.
- **Configure Properties:** Configure device, protocol, or link properties by right clicking on the device or link and modifying parameters in the properties window.
- **Model Traffic:** Click on the Application icon present in the Set Traffic tab and set traffic flows.
- **Enable Trace/Plots (optional):** To enable Packet Trace, Event Trace and Plots click on Configure Reports tab and select the respective icons to enable. Packet trace logs packet flow, event trace logs each event (NetSim is a discrete event simulator) and the Plots button enables charting of various throughputs over time.
- **Save/Save As/Open/Edit:** Click on File → Save / File → Save As to save the experiments in the current workspace. Saved experiments can then opened from NetSim home screen to run the simulation or to modify the parameters and again run the simulation.

NOTE: Example Configuration files for all experiments would available where NetSim has been installed. This directory is (<NetSim Install Directory>\Docs\Sample Configuration\NetSim Experiment Manual)

1.2 Understand the working of basic networking commands - Ping, Route Add/Delete/Print, ACL (Level 1)

1.2.1 Theory

NetSim allows users to interact with the simulation at runtime via a socket or through a file. User Interactions make simulation more realistic by allowing command execution to view/modify certain device parameters during runtime.

1.2.1.1 Ping Command

- The ping command is one of the most often used networking utilities for troubleshooting network problems.
- You can use the ping command to test the availability of a networking device (usually a computer) on a network.
- When you ping a device, you send that device a short message, which it then sends back (the echo)
- If you receive a reply then the device is in the Network, if you do not, then the device is faulty, disconnected, switched off, or incorrectly configured.

1.2.1.2 Route Commands

You can use the route commands to view, add and delete routes in IP routing tables.

- **route print:** In order to view the entire contents of the IP routing table.
- **route delete:** In order to delete all routes in the IP routing table.
- **route add:** In order to add a static TCP/IP route to the IP routing table.

1.2.1.3 ACL Configuration

Routers provide basic traffic filtering capabilities, such as blocking the Internet traffic with access control lists (ACLs). An ACL is a sequential list of **Permit** or **Deny** statements that apply to addresses or upper-layer protocols. These lists tell the router what types of packets to: **PERMIT** or **DENY**. When using an access-list to filter traffic, a PERMIT statement is used to **"allow"** traffic, while a DENY statement is used to **"block"** traffic.

1.2.2 Network setup

Open NetSim and click on **Experiments> Advanced Routing> Basic networking commands Ping Route Add/Delete/Print and ACL** then click on the tile in the middle panel to load the example as shown in below Figure 1-10.

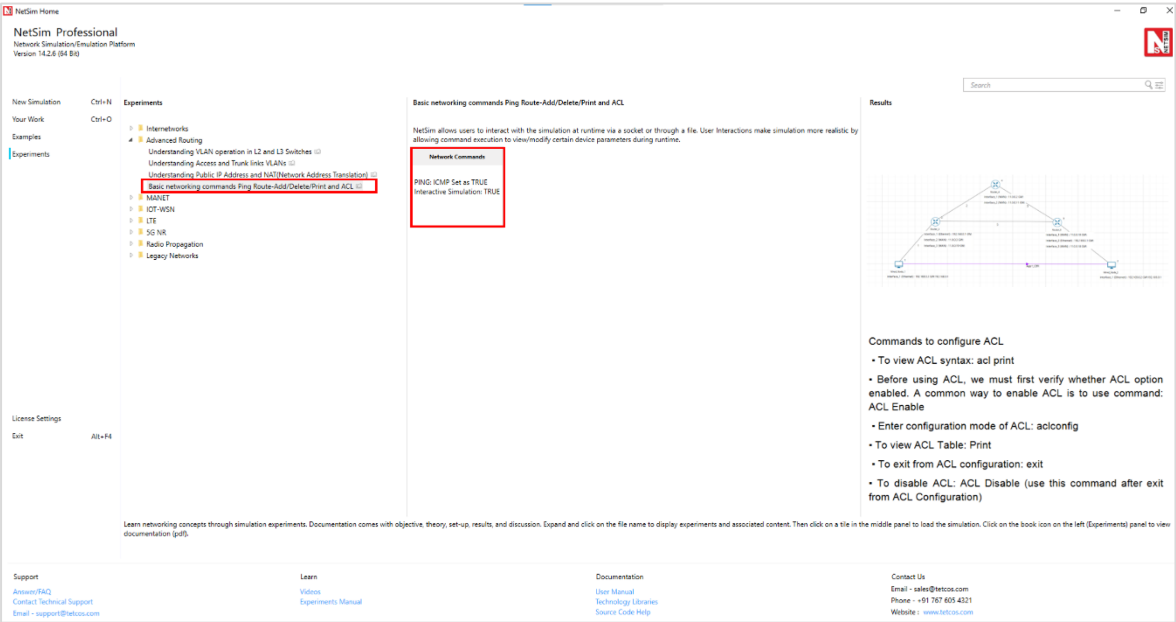


Figure 1-10: List of scenarios for the example of Basic networking commands Ping Route Add/Delete/Print and ACL

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 1-11.

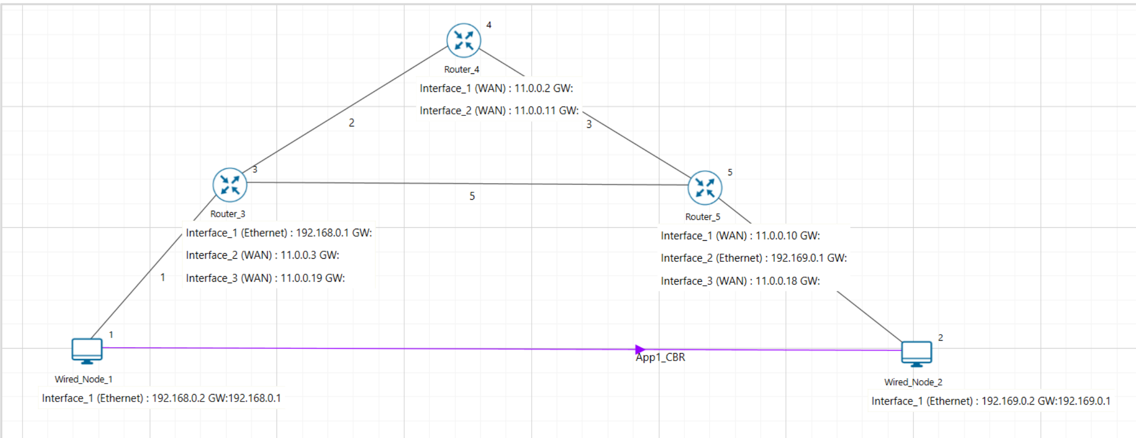


Figure 1-11: Network set up for studying the Basic networking commands: Ping, Route Add/Delete/Print, and ACL

1.2.3 Procedure

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 3 Routers in the **“Internetworks”** Network Library.

Step 2: Click on Wired Node 1 and then open right-side property panel .In the Network Layer properties of Wired Node 1, **“ICMP Status”** is set as TRUE.

Similarly, ICMP Status is set as TRUE for all the devices as shown Figure 1-12.

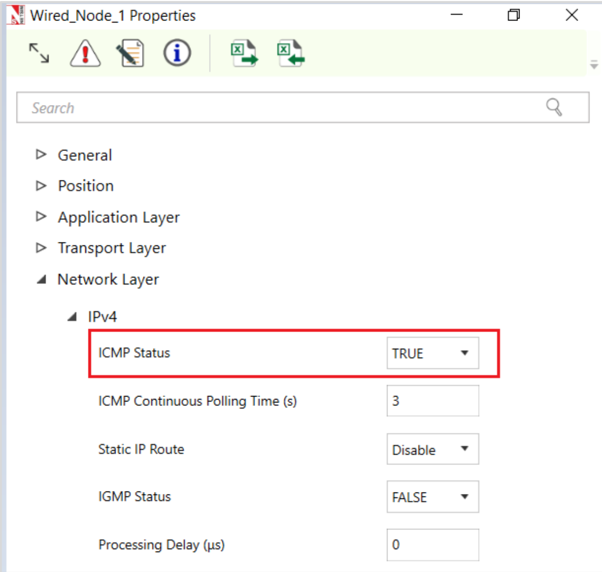


Figure 1-12: Network Layer properties of Wired Node 1

Step 3: Click on Wired Node 1 and then open right-side property panel. In the General properties of Wired Node 1, **Wireshark Capture** is set as Online .

Step 4: Configure an application between any two nodes by selecting a CBR application from Wired Node 1 i.e., Source to Wired Node 2 i.e., Destination from Set Traffic tab in the ribbon. Click on the application, expand the right-side property panel and set Transport Protocol to UDP and Set Packet Size: 1460 Bytes, Inter Arrival Time remaining 233.6µs

Additionally, the **"Start Time(s)"** parameter is set to 30, while configuring the application. This time is usually set to be greater than the time taken for OSPF Convergence (i.e., Exchange of OSPF information between all the routers), and it increases as the size of the network increases.

Step 5: Packet Trace is enabled in NetSim GUI. At the end of the simulation, a very large .csv file containing the packet information is available for the users to perform packet level analysis.

Step 6: Before simulating the scenario, right click on Router 3 or any other Router and select **"NetSim Console"** option as shown

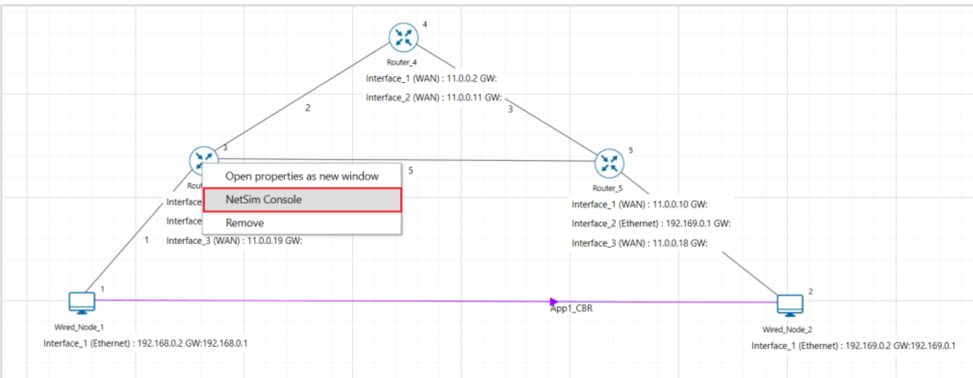


Figure 1-13: Select NetSim console

- Now client (NetSimCLI.exe) will start running and it will try to establish a connection with NetSimCore.exe as shown in Figure 1-14.

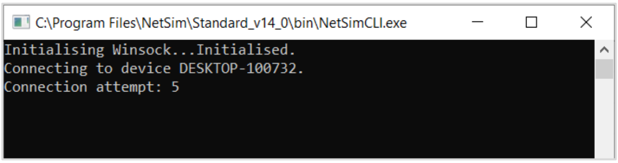


Figure 1-14: Connection established.

Step 7: Click on options and select Run Time Interactions as shown below

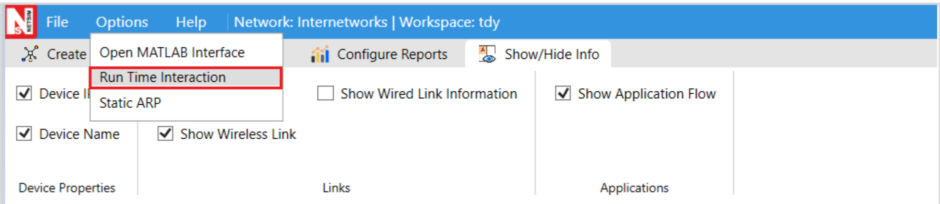


Figure 1-15: Runtime Interaction window

- In the Run time Interaction tab, Interactive simulation option is set to True and click on OK.

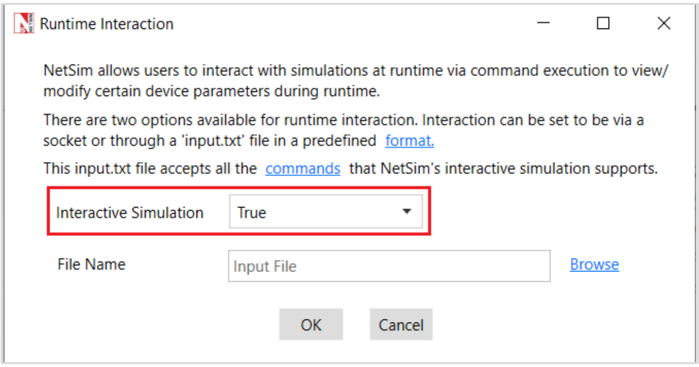


Figure 1-16: Runtime Interaction Simulation window

- Click on Run simulation window and simulate it to 300 sec.

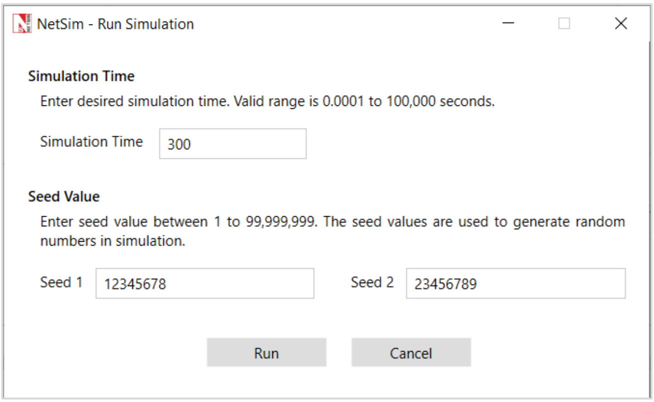


Figure 1-17: Run Simulation Window

NOTE: It is recommended to specify a longer simulation time to ensure that there is sufficient time for the user to execute the various commands and see the effect of that before the Simulation ends.

- (NetSimCore.exe) will start running and will display a message **"waiting for first client to connect"** as shown below.

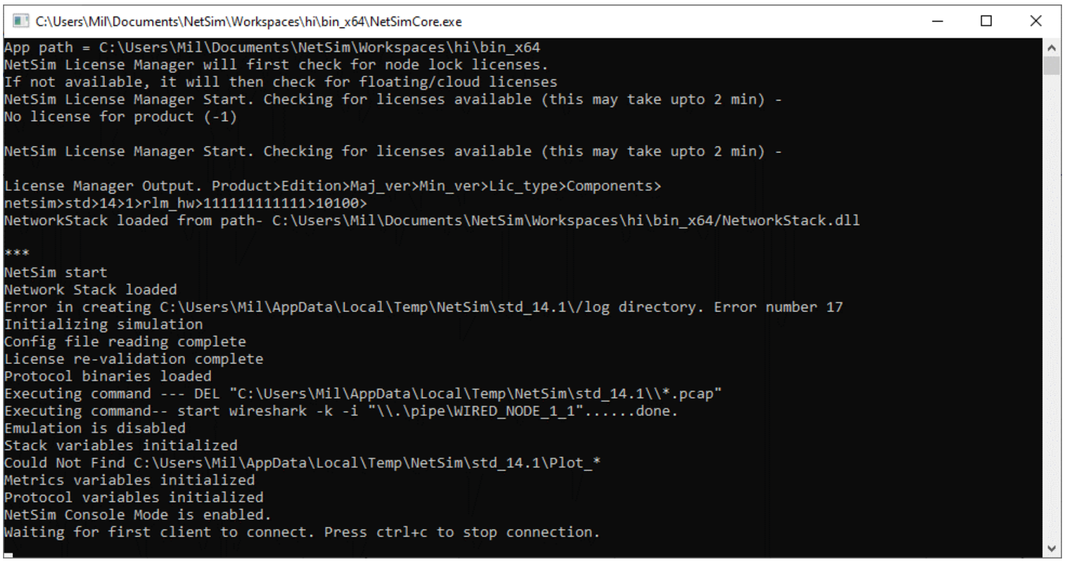


Figure 1-18: Waiting for first client to connect.

- After this the command line interface can be used to execute all the supported commands.

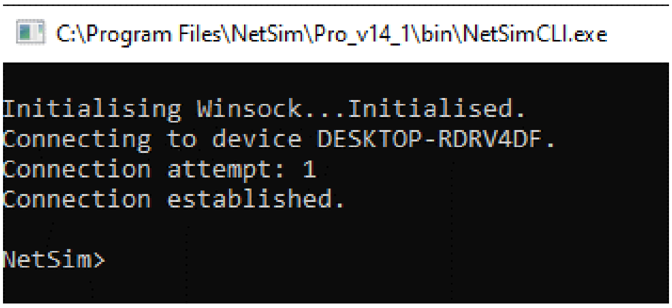


Figure 1-19: Connection establishment.

1.2.4 Network Commands

1.2.4.1 Ping Command

- You can use the ***ping*** command with an IP address or Device name.
- ICMP Status should be set as True in all nodes for ping to work.

Ping <IP address> e.g. ping 192.169.0.2

Ping <Node Name> e.g. ping Wired_Node_2

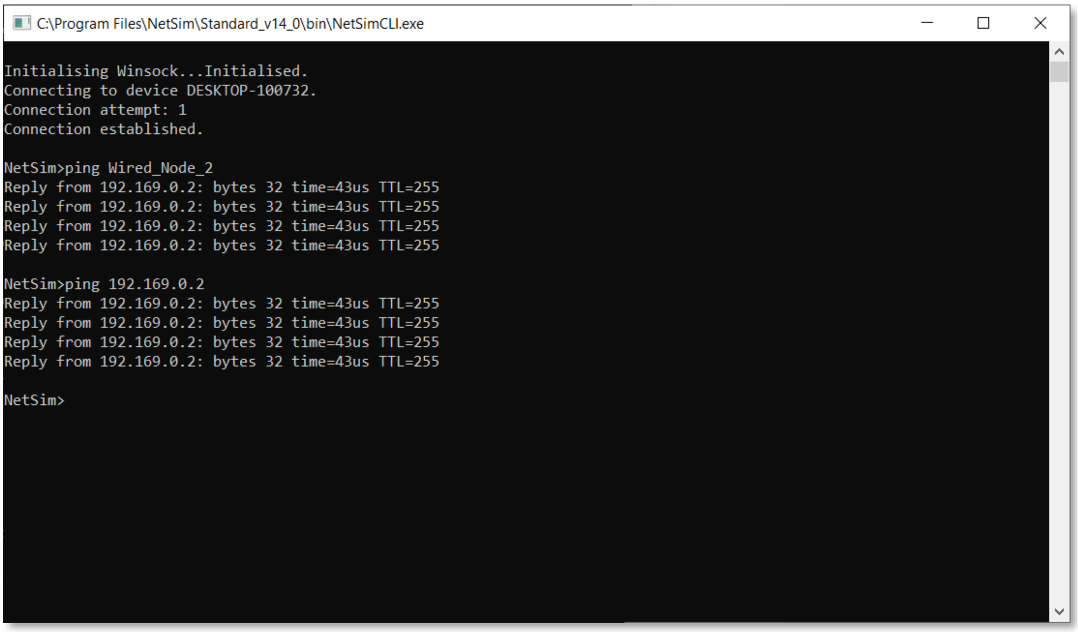


Figure 1-20: Pinging Wired Node 2

1.2.4.2 Route Commands

- To view the entire contents of the IP routing table, use following command **route print**

route print

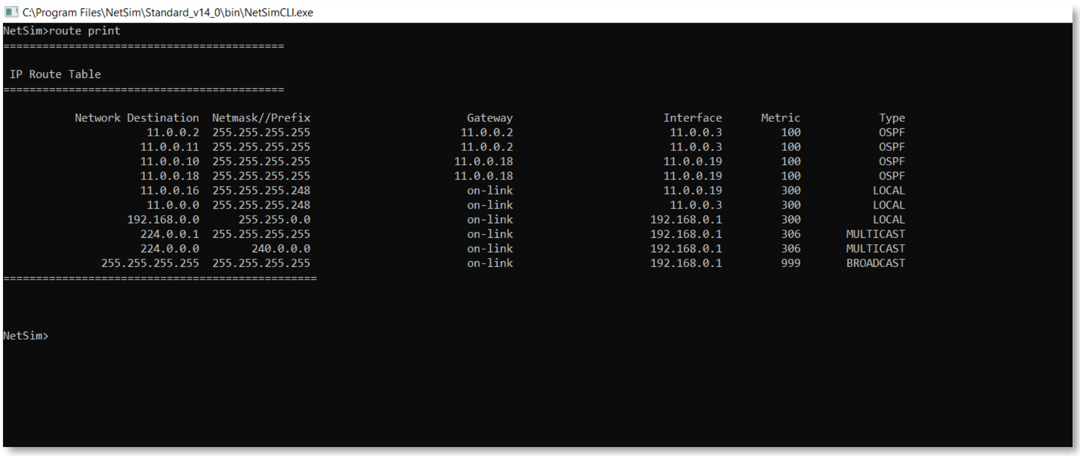


Figure 1-21: IP routing table

- You'll see the routing table entries with network destinations and the gateways to which packets are forwarded when they are headed to that destination. Unless you've already added static routes to the table, everything you see here is dynamically generated.
- To delete a route in the IP routing table you'll type a command using the following syntax

route delete *destination network*

- So, to delete the route with destination network 11.0.0.18, all we'd have to do is type this command

```
route delete 11.0.0.18
```

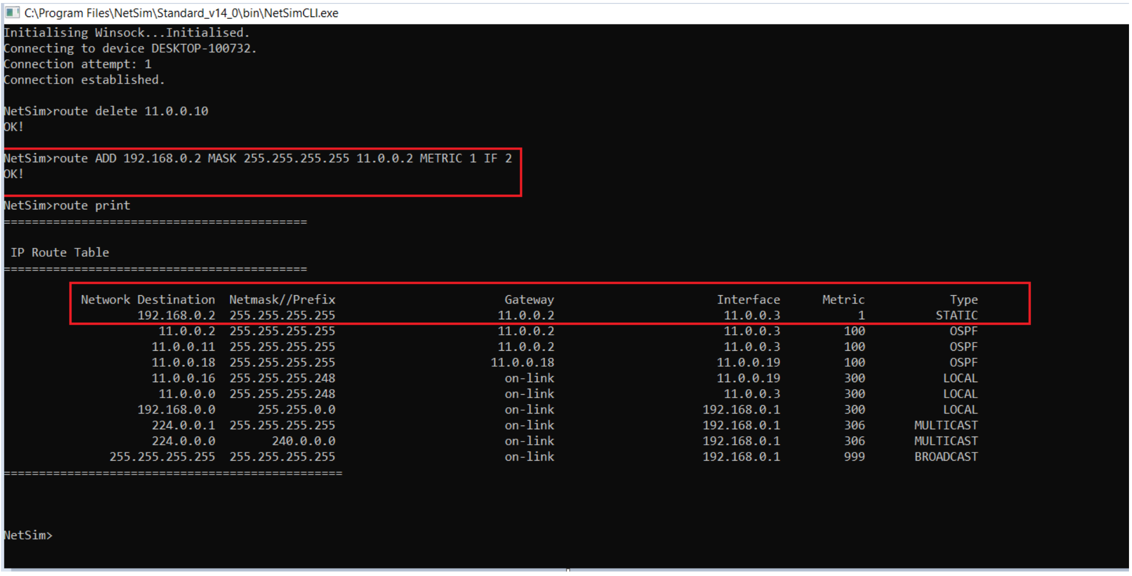
- To check whether route has been deleted or not check again using **route print** command.
- To add a static route to the table, you'll type a command using the following syntax.

route ADD *destination_network* MASK *subnet_mask gateway_ip metric_cost interface*

- So, for example, if you wanted to add a route specifying that all traffic bound for the 11.0.0.18 subnet went to a gateway at 11.0.0.19

```
route ADD 192.169.0.2 MASK 255.255.255.255 11.0.0.2 METRIC 1 IF 2
```

- If you were to use the route print command to look at the table now, you'd see your new static route.



```
C:\Program Files\NetSim\Standard_v14_0\bin\NetSimCLI.exe
Initialising Winsock...Initialised.
Connecting to device DESKTOP-100732.
Connection attempt: 1
Connection established.

NetSim>route delete 11.0.0.10
OK!

NetSim>route ADD 192.168.0.2 MASK 255.255.255.255 11.0.0.2 METRIC 1 IF 2
OK!

NetSim>route print
==========================================

 IP Route Table
==========================================

      Network Destination  Netmask//Prefix          Gateway        Interface   Metric        Type
              192.168.0.2  255.255.255.255         11.0.0.2         11.0.0.3        1      STATIC
                 11.0.0.2  255.255.255.255         11.0.0.2         11.0.0.3      100        OSPF
                11.0.0.11  255.255.255.255         11.0.0.2         11.0.0.3      100        OSPF
                11.0.0.18  255.255.255.255        11.0.0.18        11.0.0.19      100        OSPF
                11.0.0.16  255.255.255.248          on-link        11.0.0.19      300       LOCAL
                 11.0.0.0  255.255.255.248          on-link         11.0.0.3      300       LOCAL
              192.168.0.0      255.255.0.0          on-link      192.168.0.1      300       LOCAL
                224.0.0.1  255.255.255.255          on-link      192.168.0.1      306   MULTICAST
                224.0.0.0        240.0.0.0          on-link      192.168.0.1      306   MULTICAST
          255.255.255.255  255.255.255.255          on-link      192.168.0.1      999   BROADCAST
=============================================


NetSim>
```

Figure 1-22: Route delete/ Route add

NOTE: Entry added in IP table by routing protocol continuously gets updated. If a user tries to remove a route via route delete command, there is always a chance that routing protocol will re-enter this entry again. Users can use ACL / Static route to override the routing protocol entry if required.

1.2.4.3 ACL Configuration

Commands to configure ACL

- To view ACL syntax: ***acl print***
- Before using ACL, we must first verify whether ACL option enabled. A common way to enable ACL is to use command: ***ACL Enable***
- Enter configuration mode of ACL: ***aclconfig***
- To view ACL Table: ***Print***

- To exit from ACL configuration: ***exit***
- To disable ACL: ***ACL Disable*** (use this command after **exit** from ACL Configuration)

To view ACL usage syntax use: **acl print**

[PERMIT, DENY] [INBOUND, OUTBOUND, BOTH] PROTO SRC DEST SPORT DPORT IFID

1.2.4.4 Step to Configure ACL

- To create a new rule in the ACL, use command as shown below to block UDP packet in Interface 2 and Interface 3 of Router 3.
- Application properties → Transport Protocol → **UDP** as shown Figure 1-23

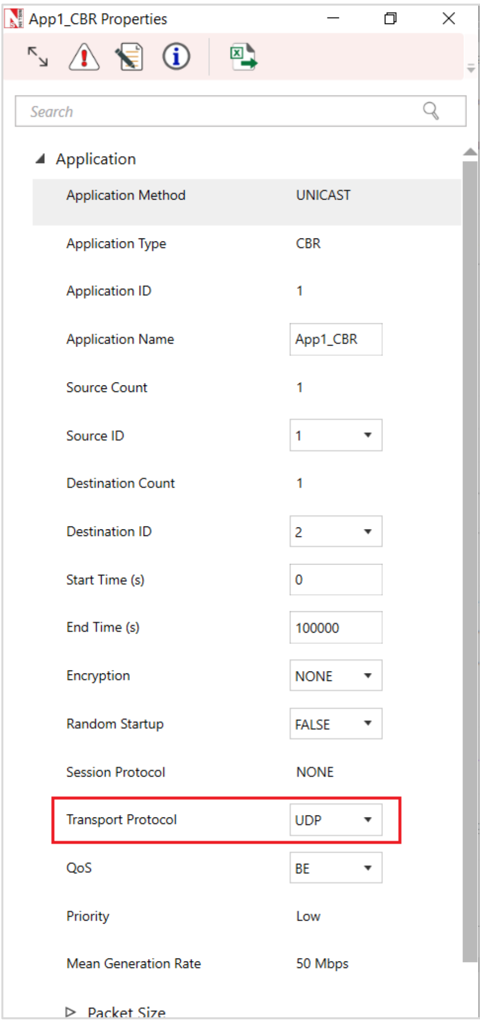


Figure 1-23: Application properties window

- Use the command as follows Figure 1-24.

*NetSim>**acl enable***

ACL is enable

*NetSim>**aclconfig***

*ROUTER_3/ACLCONFIG>**acl print***

Usage: [PERMIT, DENY] [INBOUND, OUTBOUND, BOTH] PROTO SRC DEST SPORT DPORT IFID

*ROUTER_3/ACLCONFIG>**DENY BOTH UDP ANY ANY 0 0 2***

OK!

*ROUTER_3/ACLCONFIG>**DENY BOTH UDP ANY ANY 0 0 3***

OK!

*ROUTER_3/ACLCONFIG>**print***

DENY BOTH UDP ANY/0 ANY/0 0 0 2

DENY BOTH UDP ANY/0 ANY/0 0 0 3

*ROUTER_3/ACLCONFIG>**exit***

*NetSim>**acl disable***

ACL is disable

NetSim>

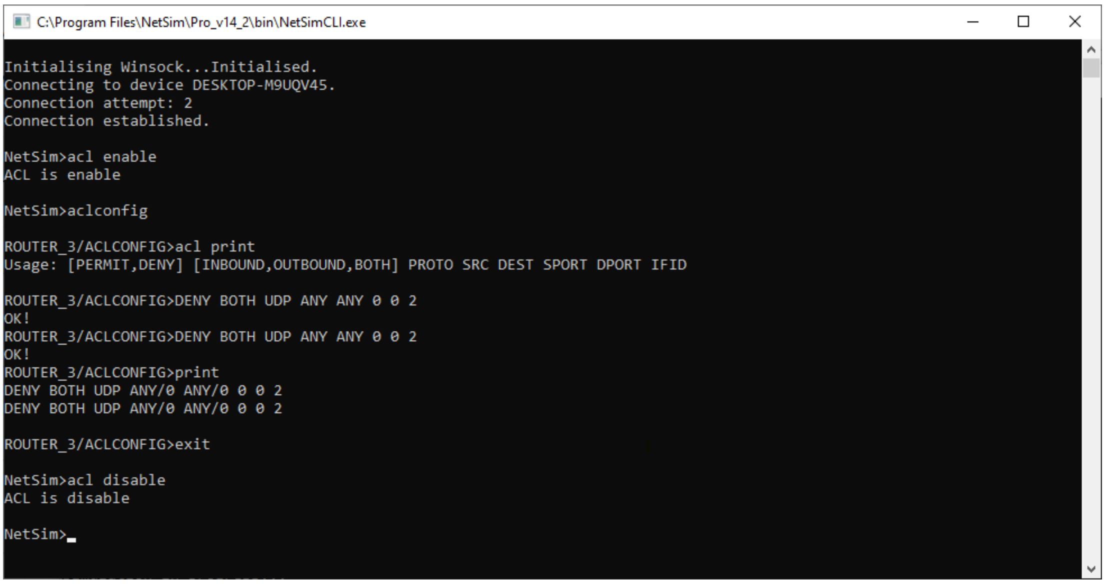


Figure 1-24: ACL Configuration command

1.2.4.5 Ping Command Results

Go to the Results Dashboard and click on **"Open Packet Trace"** option present in the Left-Hand-Side of the window and do the following:

Filter Control Packet Type/App Name to **ICMP EchoRequest** and **ICMP EchoReply** as shown Figure 1-25.

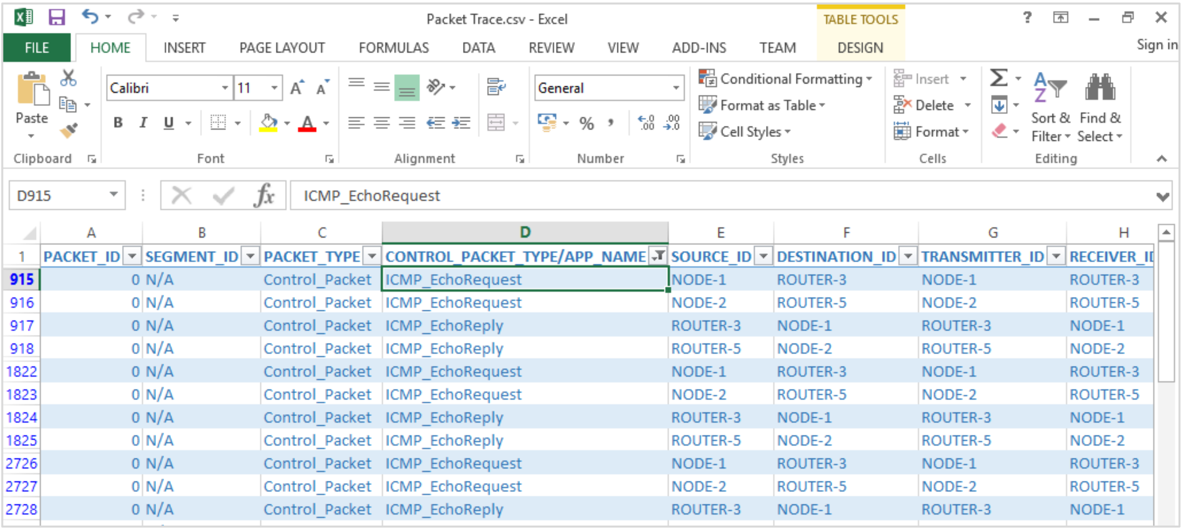


Figure 1-25: Packet trace - ICMP control packets

In wireshark, apply filter as ICMP. we can see the ping request and reply packets in wireshark as shown Figure 1-26.

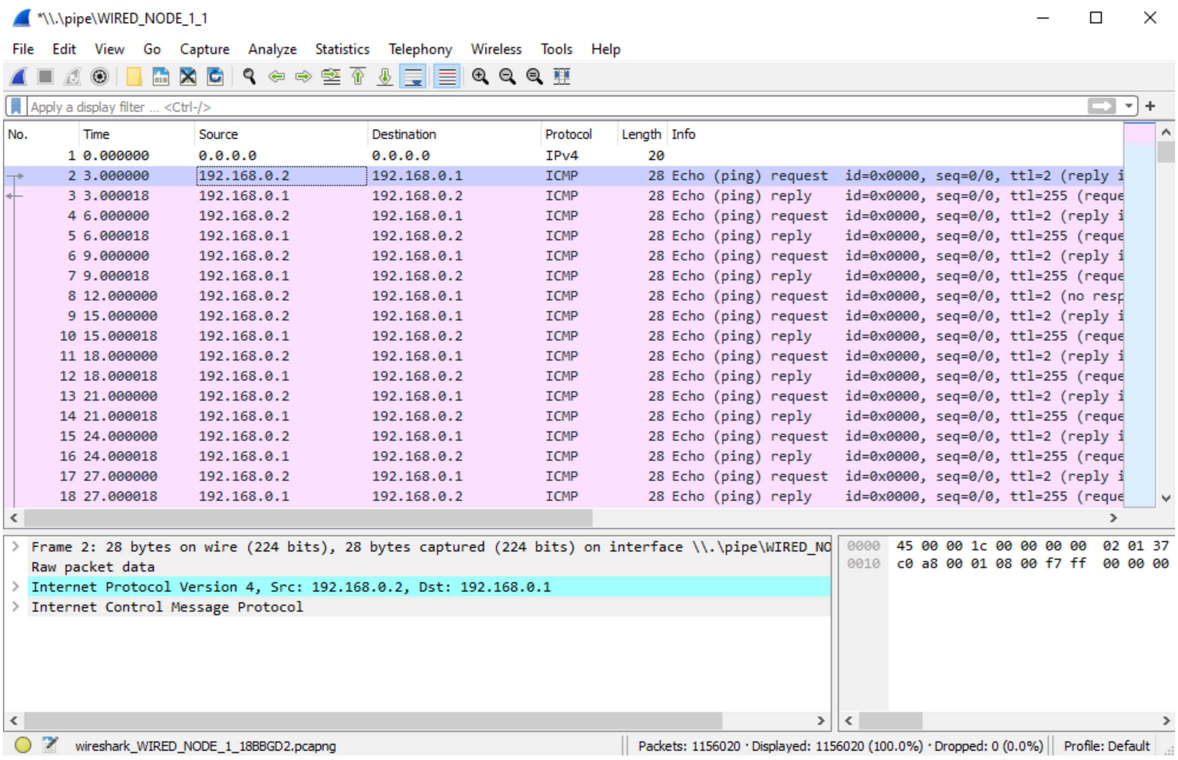


Figure 1-26: ICMP control packets in wireshark

1.2.4.6 ACL Results

The impact of ACL rule applied over the simulation traffic can be observed in the IP Metrics Table in the simulation results window. In Router 3, the number of packets blocked by firewall has been shown below Figure 1-27.

IP_Metrics

Device Id	Packet sent	Packet forwarded	Packet received	Packet discarded	TTL expired	Firewall blocked
1	55	0	52	0	0	0
2	97156	97105	53	0	0	0
3	97278	145371	54	0	0	48147
4	145548	0	0	0	0	0
5	0	0	96987	0	0	0

Figure 1-27: IP Metrics Table from result window

NOTE: Number of packets blocked may vary depending on the time at which ACL is configured.

1.2.5 Exercises

a. Construct the network scenario shown in the figure below comprising of 5 routers (3 routers in upper path and 2 routers in the lower path) and 2 end nodes. Enable the ICMP protocol in all devices and use the route commands - "route delete" and "route add" - to alter the path for data transfer. The data flow path should be Node1 > R3 > R5 > R6 > R7 > Node2. Enable packet trace and confirm that data indeed flows along this path post simulation.

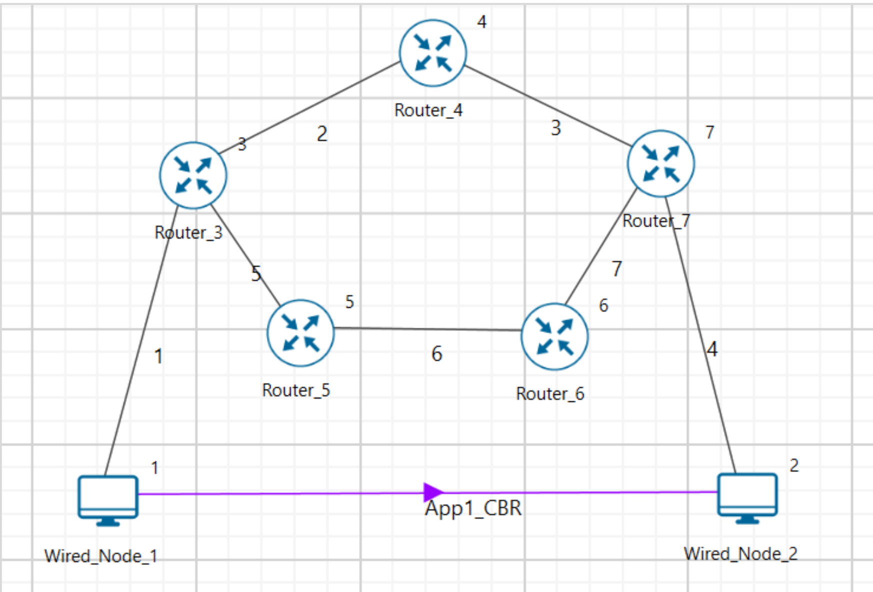


Figure 1-28: Network scenario for reference.

b. Configure the scenario as shown and apply the ACL command as follows.

- Apply DENY action for R1 interface 1 and 2.
- Apply Permit action for R1 interface 3.

Analyze the firewall blocks in the IP metrics table and explain the throughput results.

Note that ACL commands will only take effect once the commands are entered in the console window.

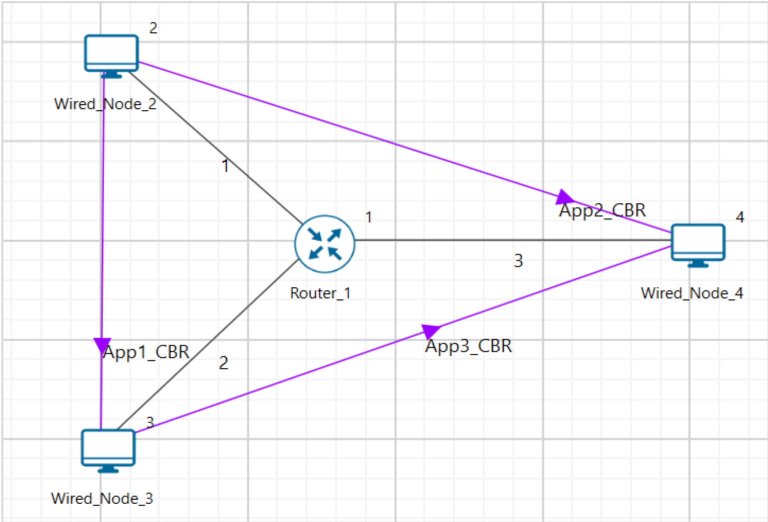


Figure 1-29: Network scenario for ACL exercise.

1.3 Understand the events involved in NetSim DES (Discrete Event Simulator) in simulating flow of one packet from a Wired node to a Wireless node (Level 2)

1.3.1 Theory

NetSim's Network Stack forms the core of NetSim, and its architectural aspects are diagrammatically explained below. Network Stack accepts inputs from the end-user in the form of Configuration file and the data flows as packets from one layer to another layer in the Network Stack. All packets, when transferred between devices move up and down the stack, and all events in NetSim fall under one of these ten categories of events, namely, **Physical In, Data Link In, Network In, Transport In, Application In, Application Out, Transport Out, Network Out, Data link Out** and **Physical Out**. The IN events occur when the packets are entering a device while the OUT events occur while the packet is leaving a device.

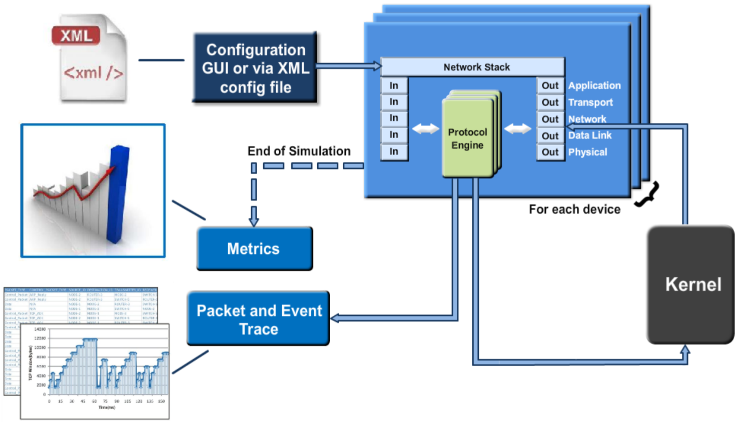


Figure 1-30: Flow of one packet from a Wired node to a Wireless node

Every device in NetSim has an instance of the Network Stack shown above. Switches & Access points have a 2-layer stack, while routers have a 3-layer stack. End-nodes have a 5-layer stack.

The protocol engines are called based on the layer at which the protocols operate. For example, TCP is called during execution of Transport In or Transport Out events, while 802.11b WLAN is called during execution of MAC IN, MAC OUT, PHY IN and PHY OUT events.

When these protocols are in operation, they in turn generate events for NetSim's discrete event engine to process. These are known as sub events. All sub events fall into one of the above 10 types of events.

Each event gets added in the simulation kernel by the protocol operating at the particular layer of the Network stack. The required sub events are passed into the simulation kernel. These sub events are then fetched by the Network stack in order to execute the functionality of each protocol. At the end of simulation, Network stack writes trace files and the metrics files that assist the user in analyzing the performance metrics and statistical analysis.

Event Trace

The event trace records every single event along with associated information such as time stamp, event Id, event type etc. in a text file or .csv file which can be stored at a user defined location.

1.3.2 Network Setup

Open NetSim and click on **Experiments> Internetworks> Network Performance> Advanced Simulation events in NetSim for transmitting one packet** then click on the tile in the middle panel to load the example as shown in below Figure 1-31.

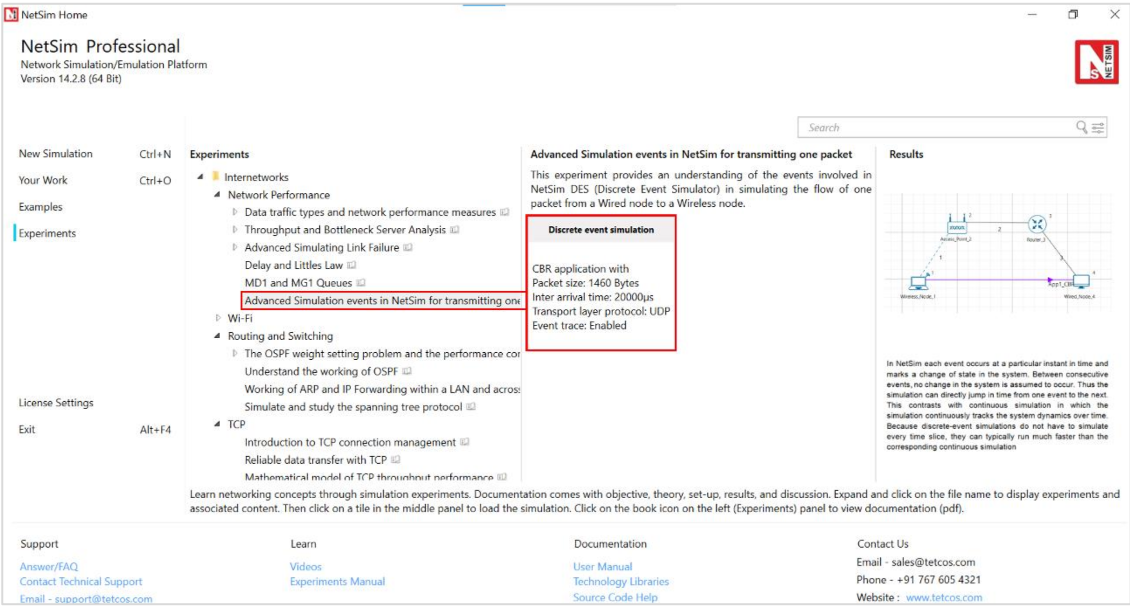


Figure 1-31: List of scenarios for the example of Advanced Simulation events in NetSim for transmitting one packet.

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 1-32.

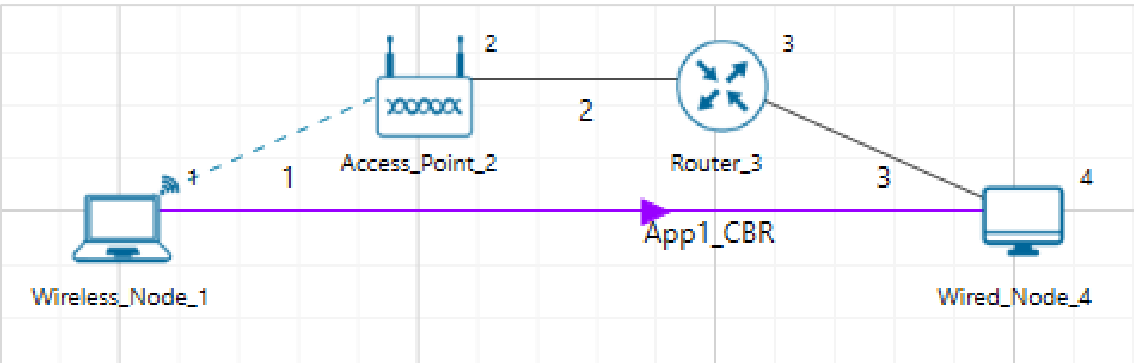


Figure 1-32: Network set up for studying the advanced simulation events in NetSim for transmitting one packet.

1.3.3 Procedure

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

Step 1: A network scenario is designed in NetSim GUI comprising of 1 wired node, 1 wireless node, 1 router, and 1 Access point in the **"Internetworks"** Network library.

Step 2: The device positions are configured according to Table 1-1 .To set the device positions, click on the device, expand the property panel on the right and change the X and Y co-ordinates in position layer.

Device positions				
	Access Point 2	**Wired Node 4**	**Wireless Node 1**	**Router 3**
X / Lon	150	250	100	200
Y / Lat	50	100	100	50

Table 1-1: Devices positions

Step 3: Click on link and expand the link property panel on right and set the Channel characteristics to No pathloss.

Step 4: A CBR Application is generated from wired node 4 i.e., source to wireless node 1 i.e., by clicking on set traffic tab from the ribbon on top. The transport protocol is set to UDP by clicking on application and expanding property panel on right.

Step 5: Event Trace is enabled by clicking on configure reports tab from the ribbon on top.

Step 6: Run the simulation for 10 secs. At the end of the simulation, a very large .csv file containing all the UDP IN and OUT EVENTS is available for the users.

NOTE: Event trace is available only in NetSim Standard and Pro versions.

1.3.4 Output

Post simulation, open the event trace from the simulation results window. An Event trace file opens in excel as shown below:

Event_Id	Event_Type	Event_Time(US)	Device_Type	Device_Id	Interface_Id	Application_Id	Packet_Id	Segment_Id	Protocol_Name	Subevent_Type	Packet_Size(Bytes)	Prev_Event_Id
1	TIMER_EVENT	0	NODE	1	0	0	0	0	IPV4	IP_INIT_TABLE	0	0
2	TIMER_EVENT	0	ROUTER	3	0	0	0	0	IPV4	IP_INIT_TABLE	0	0
3	TIMER_EVENT	0	NODE	4	0	0	0	0	IPV4	IP_INIT_TABLE	0	0
4	TIMER_EVENT	0	ACCESSPOINT	2	2	0	0	0	ETHERNET	ETH_IF_UP	0	0
5	TIMER_EVENT	0	ROUTER	3	1	0	0	0	ETHERNET	ETH_IF_UP	0	0
6	TIMER_EVENT	0	ROUTER	3	2	0	0	0	ETHERNET	ETH_IF_UP	0	0
7	TIMER_EVENT	0	NODE	4	1	0	0	0	ETHERNET	ETH_IF_UP	0	0
8	TIMER_EVENT	0	NODE	4	0	1	1	0	APPLICATION	APP_START	1460	0
9	APPLICATION_	0	NODE	4	0	1	1	0	APPLICATION	0	1460	8
11	TRANSPORT_O	0	NODE	4	0	1	2	0	UDP	0	1460	9
12	NETWORK_OU	0	NODE	4	0	1	1	0	IPV4	0	1468	11
13	TIMER_EVENT	0	NODE	4	1	1	1	0	IPV4	IP_PROCESSING_DELAY	1488	12
14	MAC_OUT	0	NODE	4	1	1	1	0	ETHERNET	0	1488	13
15	PHYSICAL_OUT	0	NODE	4	1	1	1	0	ETHERNET	0	1514	14
16	PHYSICAL_IN	127.08	ROUTER	3	2	1	1	0	ETHERNET	0	1514	15
17	MAC_IN	127.08	ROUTER	3	2	1	1	0	ETHERNET	0	1514	16
18	NETWORK_IN	127.08	ROUTER	3	2	1	1	0	IPV4	0	1488	17
19	NETWORK_OU	127.08	ROUTER	3	2	1	1	0	IPV4	0	1468	18
20	TIMER_EVENT	127.08	ROUTER	3	1	1	1	0	IPV4	IP_PROCESSING_DELAY	1488	19
21	MAC_OUT	127.08	ROUTER	3	1	1	1	0	ETHERNET	0	1488	20
22	PHYSICAL_OUT	127.08	ROUTER	3	1	1	1	0	ETHERNET	0	1514	21
23	PHYSICAL_IN	253.2	ACCESSPOINT	2	2	1	1	0	ETHERNET	0	1514	22
24	MAC_IN	253.2	ACCESSPOINT	2	2	1	1	0	ETHERNET	0	1514	23
25	MAC_OUT	253.2	ACCESSPOINT	2	1	1	1	0	WLAN	0	1488	24
26	MAC_OUT	253.2	ACCESSPOINT	2	1	1	1	0	WLAN	CS	1488	25
27	MAC_OUT	303.2	ACCESSPOINT	2	1	1	1	0	WLAN	IEEE802_11_EVENT_DIFS_EN	1488	26
28	MAC_OUT	323.2	ACCESSPOINT	2	1	1	1	0	WLAN	IEEE802_11_EVENT_BACKOF	1488	27
29	MAC_OUT	343.2	ACCESSPOINT	2	1	1	1	0	WLAN	IEEE802_11_EVENT_BACKOF	1488	28
30	MAC_OUT	363.2	ACCESSPOINT	2	1	1	1	0	WLAN	IEEE802_11_EVENT_BACKOF	1488	29

Figure 1-33: Event trace

We start from the **APPLICATION OUT** event of the first packet, which happens in the Wired Node and end with the **MAC IN** event of the **WLAN ACK** packet which reaches the Wired Node. Events in the event trace are logged with respect to the time of occurrence due to which, event id may not be in order.

1.3.4.1 Events Involved

Events are listed in the following format:

[EVENT_TYPE,	EVENT_TIME,	PROTOCOL,	EVENT_NO,	SUBEVENT_TYPE]
[APP_OUT,	20000,	APP,	6,	-]
[TRNS_OUT,	20000,	UDP,	7,	-]
[NW_OUT,	20000,	IPV4,	9,	-]
[MAC_OUT,	20000,	ETH,	10,	-]
[MAC_OUT,	20000,	ETH,	11,	CS]
[MAC_OUT,	20000.96,	ETH,	12,	IFG]
[PHY_OUT,	20000.96,	ETH,	13,	-]
[PHY_OUT,	20122.08,	ETH,	14,	PHY_SENSE]
[PHY_IN,	20127.08,	ETH,	15,	-]
[MAC_IN,	20127.08,	ETH,	16,	-]
[NW_IN,	20127.08,	IPV4,	17,	-]
[NW_OUT,	20127.08,	IPV4,	18,	-]
[MAC_OUT,	20127.08,	ETH,	19,	-]
[MAC_OUT,	20127.08,	ETH,	20,	CS]
[MAC_OUT,	20128.04,	ETH,	21,	IFG]
[PHY_OUT,	20128.04,	ETH,	22,	-]
[PHY_OUT,	20249.16,	ETH,	23,	PHY_SENSE]
[PHY_IN,	20254.16,	ETH,	24,	-]
[MAC_IN,	20254.16,	ETH,	25,	-]
[MAC_OUT,	20254.16,	WLAN,	26,	-]
[MAC_OUT,	20254.16,	WLAN,	27,	DIFS_END]
[MAC_OUT,	20304.16,	WLAN,	28,	BACKOFF]

[MAC_OUT,	20324.16,	WLAN,	29,	BACKOFF]
[MAC_OUT,	20344.16,	WLAN,	30,	BACKOFF]
[MAC_OUT,	20364.16,	WLAN,	31,	BACKOFF]
[PHY_OUT,	20364.16,	WLAN,	32,	-]
[TIMER_EVENT,	21668.16,	WLAN,	35,	UPDATE_DEVICE_STATUS]
[PHY_IN,	21668.4,	WLAN,	33,	-]
[MAC_IN,	21668.4,	WLAN,	36,	RECEIVE_MPDU]
[NW_IN,	21668.4,	IPV4,	37,	-]
[MAC_OUT,	21668.4,	WLAN,	38,	SEND_ACK]
[TRNS_IN,	21668.4,	UDP,	39,	-]
[APP_IN,	21668.4,	APP,	41,	-]
[PHY_OUT,	21678.4,	WLAN,	40,	-]
[TIMER_EVENT,	21982.4,	WLAN,	43,	UPDATE_DEVICE]
[PHY_IN,	21982.63,	WLAN,	42,	-]
[MAC_IN,	21982.63,	WLAN,	44,	RECEIVE_ACK]
[TIMER_EVENT,	21985,	WLAN,	34,	ACK_TIMEOUT]

Event Flow Diagram for one packet from Wired Node to Wireless Node

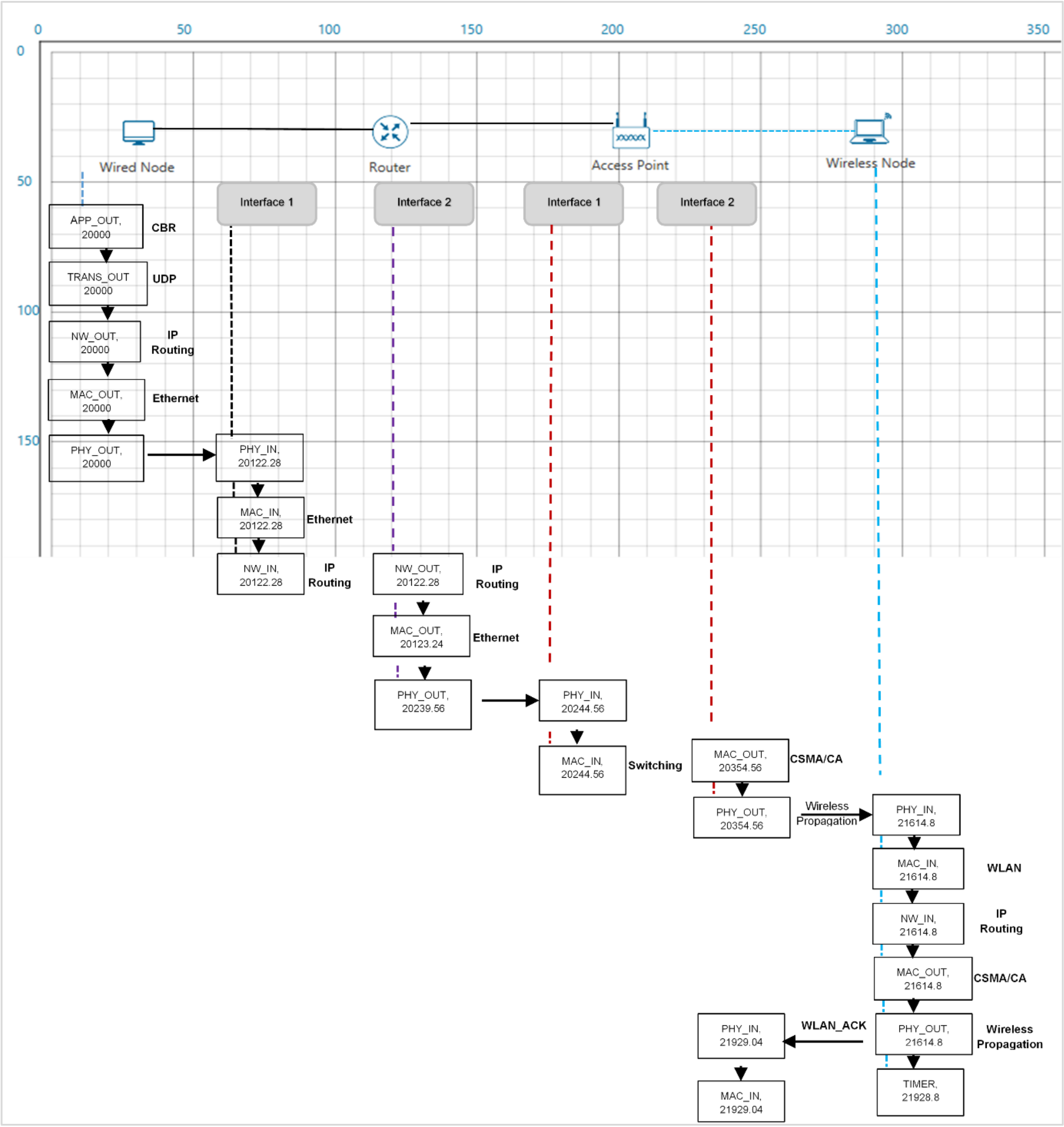


Figure 1-34: Event Flow Diagram for one packet from Wired Node to Wireless Node

For Example:

MAC OUT in the Access Point involves sub events like CS, IEEE802.11 EVENT DIFS END and IEEE802.11 EVENT BACKOFF. As you can see in the trace file shown below, CS happens at event time 20252.24. Adding DIFS time of 50µs to this will give IEEE802.11 EVENT DIFS END sub event at 20302.24. Further it is followed by three backoffs each of 20 µs, at event time 20322.24, 20342.24, 020362.24 respectively.

Event_Id	Event_Type	Event_Time(US)	Device_Type	Device_Id	Interface_Id	Applicatio	Packet_Id	Segment_Id	Protocol_Name	Subevent_Type	Packet_Size	Prev_Event_Id
53	TIMER_EVENT	20126.12	ROUTER	3	1	1	2	0	IPV4	IP_PROCESSING_DELAY	1488	52
54	MAC_OUT	20126.12	ROUTER	3	1	1	2	0	ETHERNET	0	1488	53
55	PHYSICAL_OUT	20126.12	ROUTER	3	1	1	2	0	ETHERNET	0	1514	54
56	PHYSICAL_IN	20252.24	ACCESSPOINT	2	2	1	2	0	ETHERNET	0	1514	55
57	MAC_IN	20252.24	ACCESSPOINT	2	2	1	2	0	ETHERNET	0	1514	56
58	MAC_OUT	20252.24	ACCESSPOINT	2	1	1	2	0	WLAN	0	1488	57
59	MAC_OUT	20252.24	ACCESSPOINT	2	1	1	2	0	WLAN	CS	1488	58
60	MAC_OUT	20302.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_DIFS_END	1488	59
61	MAC_OUT	20322.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	60
62	MAC_OUT	20342.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	61
63	MAC_OUT	20362.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	62
64	MAC_OUT	20382.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	63
65	MAC_OUT	20402.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	64
66	MAC_OUT	20422.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	65
67	MAC_OUT	20442.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	66
68	MAC_OUT	20462.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	67
69	MAC_OUT	20482.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	68
70	MAC_OUT	20502.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	69
71	MAC_OUT	20522.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	70
72	MAC_OUT	20542.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	71
73	MAC_OUT	20562.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	72
74	MAC_OUT	20582.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_EVENT_BACKOFF	1488	73
75	PHYSICAL_OUT	20582.24	ACCESSPOINT	2	1	1	2	0	WLAN	IEEE802_11_PHY_TXSTART_REQUEST	1528	74
76	PHYSICAL_IN	21886.25	NODE	1	1	1	2	0	WLAN	0	1528	75
78	MAC_IN	21886.25	NODE	1	1	1	2	0	WLAN	RECEIVE_MPDU	1528	76
79	NETWORK_IN	21886.25	NODE	1	1	1	2	0	IPV4	0	1528	78
80	MAC_OUT	21886.25	NODE	1	1	1	2	0	WLAN	SEND_ACK	1528	78
81	TRANSPORT_IN	21886.25	NODE	1	1	1	2	0	UDP	0	1468	79
83	APPLICATION_IN	21886.25	NODE	1	1	1	2	0	APPLICATION	0	1460	81

Figure 1-35: Sub events like CS, IEEE802.11 event DIFS END and IEEE802.11 event BACKOFF event times

In this manner the event trace can be used to understand the flow of events in NetSim Discrete Event Simulator.

1.3.5 Discussion

In NetSim each event occurs at a particular instant in time and marks a change of state in the system. Between consecutive events, no change in the system is assumed to occur. Thus the simulation can directly jump in time from one event to the next.

This contrasts with continuous simulation in which the simulation continuously tracks the system dynamics over time. Because discrete-event simulations do not have to simulate every time slice, they can typically run much faster than the corresponding continuous simulation.

Understanding NetSim's Event trace and its flow is helpful for source code customization and debugging. The event IDs provided in the event trace can be used to go to a specific event while debugging. This capability is valuable when verifying the correctness of changes made to the source code.

1.4 Plot the characteristic curve of throughput versus offered traffic for a Pure and Slotted Aloha system (Level 2)

1.4.1 Theory

ALOHA provides a wireless data network. It is a multiple access protocol (this protocol is for allocating a multiple access channel). There are two main versions of ALOHA: pure and slotted. They differ with respect to whether time is divided up into discrete slots into which all frames must fit.

1.4.1.1 Pure Aloha

In Pure Aloha, users transmit whenever they have data to be sent. There will be collisions and the colliding frames will then be retransmitted. In NetSim's Aloha library, the sender waits a random amount of time per the exponential back-off algorithm and sends it again. The frame is discarded when the number of collisions a packet experiences crosses the "Retry Limit" - a user settable parameter in the GUI.

Let ''frame time'' denotes the amount of time needed to transmit the standard, fixed-length frame. In this experiment point, we assume that the new frames generated by the stations are modeled by a Poisson distribution with a mean of N frames per frame time. If N > 1, the nodes are generating frames at a higher rate than the channel can handle, and nearly every frame will suffer a collision. For reasonable throughput, we would expect 0 < N < 1. In addition to the new frames, the stations also generate retransmissions of frames that previously suffered collisions.

The probability of no other traffic being initiated during the entire vulnerable period is given by e^{-2G} which leads to $S = G \times e^{-2G}$ where, S is the throughput and G is the offered load. The units of S and G is frames per frame time.

G is the mean of the Poisson distribution followed by the transmission attempts per frame time, old and new combined. Old frames mean those frames that have previously suffered collisions.

The maximum throughput occurs at $G = 0.5$, with $S = \frac{1}{2e}$, which is about 0.184. In other words, the best we can hope for is a channel utilization of 18%. This result is not very encouraging, but with everyone transmitting at will, we could hardly have expected a 100% success rate.

1.4.1.2 Slotted Aloha

In slotted Aloha, time is divided up into discrete intervals, each interval corresponding to one frame. In Slotted Aloha, a node is required to wait for the beginning of the next slot in order to send the next packet. The probability of no other traffic being initiated during the entire vulnerable period is given by e^{-G} which leads to $S = G \times e^{-G}$. It is easy to compute that Slotted Aloha peaks at G = 1, with a throughput of $s = \frac{1}{e}$ or about 0.368.

1.4.2 Offered load and throughput calculations[1]

Using NetSim, the attempts per packet time (G) can be calculated as follows.

$$G = \frac{Number\ of\ packets\ transmitted \times PacketTime(s)}{SimulationTime\ (s)}$$

where, G is Attempts per packet time. We derive the above formula keeping in mind that (i) NetSim's output metric, the number of packets transmitted, is nothing but the number of attempts, and (ii) since packets transmitted is computed over the entire simulation time, the number of "packet times" would be $\frac{SimulationTime(s)}{PacketTransmissionTime(s)}$, which is in the denominator. Note that in NetSim the output metric Packets transmitted is counted at link (PHY layer) level. Hence MAC layer re-tries are factored into this metric.

The throughput (in Mbps) per packet time can be obtained as follows.

$$S = \frac{Number\ of\ packets\ successful \times PacketTime(s)}{SimulationTime\ (s)}$$

where, S = Throughput per packet time. In case of slotted aloha packet (transmission) time is equal to slot length (time). The packet transmission time is the PHY layer packet size in bits divided by the PHY rate in bits/s. Considering the PHY layer packet size as 1500B, and the PHY rate as 10 Mbps, the packet transmission time (or packet time) would be $\frac{1500\times8}{10\times10^6} = 1200\ \mu s$.

In the following experiment, we have taken packet size as 1460 B (Data Size) plus 28 B (Overheads) which equals 1488 B. The PHY data rate is 10 Mbps and hence packet time is equal to 1.2 milliseconds.

1.4.3 Network Set Up

Open NetSim and click on **Experiments> Legacy Networks> Throughput versus load for Pure and Slotted Aloha> Pure Aloha** then click on the tile in the middle panel to load the example as shown in below figure

[1] A good reference for this topic is Section 4.2.1: ALOHA, of the book, Computer Networking, 5th Edition by Tanenbaum and Wetherall

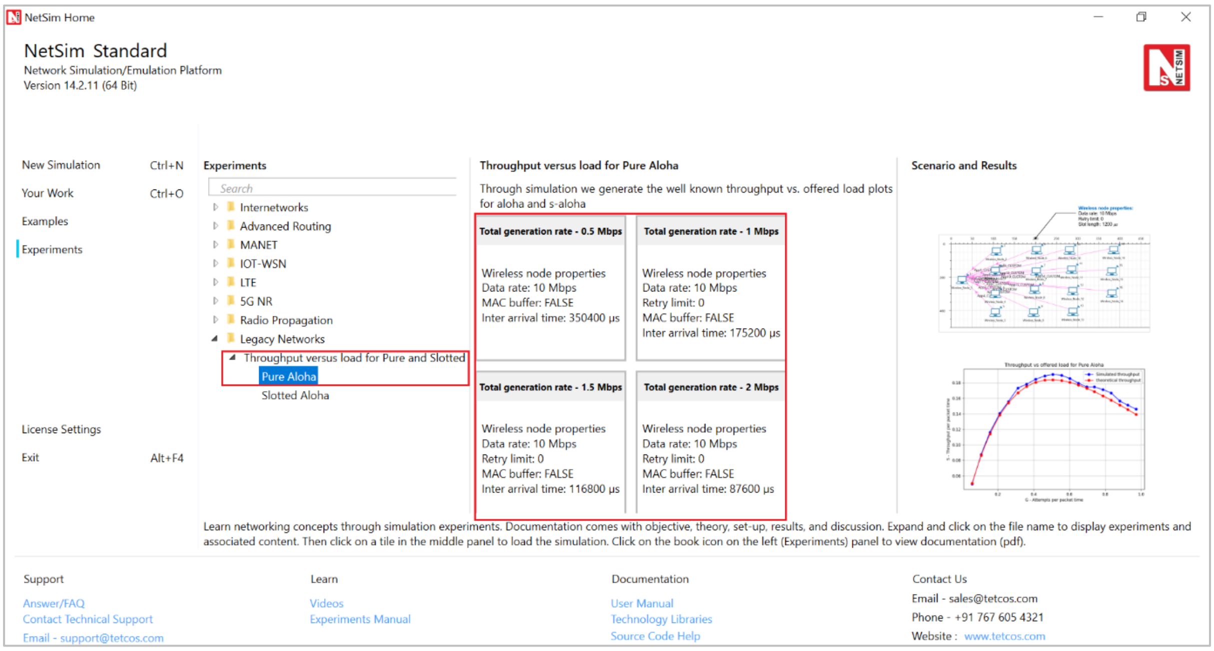


Figure 1-36: List of scenarios for the example of Throughput versus load for Pure and Slotted Aloha

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 1-37.

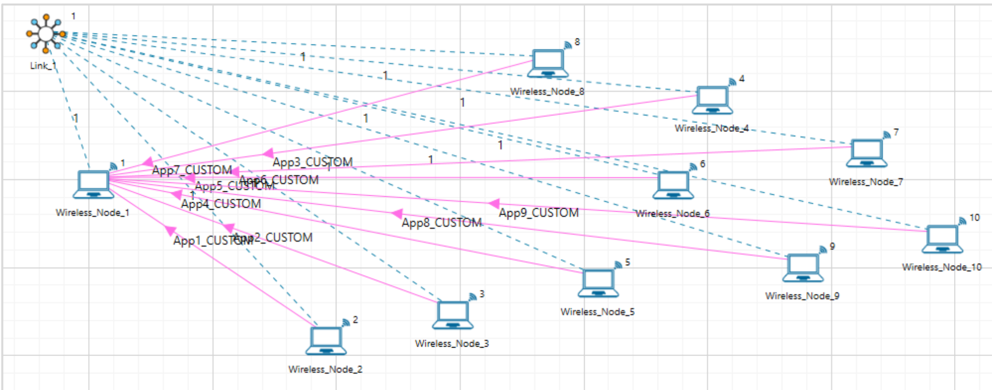


Figure 1-37: Network set up for studying the Pure aloha

Pure Aloha: Input for 0.5 Mbps Generation rate sample

Step 1: Drop 16 nodes (i.e., 15 Nodes are generating traffic.) Node 2 to 16 generates traffic to Node 1. The properties of transmitter nodes which transmits data to Node 1 are given in the below table.

Step 2: Click on the wireless node and expand the property panel on right and set the properties as mentioned in below table.

Wireless Node Properties	
Interface1 Wireless (PHYSICAL LAYER)	
Data Rate (Mbps)	10
Interface1 Wireless (DATALINK LAYER)	
Retry Limit	0
MAC Buffer	FALSE

Slot Length(µs)	1200

Table 1-2: Wireless Node Properties

(Note: Slot Length(µs) parameter present only in Slotted Aloha → Wireless Node Properties → Interface_1 (Wireless))

Step 3: Click on Adhoc link and expand property panel on right and set channel characteristics to **No Path Loss**.

Step 4: Configure a custom application from the Set Traffic tab in ribbon on top. The properties are set according to the values given in the below Table 1-3.

Application 1 Properties		
Application method	Unicast	
Application type	Custom	
Source Id	2	
Destination Id	1	
Transport protocol	UDP	
Packet size	Distribution	Constant
	Value (Bytes)	1460
Inter Arrival Time	Distribution	Exponential
	Packet Inter Arrival Time (µs)	350400

Table 1-3: For Application 1 Properties

- Similarly create 14 more application, i.e., Source Id as 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16 and Destination Id as 1, set Packet Size and Inter Arrival Time as shown in above table.

Step 5: Run the simulation for 10 seconds.

NOTE: Users can refer to Section 3.10 in the user manual for configuring the device properties in all wireless nodes at once using rapid configurator.

Input for 1 Mbps Generation rate sample

Step 1: We increase the network load by increasing the generation rate in both pure and slotted Aloha scenarios.

In the Aloha example, we incrementally increase the total generation rate up to 10 Mbps in steps of 0.5 Mbps, as shown in the table below. Here, the generation rate per node is calculated by dividing the total generation rate by the number of transmitters.

$$Generation\ rate\ (per\ node) = \frac{Generation\ rate\ (Total)}{No.of\ Transmitters}$$

Number of nodes generating traffic	Generation rate (per node) Mbps	Generation rate (In total) Mbps	Packet size (Bytes)	Inter packet arrival time (µs)
15	0.03	0.5	1460	350400
15	0.07	1	1460	175200
15	0.10	1.5	1460	116800

15	0.13	2	1460	87600
15	0.17	2.5	1460	70080
15	0.20	3	1460	58400
15	0.23	3.5	1460	50057
15	0.27	4	1460	43800
15	0.30	4.5	1460	38933
15	0.33	5	1460	35040
15	0.37	5.5	1460	31855
15	0.40	6	1460	29200
15	0.43	6.5	1460	26954
15	0.47	7	1460	25029
15	0.50	7.5	1460	23360
15	0.53	8	1460	21900
15	0.57	8.5	1460	20612
15	0.60	9	1460	19467
15	0.63	9.5	1460	18442
15	0.67	10	1460	17520

Table 1-4: Table shows the inter packet arrival times for various generation rates in Aloha.

Slotted ALOHA: Input for 0.5 Mbps Generation Rate sample

Step 1: Drop 16 nodes (i.e., 15 Nodes are generating traffic.)

Nodes 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16 transmit data to Node 1 and set properties for nodes and application as mentioned above.

Continue the experiment by increasing the total no of generation rate up to 20 Mbps in the steps of 1 Mbps as shown in the below Table 1-5.

Number of nodes generating traffic	Generation rate (per node) Mbps	Generation rate (In total) Mbps	Packet size (Bytes)	IAT (µs)
15	1	0.07	1460	175200
15	2	0.13	1460	87600
15	3	0.20	1460	58400
15	4	0.27	1460	43800
15	5	0.33	1460	35040
15	6	0.40	1460	29200
15	7	0.47	1460	25029
15	8	0.53	1460	21900
15	9	0.60	1460	19467
15	10	0.67	1460	17520
15	11	0.73	1460	15927
15	12	0.80	1460	14600
15	13	0.87	1460	13477
15	14	0.93	1460	12514

15	15	1.00	1460	11680
15	16	1.07	1460	10950
15	17	1.13	1460	10306
15	18	1.20	1460	9733
15	19	1.27	1460	9221
15	20	1.33	1460	8760

Table 1-5: Table shows the inter packet arrival times for various generation rates in Sl. Aloha

1.4.4 Output

Comparison Table: The values for the total number of packets transmitted and collided are obtained from the NetSim simulation results window after running the simulation. The network statistics are provided in the table below, along with calculation of throughput per packet time (S) and the number of packets transmitted per packet time (G).

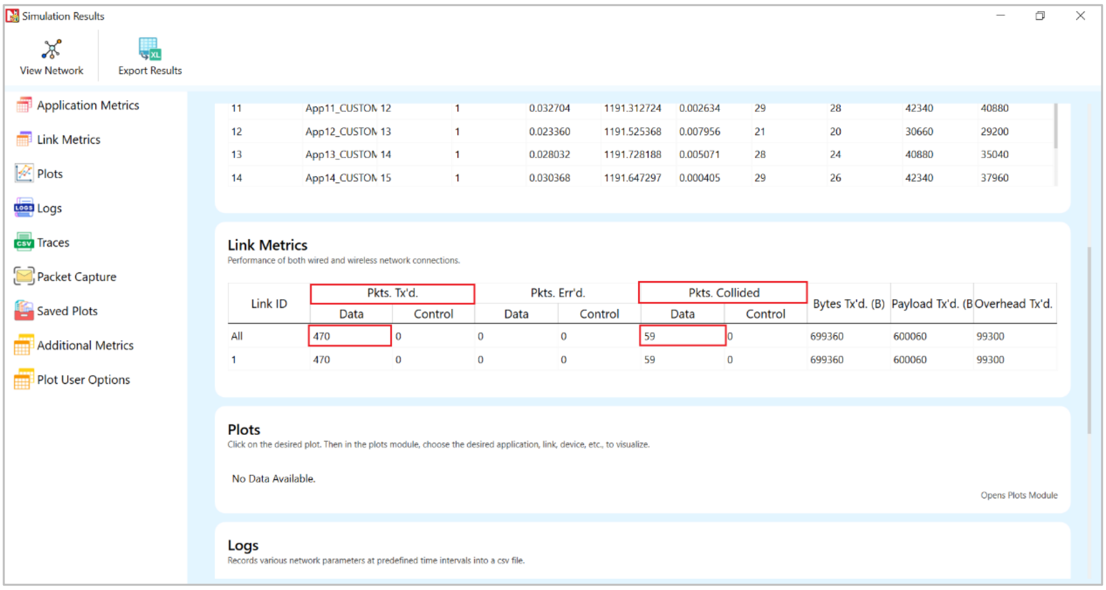


Figure 1-38: Simulation results window showing the No. of packets transmitted and collided.

Pure Aloha:

Total no of nodes generating traffic (in Total)	Total number of Packets Transmitted	Total number of Packets Collided	Successful Packets (Packets Transmitted -Packets Collided)	Attempts per packet time(G)	Throughput per packet time(S)	Throughput per packet time. Theoretical ($S = G * e^{-2G}$)
0.5	470	59	411	0.0564	0.0493	0.0504
1	891	162	729	0.1069	0.0875	0.0863
1.5	1299	331	968	0.1558	0.1162	0.1141
2	1759	587	1172	0.2110	0.1406	0.1384
2.5	2159	861	1298	0.2590	0.1558	0.1543
3	2603	1158	1445	0.3123	0.1734	0.1672
3.5	3009	1525	1484	0.3610	0.1781	0.1754
4	3426	1892	1534	0.4111	0.1841	0.1807
4.5	3845	2274	1571	0.4614	0.1885	0.1834

5	4225	2641	1584	0.507	0.1901	0.1839
5.5	4636	3055	1581	0.5563	0.1897	0.1829
6	5010	3469	1541	0.6012	0.4163	0.1806
6.5	5418	3926	1492	0.6501	0.1790	0.1771
7	5800	4346	1454	0.696	0.1745	0.1730
7.5	6154	4698	1456	0.7384	0.1747	0.1686
8	6558	5132	1426	0.7869	0.1711	0.1631
8.5	6938	5548	1390	0.8326	0.1668	0.1575
9	7334	6034	1300	0.8801	0.156	0.1514
9.5	7703	6445	1258	0.9244	0.1510	0.1455
10	8095	6886	1209	0.9715	0.1451	0.1392

Table 1-6: Total No. of Packets transmitted, Collided, Attempts per packet time and throughput per packet time for Pure Aloha.

Slotted Aloha

Number of nodes generating traffic	Total number of Packets Transmitted	Total number of Packets Collided	Successful Packets (Packets Transmitted-Packets Collided)	Attempts per packet time(G)	Throughput per packet time(S)	Throughput per packet time. Theoretical (S = $G * e^{-G}$)
1	887	105	782	0.1064	0.0938	0.0957
2	1748	326	1422	0.2097	0.1706	0.1701
3	2568	679	1889	0.3081	0.2267	0.2264
4	3386	1135	2251	0.4063	0.2701	0.2706
5	4155	1581	2574	0.4986	0.3089	0.3028
6	4913	2121	2792	0.5895	0.3350	0.3270
7	5671	2765	2906	0.6805	0.3487	0.3446
8	6377	3356	3021	0.7652	0.3625	0.3560
9	7116	3988	3128	0.8541	0.3754	0.3636
10	7840	4724	3116	0.9408	0.3740	0.3672
11	8495	5343	3152	1.0194	0.3782	0.1327
12	9200	6003	3197	1.104	0.3836	0.1214
13	9847	6683	3164	1.1816	0.3797	0.1112
14	10537	7411	3126	1.2645	0.3750	0.3571
15	11158	8135	3023	1.3390	0.3629	0.3510
16	11737	8791	2946	1.4085	0.3536	0.3444
17	12374	9426	2948	1.4849	0.3536	0.0762
18	13006	10170	2836	1.5608	0.3402	0.3277
19	13575	10875	2700	1.6292	0.324	0.3195
20	14163	11471	2692	1.6996	0.3230	0.0568

Table 1-7: Total No. of Packets Transmitted, Collided, Throughput per packet time and throughput per packet time for Slotted Aloha

Thus, the following characteristic plot for the Pure Aloha and Slotted Aloha is obtained, which agrees well with the theoretical results.

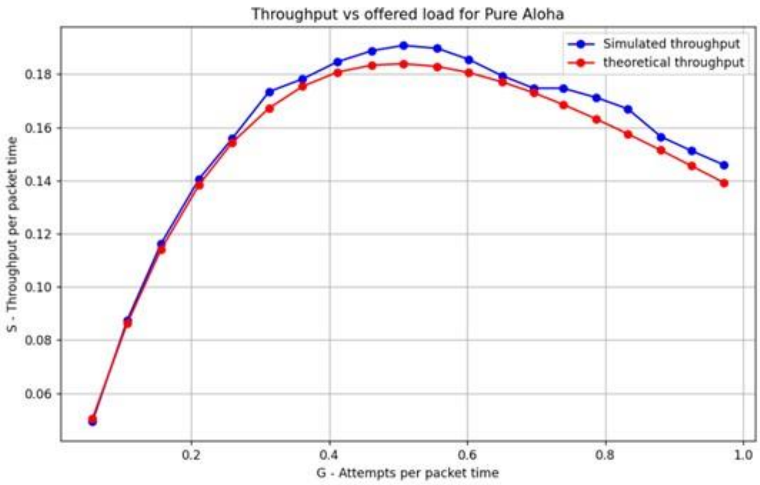


Figure 1-39: Throughput vs offered load for Pure Aloha

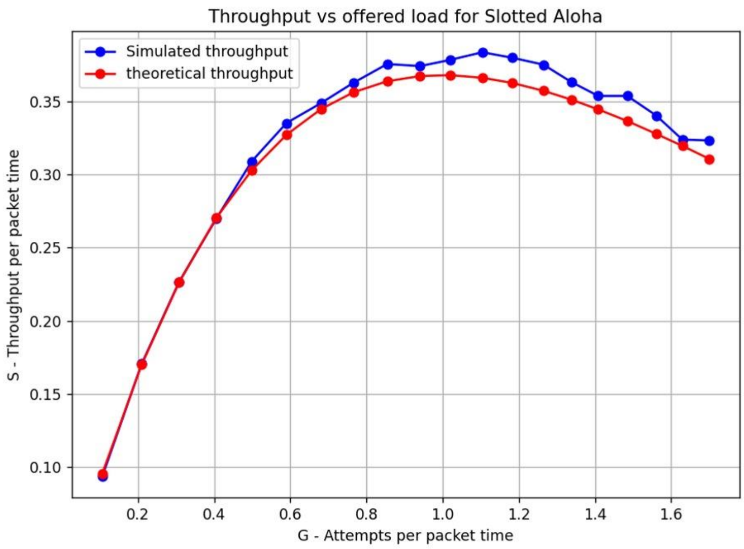


Figure 1-40: Throughput vs. Offered load for Slotted Aloha

1.4.5 Exercises

1. Redo the experiment with PHY data rates of 2 Mbps and 5 Mbps, and then similarly compare the Total number of packets transmitted, Collided, Attempts per packet time, and Throughput per packet time for Pure Aloha and Slotted Aloha. Plot the characteristic curve of the same.

    - Slot time should be 6000 μs for PHY data rate of 2 Mbps.
    - Slot time should be 2400 μs for PHY data rate of 5 Mbps.

2 Network Performance

2.1 Data traffic types and network performance measures (Level 1)

2.1.1 Network Performance

2.1.1.1 Throughput

We begin by understanding the *throughput* of a *flow* through a *subsystem*. A flow is a stream of bits of interest, e.g., all bits flowing through a link in one direction, the bits corresponding to a video stored on a server and playing at a device, etc. A subsystem is any part of the network through which bits can flow, e.g., a link in a network, a subnetwork, a router, an entire network except for the endpoints of a particular flow.

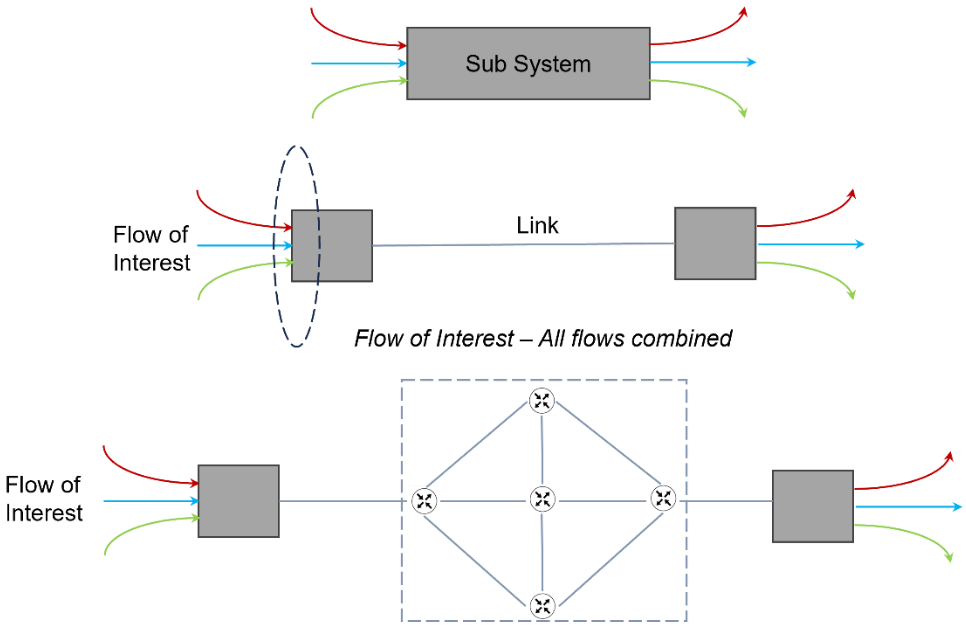


Figure 2-1: We define the throughput of a flow through a sub system. In the top image, the flow of interest is in blue. In the second figure the sub system is the link and throughput is measured on all flows combined. In the bottom image, again the flow of interest is in blue, while the sub system is a network.

In the context of NetSim, we have:

- Application throughput: The subsystem is the entire network, except for the two entities involved in the application, and the flow is the bit stream of that application.
- Point-to-point link throughput: The subsystem is the link, and the flow is all the bits flowing through the link in a specified direction.

Let $A(t)$ be the number of bits of the flow during the interval $[0, t]$, then the throughput up to t, is equal to $\frac{A(t)}{t}$, and the long-run average throughput is equal to $\lim_{t\to\infty} \frac{A(t)}{t}$.

2.1.1.2 Delay

Delay is reported for an entity, e.g., packets in a stream flowing through a subsystem, a user request (such as a connection setup request), a task such as a file transfer.

Let t_k be the time instant at which delay for the k^{th} entity starts,

- e.g., the time of arrival of a packet at a router, the time of a connection request initiation, the time at which a file transfer is requested.

Let u_k be the time instant at which the delay of the k^{th} entity ends,

- e.g., the time at which a packet leaves the router, the time at which a connection request is completed, or the time at which a file transfer is completed.

Then $d_k = u_k - t_k$ is the delay of the k^{th}entity among the entities of interest.

- e.g., the delay of the k^{th} packet passing through a router, the delay for the connection set up, or the file transfer delay.

The average delay over last K entities is equal to

$$\frac{1}{K}\sum_{k=(N(K-1))}^{N} d_k$$

This is called a "moving window" average with a window of K, and will track any changes in the system. As it can be seen, the implementation of the above average requires the storage of the past K values of d_k. An averaging approach that does not require such storage is the Exponentially Weighted Moving Average (EWMA)

$$\bar{d}(N) = \sum_{k=1}^{N}(1-\alpha)^{N-k}\alpha\, d_k = \bar{d}(N-1) + \alpha(d_N - \bar{d}\,(N-1))$$

which is roughly like averaging over the past $1/\alpha$ values of d_k; for example, $\alpha = 0.001$ gives an averaging window of 1000. The limiting average delay is equal to,

$$\lim_{K\to\infty}\left(\frac{1}{K}\right)\sum_{k=1}^{K} d_k$$

This limiting average delay is meaningful if the system is statistically invariant over a long time.

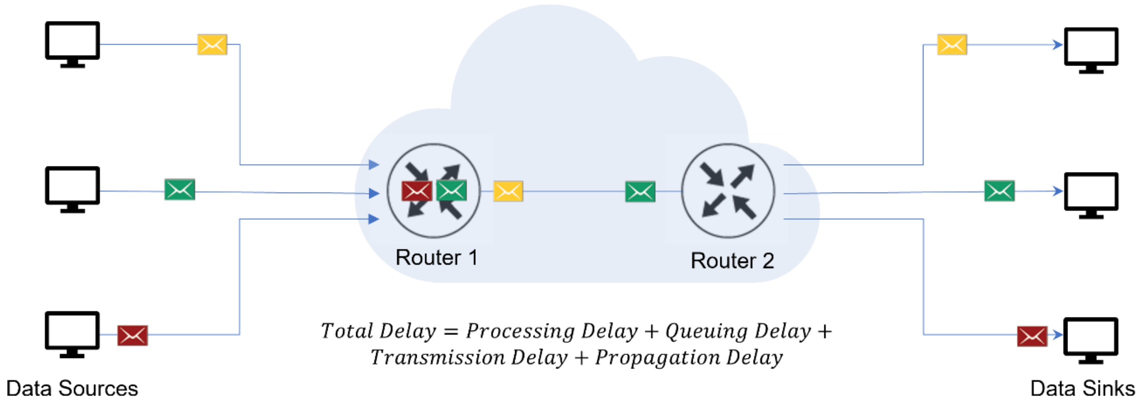


Figure 2-2: Components of network delay

The following are some examples of the network aspects that determine the delay of an entity.

- Packet delay through a router
  - Processing delay: in moving the packet through the "fabric" to the queue at its exit port.
  - Queuing delay: at the exit port
  - Transmission delay: in sending out the packet bits onto the digital transmission link
- File transfer delay
  - Connection set up delay: between the request packet from the user, until the connection is set up and transfer starts.
  - Data transfer delay is the time taken to transmit all the bytes in the file reliably to the user, so that the user has an exact copy of the file, and the file shows up at the user level in the user's computer.

Generally, the delay of an entity in a network subsystem will depend on:

- the "capacity" of the subsystem (e.g., the bit rate of a link), and
- the way the capacity is allotted to the various entities being transmitted over the link.

The delay through a subsystem will include the propagation delay (due to speed of light), router queuing delay, router processing delay, and transmission delay.

2.1.2 Types of Traffic

2.1.2.1 Elastic Traffic

Elastic traffic is generated by applications that do not have any intrinsic "time dependent" rate requirement and can, thus, be transported at arbitrary transfer rates. For example, the file transfer application only requires that each file be transferred reliably but does not impose a transfer rate requirement. Of course, it is desirable that the file is transferred quickly, but that is not a part of the file transfer application requirement.

The following are the QoS requirements of elastic traffic.

- Transfer delay and delay variability can be tolerated. An elastic transfer can be performed over a wide range of transfer rates, and the rate can even vary over the duration of the transfer.
- The application cannot tolerate data loss.
  - This does not mean, however, that the network cannot lose any data. Packets can be lost in the network - owing to uncorrectable transmission errors or buffer overflows - provided that the lost packets are recovered by an automatic retransmission procedure.
  - Thus, effectively the application would see a lossless transport service. Because elastic sources do not require delay guarantees, the delay involved in recovering lost packets can be tolerated.

2.1.2.2 Stream Traffic

This source of traffic has an intrinsic "time-dependent" behavior. This pattern must be preserved for faithful reproduction at the receiver. The packet network will introduce delays: a fixed propagation delay and queueing delay that can vary from packet to packet. Applications such as real-time interactive speech or video telephony are examples of stream sources.

The following are typical QoS requirements of stream sources.

- Delay (average and variation) must be controlled. Real-time interactive traffic such as that from packet telephony would require tight control of end-to-end delay; for example, for packet telephony the end-to-end delay may need to be controlled to less than 200 ms, with a probability more than 0.99.
- There is tolerance for data loss. Because of the high levels of redundancy in speech and images, a certain amount of data loss is imperceptible. As an example, for packet voice 5 to 10% of the packets can be lost without significant degradation of the speech quality.

2.1.3 Featured Examples

2.1.3.1 Analyzing throughput and file transfer delay for elastic traffic.

Open NetSim and click on Experiments> Internetworks> Network Performance> Analyzing throughput and file transfer delay for elastic traffic and then click on the tile in the middle panel to load the example as shown below.

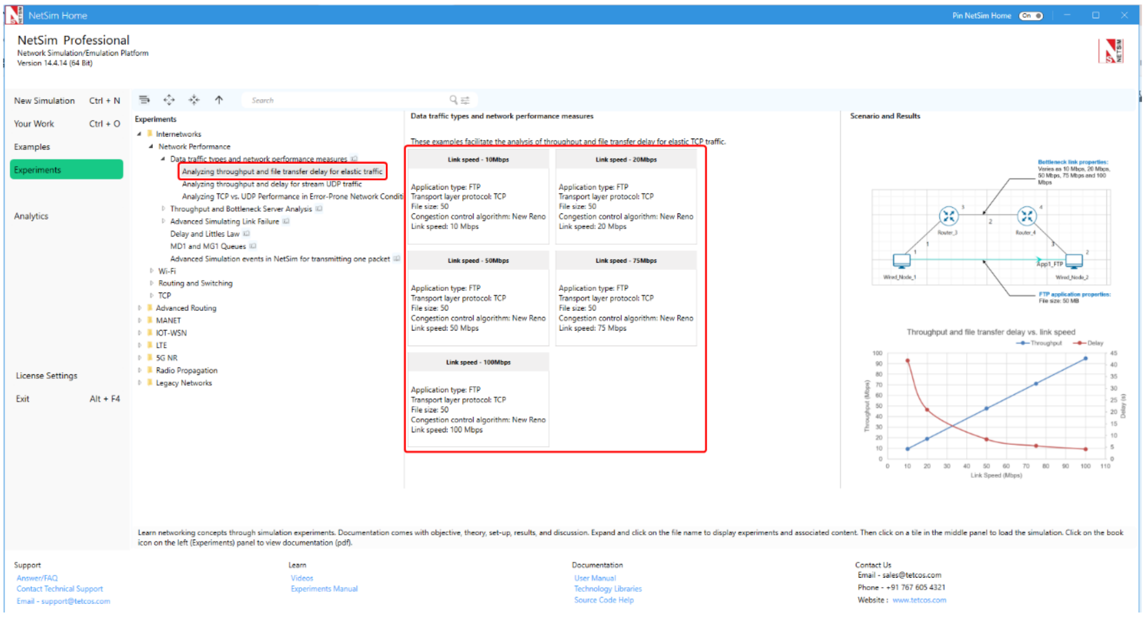


Figure 2-3: List of scenarios for the example of Data traffic types and network performance measures

Elastic Traffic is configured as FTP over TCP. We create a network with client, server and 2 routers (R1, R2), and configure an FTP application between client and server.

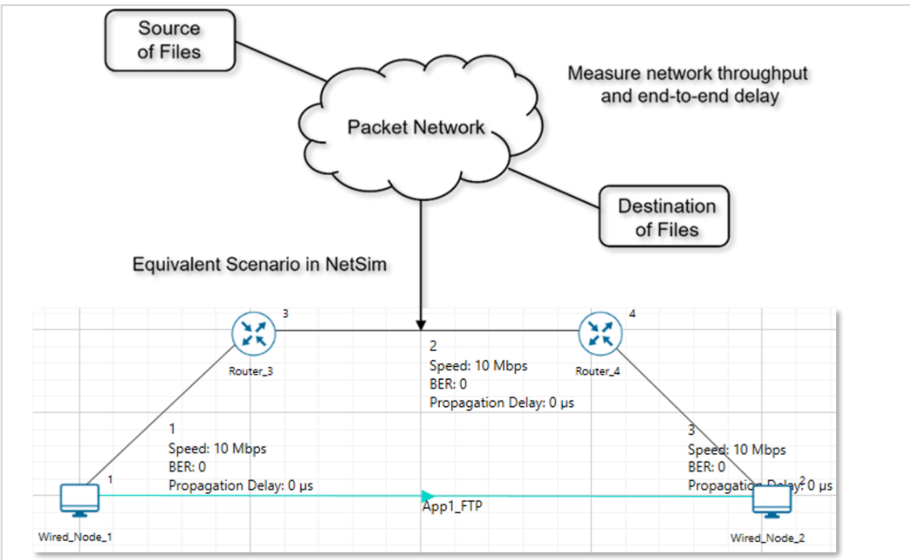


Figure 2-4: Virtual scenario of FTP application between source and destination created in NetSim.

One file of size 50 MB will be transferred from the server to the client with Application start time = 0, Application end time = 1s. This setting is to ensure that only one file is generated.

We set TCP congestion control algorithm as New Reno with window scaling enabled. Then, in the links, we set BER = 0 and Propagation delay = 0.

Enable packet trace from Configure Reports tab.

Finally, we run the simulation for 100 seconds with (all three) link speeds varying as 10, 20, 50, 75 and 100 Mbps respectively.

Configuration	Value
Network components	Client, Server, R1, R2
Application properties	
Traffic type	Elastic traffic configured as FTP over TCP
Application start time	0
Application end time	1s
File size	50 MB
Wired node(Transport layer properties)	
Congestion control algorithm	New Reno with window scaling enabled
Link settings	
Link speed	10, 20, 50, 75, 100 Mbps
Medium property	BER=0, Propagation Delay=0
Simulation time	
100s	

Table 2-1: Configuration properties.

2.1.3.1.1 Results and discussions

Link Speed (Mbps)	Application throughput (Mbps)	No. of packets received	Time taken to transfer the entire file in seconds. Simulation output	$FileTransferTime(s) \approx \frac{\left(50\times 8\times\frac{1526}{1460}\right)}{LinkSpeed}$ (Theoretical)
10	9.52	34247	20.98	41.80
20	18.98	34247	10.53	20.90
50	47.45	34247	4.21	8.36
75	71.18	34247	2.80	5.57
100	94.92	34247	2.10	4.18

Table 2-2: NetSim results and comparison against theoretical calculations.

Given that the application layer packet size is 1460B packets, a 50 MB file gets fragmented into $\frac{50\times 10^6}{1460} = 34247$ packets. The PHY layer packet size is equal to 1460 B plus 66 B of UDP, IP, MAC overheads, which is equal to 1526B. Now, theoretically the

$$FileTransferTime\ (s) = \frac{FileSize(MB)\times 8\times\left(\frac{PacketSize + OH}{PacketSize}\right)}{LinkSpeed\ (Mbps)}$$

In NetSim, Time taken to transfer the entire file can be computed using the packet trace. It is the time when the last packet was received at the destination less the time at which the file was generated at the source. We see that simulation output agrees well with theory. The slight difference is due to additional time (0.03s) taken for initial connection establishment which is not accounted for in the theoretical calculation.

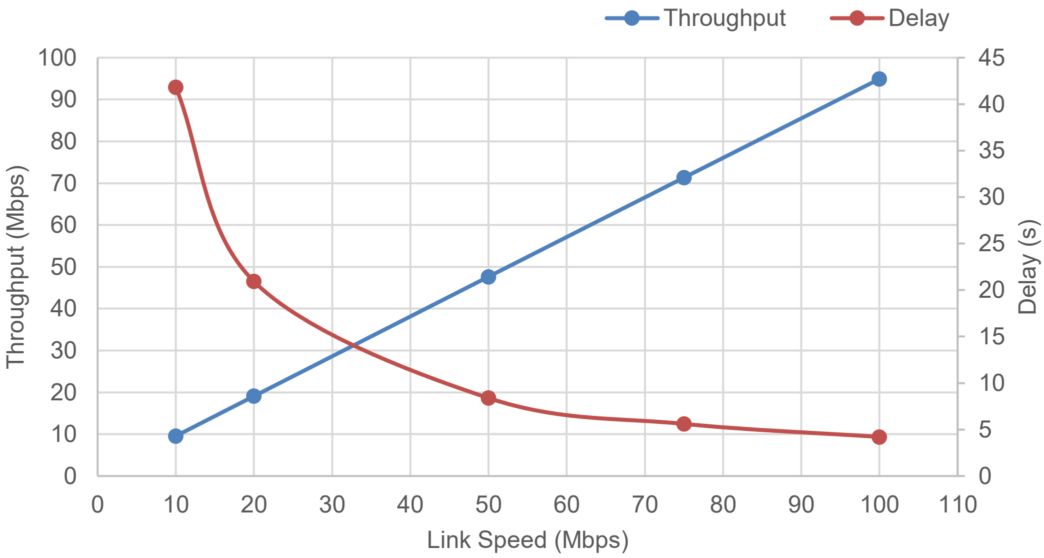


Figure 2-5: Throughput and file transfer delay vs. link speed.

2.1.3.2 Analyzing throughput and delay for stream UDP traffic

Open NetSim and click on Experiments> Internetworks> Network Performance> **Analyzing throughput and delay for stream UDP traffic** and then click on the tile in the middle panel to load the example as shown below.

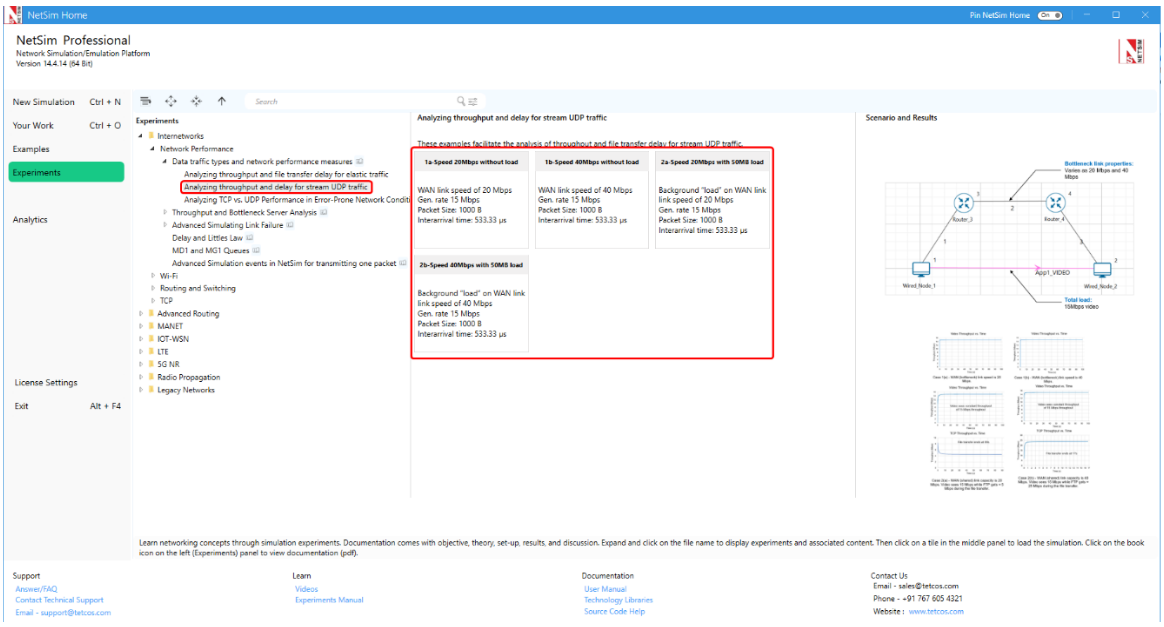


Figure 2-6: List of scenarios for the example of Data traffic types and network performance measures

Stream traffic is configured as streaming video over UDP. We create a scenario with 2 wired nodes (N1, N2), 2 routers (R1, R2). Then, we set the processing delay at routers = 0.

In the links,

- Access links (Node-Router): We set the link speed as 100 Mbps, Propagation delay = 0 ms and the BER = 0.
- WAN links (Router to Router): We set link speeds as 20 Mbps in cases 1a and 2a, and as 40 Mbps in cases 1b and 2b. Then we set BER = 0, and Propagation delay = 5 ms.

Enable packet trace from Configure Reports tab.

We then simulate two cases as explained below.

- Case #1: Configure a video call between N1 and N2
  - Gen. rate 15 Mbps with Pkt. Size = 1000 B (Const.), and Interarrival time $= 533.33\ \mu s$ (Const.)). Then we set the transport protocol: UDP.
- Case #1a: WAN Link (Link # 3) speed = 20 Mbps. Case #1b: WAN link speed = 40 Mbps
  - Measure throughput and delay
- Case #2: Introduce background "load" on the WAN Link, by adding two more nodes (N3, N4) and configuring FTP traffic (over TCP) between them.
  - We configure a 50 MB file transfer between N3 and N4.
- Case #2a: WAN Link (Link # 3) speed = 20 Mbps. Case #2b: WAN link speed = 40 Mbps
  - Measure throughput and delay

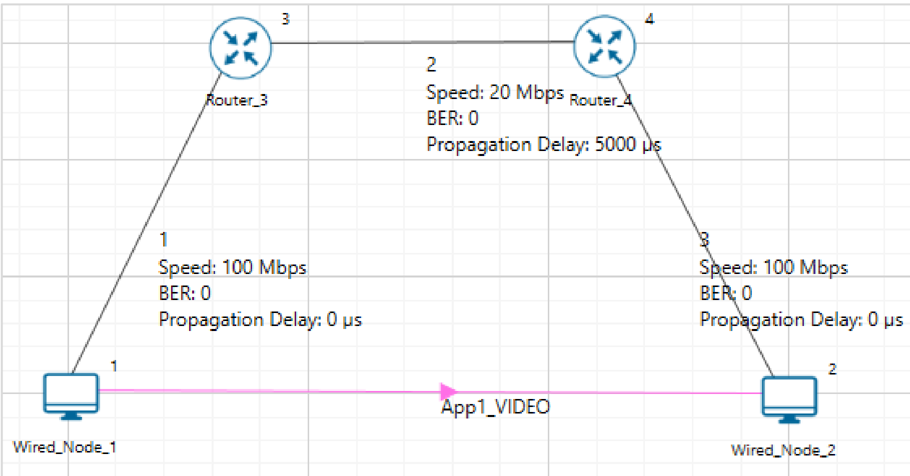


Figure 2-7: Case 1(a) with WAN link speed of 20 Mbps

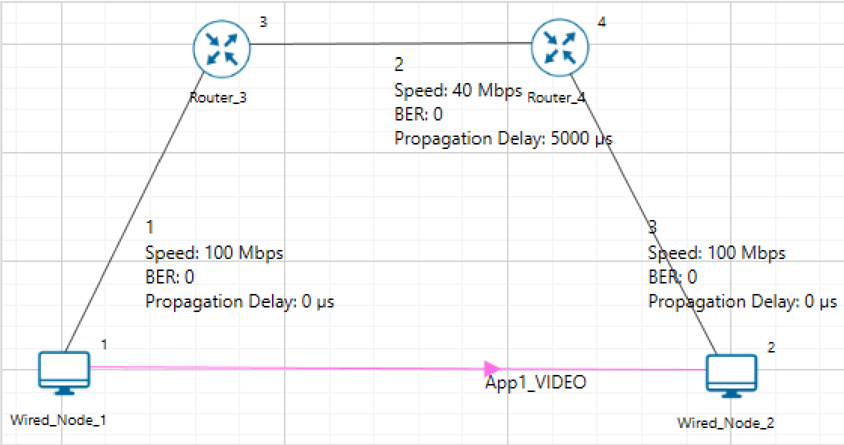


Figure 2-8: Case 1(b) with WAN link speed of 40 Mbps

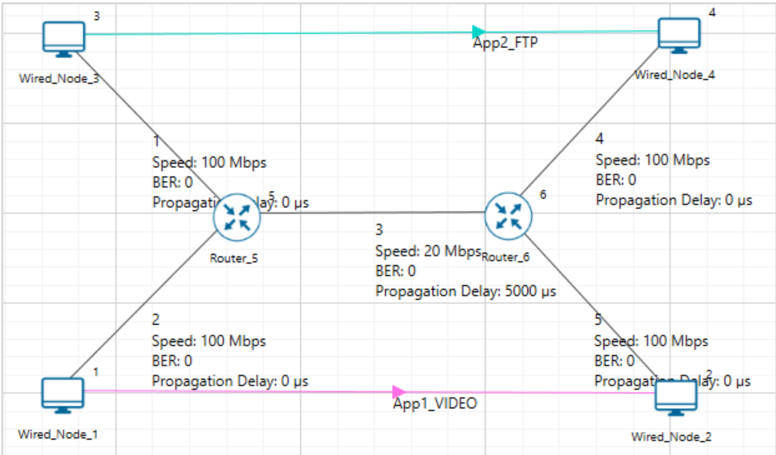


Figure 2-9: Case 2(a) with WAN link speed of 20 Mbps

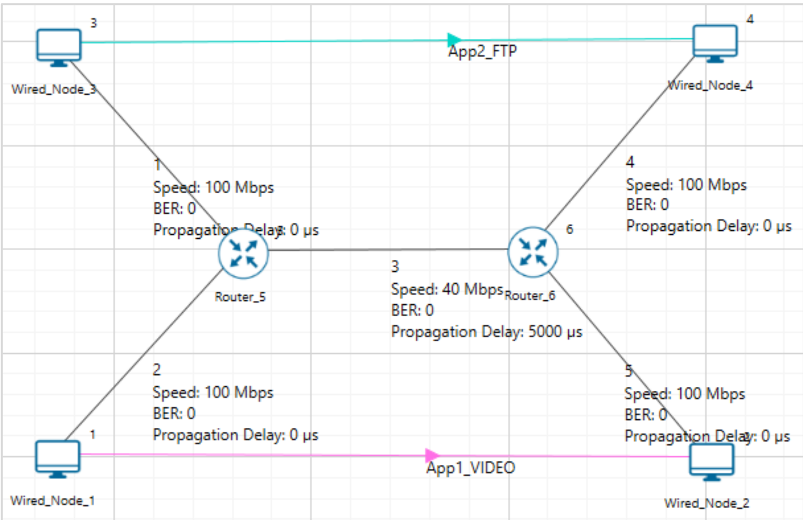


Figure 2-10: Case 2(b) with WAN link speed of 40 Mbps

2.1.3.2.1 Results and observations

Case #	1a	1b	2a	2b
WAN Link Speed (Mbps)	20	40	20	40
WAN link Propagation delay (ms)	5	5	5	5

Total load (Mbps)	15Mbps video	15 Mbps video	15 Mbps video + 50 MB file	15 Mbps video + 50 MB file
Avg. propagation delay (µs). (Three entries, one per link)	0 + 5000 + 0	0 + 5000 + 0	0 + 5000 + 0	0 + 5000 + 0
Avg. pkt. transmission time (µs)	84.32 + 411.2 + 84.32	84.32 + 205.6 + 84.32	84.32 + 411.2 + 84.32	84.32 + 205.6 + 84.32
Avg. Queuing Delay (µs)	0	0	0+93395.4+0	0+1836.82+0
FTP File transfer Time (s)	N/A	N/A	89.39	16.74
Video End-to-end application throughput (Mbps)	14.99	14.99	14.99	14.99
Video Avg. end-to-end Pkt. Delay (µs)	5579.84	5374.24	98975.24	7211.07
FTP End-to-end application throughput (Mbps)	N/A	N/A	4.47	23.88

Table 2-3: Results of the 4 different cases.

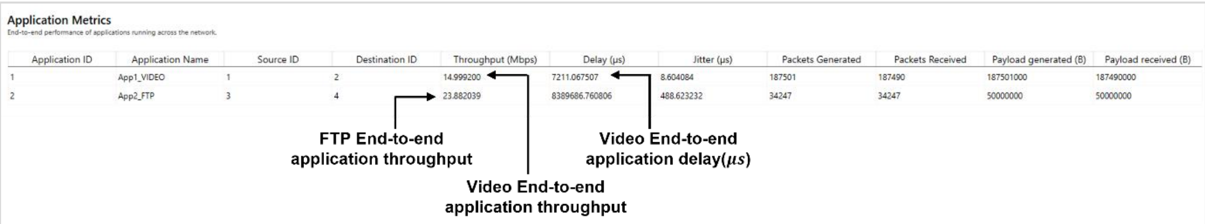


Figure 2-11: Throughput and delay values of case 2(b) from results dashboard.

Since traffic is a stream of packets, all delay calculations are based on the average of the metric taken over all packets. In case 2a and 2b, we have a 50 MB File that is transferred from N3 to N4. It uses the same WAN link that video does. We see that,

$$Avg.\ End\ to\ End\ Delay = Avg. queuing\ delay\ + Avg. transmission\ time + Avg. propagation\ delay$$

For FTP we observe that

$$File\ Transfer\ Time(s) \times FTP\ Throughput(Mbps) = File\ size\ (MB) \times 8$$

i.e,

$$89.39 \times 4.47 = 50 \times 8$$

$$399.57 \approx 400$$

where the factor of 8 is for byte to bit conversion.

2.1.3.2.2 Throughput Plots

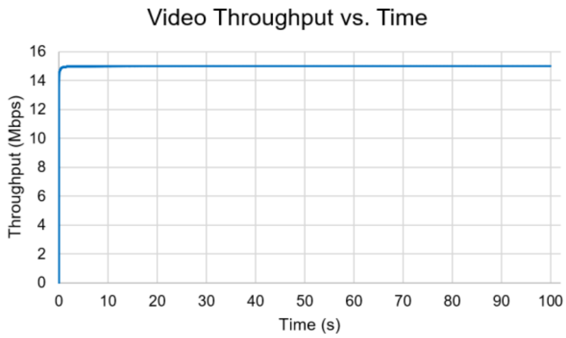


Figure 2-12: Case 1(a) - WAN (bottleneck) link speed is 20 Mbps.

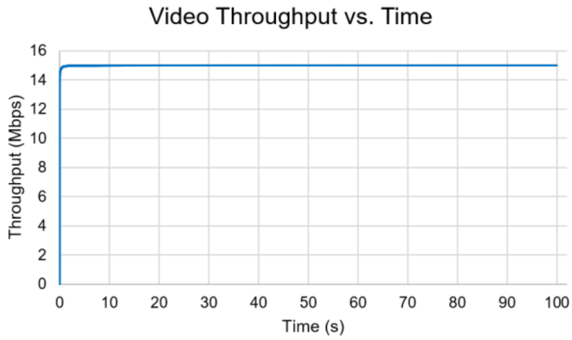


Figure 2-13: Case 1(b) - WAN (bottleneck) link speed is 40 Mbps.

We observe that:

- In both cases video sees a steady throughput of 15 Mbps
- No queuing since the link capacity is 20 Mbps in case 1a and 40 Mbps in case 1b.
- From the Table 2-3 we see that the transmission time is lower in case 1b since it uses a 40 Mbps link.

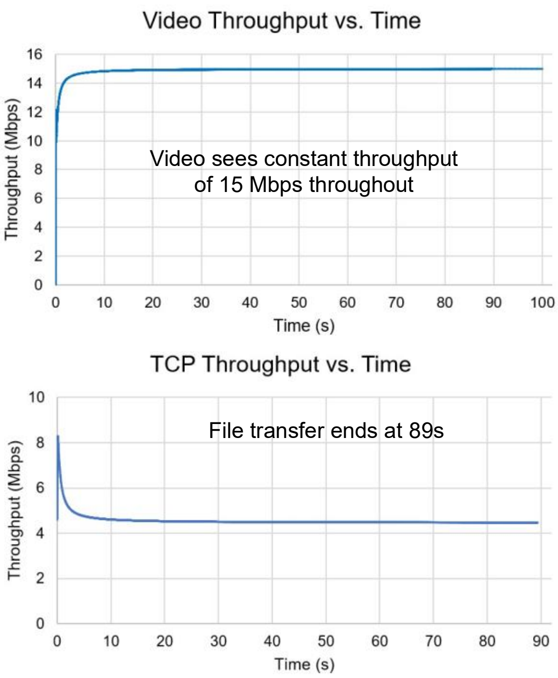


Figure 2-14: Case 2(a) - WAN (shared) link capacity is 20 Mbps. Video sees 15 Mbps while FTP gets ≈ 5 Mbps during the file transfer.

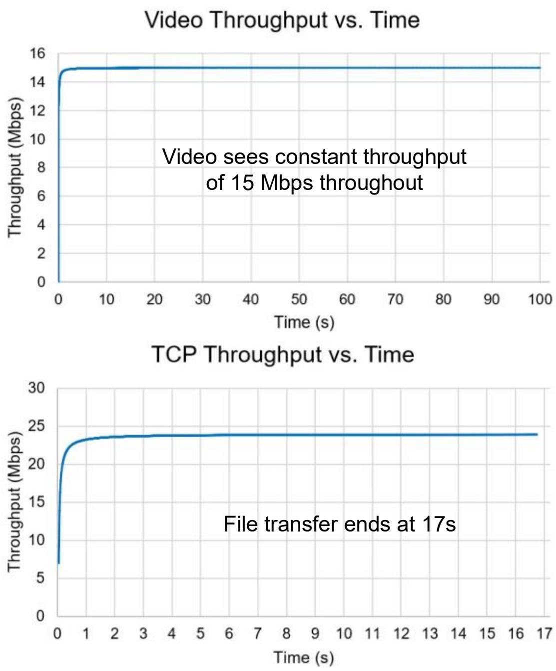


Figure 2-15: Case 2(b) - WAN (shared) link capacity is 40 Mbps. Video sees 15 Mbps while FTP gets ≈ 25 Mbps during the file transfer.

We observe the following:

- Case 2(a)
  - Shared link 20 Mbps with video rate 15 Mbps and file transfer
  - Video throughput is 15 Mbps and FTP throughput is 5 Mbps during transfer.

- Case 2(b)
  - Shared link 40 Mbps with video rate 15 Mbps and file transfer
  - Video throughput is 15 Mbps and FTP throughput is 25 Mbps during transfer.

This clearly shows us that the FTP transfer uses TCP which “adapts” to the “remaining” bandwidth.

2.1.3.3 Analyzing TCP vs. UDP Performance in Error-Prone Network Conditions

Open NetSim and click on Experiments> Internetworks> **Network Performance> Analyzing TCP vs. UDP Performance in Error-Prone Network Conditions** and then click on the tile in the middle panel to load the example as shown below.

In this example, we observe the impact of link errors on the performance of TCP-based applications compared to UDP-based applications.

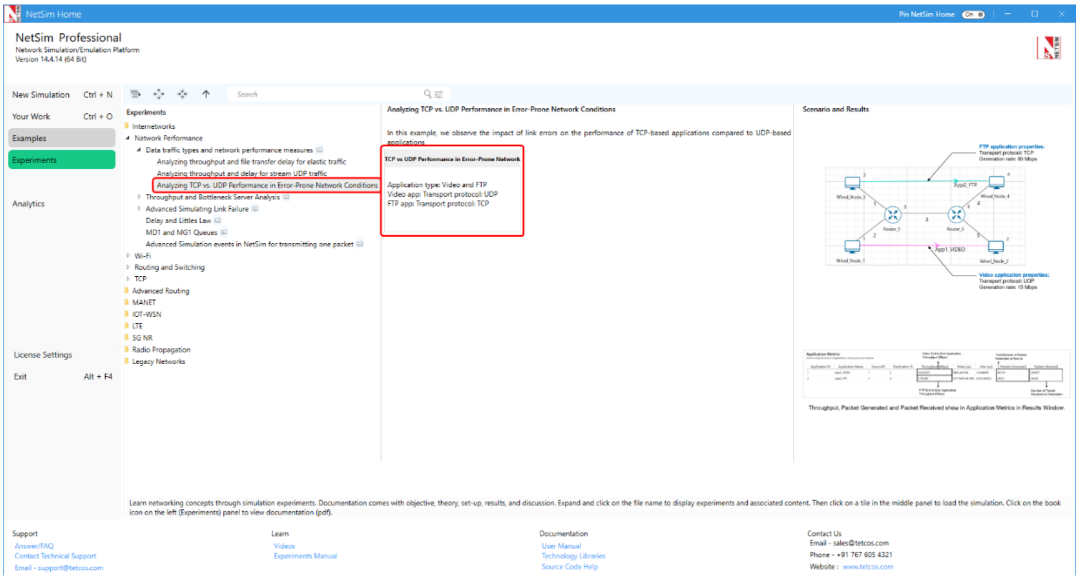


Figure 2-16: List of scenarios for the example of Data traffic types and network performance measures.

Network Scenario

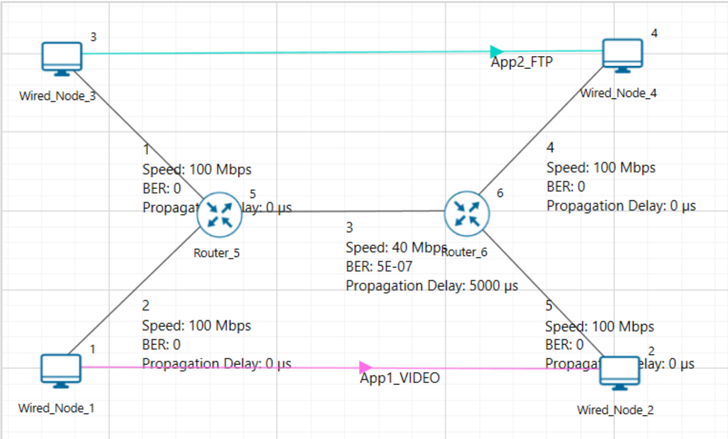


Figure 2-17: Network scenario with WAN link speed of 40 Mbps.

Network settings:

1. Create a Scenario with 4 wired nodes and 2 routers connected as shown above Figure 2-17.
2. Set the Wired Link properties, for Link ID 1,2,4,5
   a. Uplink/Downlink speed to 100 Mbps
   b. Uplink/Downlink BER to 0
   c. Propagation delay to 0 µs.
3. Set the WAN Link properties as below, that Link ID 3 between router 5 and router 6
   a. Uplink/Downlink speed to 40 Mbps
   b. Uplink/Downlink BER to 0.0000005
   c. Propagation delay to 5000 µs.
4. Create two traffic flows, one running TCP and another running UDP as shown below.
   a. VIDEO application

Parameter name	Value
Application Type	Custom
Source Id	1
Destination Id	2
Transport protocol	UDP
Packet size (B)	1000
Inter Arrival Time (µs)	533.3333
Generation rate (Mbps)	15

Table 2-4: VIDEO Application properties.

   b. File Transfer Application

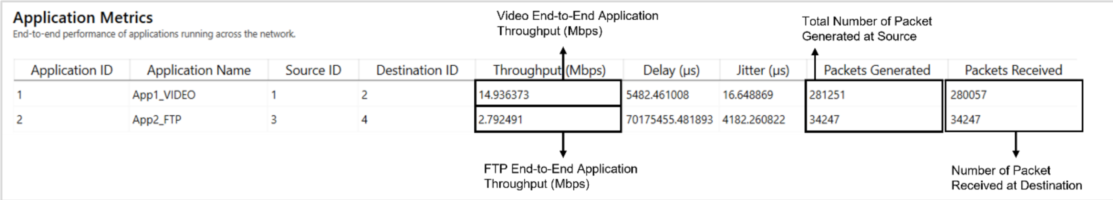
Parameter Name	Value
Application Type	FTP
Source Id	3
Destination Id	4
End time (s)	1
Transport protocol	TCP
File size (B)	50000000
Inter Arrival time (s)	5
Generation rate (Mbps)	80

Table 2-5: FTP Application Properties.

5. Go to Configure Reports Tab > Enable Packet Trace.
6. Run the Simulation for 150 seconds.

2.1.3.3.1 Results and observations

Application Metrics

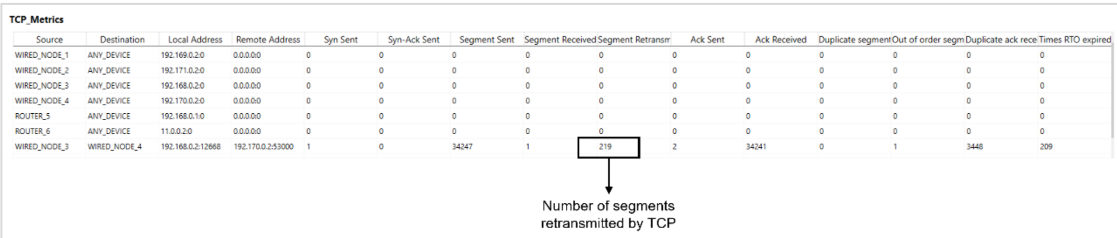


Application Metrics
End-to-end performance of applications running across the network.

Application ID	Application Name	Source ID	Destination ID	Throughput (Mbps)	Delay (µs)	Jitter (µs)	Packets Generated	Packets Received
1	App1_VIDEO	1	2	14.936373	5482.461008	16.648869	281251	280057
2	App2_FTP	3	4	2.792491	70175455.481893	4182.260822	34247	34247

Figure 2-18: Throughput, packet generated and packet received show in Application metrics in results window.

- UDP achieves a throughput of 14.96 Mbps, while TCP achieves a throughput of 2.67 Mbps.
- UDP achieves a higher throughput than TCP since UDP doesn't need to wait for acknowledgements in the reverse direction.
- However, in UDP several packets $(281{,}251 - 280{,}057 = 1{,}194)$ are lost due to error, while TCP all packets are received since TCP retransmits packets errored. The table below shows the number of errored packets that were retransmitted by TCP.



TCP_Metrics

Source	Destination	Local Address	Remote Address	Syn Sent	Syn-Ack Sent	Segment Sent	Segment Received	Segment Retransm	Ack Sent	Ack Received	Duplicate segment	Out of order segm	Duplicate ack rece	Times RTO expired
WIRED_NODE_1	ANY_DEVICE	192.169.0.2:0	0.0.0.0:0	0	0	0	0	0	0	0	0	0	0	0
WIRED_NODE_2	ANY_DEVICE	192.171.0.2:0	0.0.0.0:0	0	0	0	0	0	0	0	0	0	0	0
WIRED_NODE_3	ANY_DEVICE	192.168.0.2:0	0.0.0.0:0	0	0	0	0	0	0	0	0	0	0	0
WIRED_NODE_4	ANY_DEVICE	192.170.0.2:0	0.0.0.0:0	0	0	0	0	0	0	0	0	0	0	0
ROUTER_5	ANY_DEVICE	192.168.0.1:0	0.0.0.0:0	0	0	0	0	0	0	0	0	0	0	0
ROUTER_6	ANY_DEVICE	11.0.0.2:0	0.0.0.0:0	0	0	0	0	0	0	0	0	0	0	0
WIRED_NODE_3	WIRED_NODE_4	192.168.0.2:12668	192.170.0.2:53000	1	0	34247	1	219	2	34241	0	1	3448	209

Figure 2-19: TCP Metrics from Result dashboard > Additional metrics.

2.1.4 Appendix: Obtaining delay metrics in NetSim

The Avg end-to-end packet delay can be got directly from the delay column in the application metrics of NetSim Results dashboard. Queuing delay, transmission time and propagation delay can be calculated from the packet trace which is explained in Section 2.1.4.1.

In the packet trace:

- The average of difference between the PHY LAYER ARRIVAL TIME (μs) and the NW LAYER ARRIVAL TIME (μs) will give us the average queuing delay of packets in a link.

$$Avg. Queuing\ delay\ (\mu s)$$
$$= Avg.\ \big(PHY\ LAYER\ ARRIVAL\ TIME\ (\mu s) - NW\ LAYER\ ARRIVAL\ TIME\ (\mu s)\big)$$

- The average of difference between the PHY LAYER START TIME (μs) and the PHY LAYER ARRIVAL TIME (μs) will give us the average transmission time of packets in a link.

$$Avg.Transmission\,Time\ (\mu s) = Avg.\big(PHY\ LAYER\ START\ TIME\ (\mu s) - PHY\ LAYER\ ARRIVAL\ TIME\ (\mu s)\big)$$

- The average of difference between the PHY LAYER END TIME (μs) and the PHY LAYER START TIME (μs) will give us the average propagation delay of packets in a link.

$$Avg.Propagation\ delay\ (\mu s) = Avg.\ \big(PHY\ LAYER\ END\ TIME\ (\mu s) - PHY\ LAYER\ START\ TIME\ (\mu s)\big)$$

2.1.4.1 The procedure for calculating $Avg.\,Queuing\ delay\ (\mu s)$

Consider case 2(b),

- To compute average queuing delay, we have to take the avg. queuing delay in each link (2, 3, 5) and take the sum of the delays.
- Open Packet Trace file using the **Open packet trace** option available in the Simulation Results window.
- In the Packet Trace, filter the data packets using the column **Control Packet Type/App Name** to option **App1 Video** (see Figure 2-43).

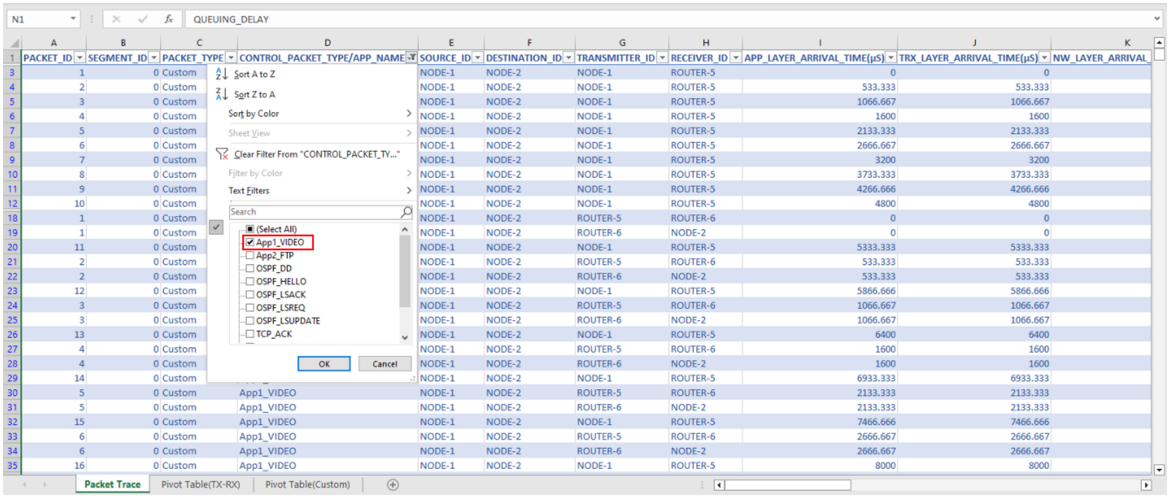


Figure 2-20: Filter the data packets in Packet Trace by selecting App1 video.

- Now, to compute the average queue in Link 3, we will select **Transmitter Id** as **Router-5** and **Receiver Id** as **Router-6**. This filters all the successful packets from Router 5 to Router 6.
- The columns **Network Layer Arrival Time(µS)** and **Phy Layer Arrival Time(µS)** correspond to the arrival time and departure time of the packets in the buffer at Link 3, respectively (see Figure 2-44).

	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)	TRX_LAYER_ARRIVAL_TIME(µS)	NW_LAYER_ARRIVAL_TIME(µS)	MAC_LAYER_ARRIVAL_TIME(µS)	PHY_LAYER_ARRIVAL_TIME(µS)	PHY_LAYER_START_TIME(µS)	PHY_LA
18	ROUTER-5	ROUTER-6	0	0	85.28	85.28	85.28	290.88	
21	ROUTER-5	ROUTER-6	533.333	533.333	617.653	617.653	617.653	823.253	
24	ROUTER-5	ROUTER-6	1066.667	1066.667	1150.987	1150.987	1150.987	1356.587	
27	ROUTER-5	ROUTER-6	1600	1600	1684.32	1684.32	1684.32	1889.92	
30	ROUTER-5	ROUTER-6	2133.333	2133.333	2217.653	2217.653	2217.653	2423.253	
33	ROUTER-5	ROUTER-6	2666.667	2666.667	2750.987	2750.987	2750.987	2956.587	
36	ROUTER-5	ROUTER-6	3200	3200	3284.32	3284.32	3284.32	3489.92	
39	ROUTER-5	ROUTER-6	3733.333	3733.333	3817.653	3817.653	3817.653	4023.253	
42	ROUTER-5	ROUTER-6	4266.666	4266.666	4350.986	4350.986	4350.986	4556.586	
48	ROUTER-5	ROUTER-6	4800	4800	4884.32	4884.32	4884.32	5089.92	
54	ROUTER-5	ROUTER-6	5333.333	5333.333	5417.653	5417.653	5417.653	5623.253	
57	ROUTER-5	ROUTER-6	5866.666	5866.666	5950.986	5950.986	5950.986	6156.586	
60	ROUTER-5	ROUTER-6	6400	6400	6484.32	6484.32	6484.32	6689.92	
63	ROUTER-5	ROUTER-6	6933.333	6933.333	7017.653	7017.653	7017.653	7223.253	
66	ROUTER-5	ROUTER-6	7466.666	7466.666	7550.986	7550.986	7550.986	7756.586	
69	ROUTER-5	ROUTER-6	8000	8000	8084.32	8084.32	8084.32	8289.92	
72	ROUTER-5	ROUTER-6	8533.333	8533.333	8617.653	8617.653	8617.653	8823.253	
75	ROUTER-5	ROUTER-6	9066.666	9066.666	9150.986	9150.986	9150.986	9356.586	
78	ROUTER-5	ROUTER-6	9599.999	9599.999	9684.319	9684.319	9684.319	9889.919	
87	ROUTER-5	ROUTER-6	10133.333	10133.333	10217.653	10475.52	10475.52	10681.12	
96	ROUTER-5	ROUTER-6	10666.666	10666.666	10750.986	11281.12	11281.12	11486.72	
99	ROUTER-5	ROUTER-6	11199.999	11199.999	11284.319	11486.72	11486.72	11692.32	
101	ROUTER-5	ROUTER-6	11733.333	11733.333	11817.653	11817.653	11817.653	12023.253	
104	ROUTER-5	ROUTER-6	12266.666	12266.666	12350.986	12350.986	12350.986	12556.586	
107	ROUTER-5	ROUTER-6	12799.999	12799.999	12884.319	12884.319	12884.319	13089.919	
110	ROUTER-5	ROUTER-6	13333.333	13333.333	13417.653	13417.653	13417.653	13623.253	
113	ROUTER-5	ROUTER-6	13866.666	13866.666	13950.986	13950.986	13950.986	14156.586	
116	ROUTER-5	ROUTER-6	14399.999	14399.999	14484.319	14484.319	14484.319	14689.919	
119	ROUTER-5	ROUTER-6	14933.332	14933.332	15017.652	15017.652	15017.652	15223.252	

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

Figure 2-21: Packet arrival and departure times in the link buffer

- The difference between the Phy Layer Arrival Time(µS) and the Network Layer Arrival Time (µS) will give us the delay of a packet in the link (see Figure 2-45).

$$\text{Queuing Delay} = \text{ Phy layer arrival time}(\mu S) - \text{ NW layer arrival time}(\mu S)$$

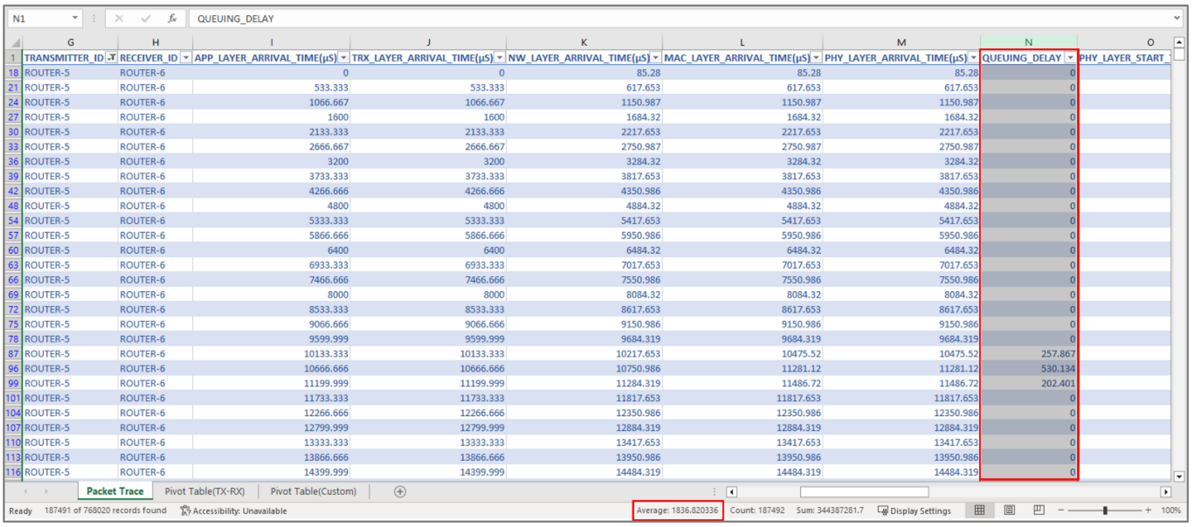
N18 =[@[PHY_LAYER_ARRIVAL_TIME(µS)]]-[@[NW_LAYER_ARRIVAL_TIME(µS)]]

	G	H	I	J	K	L	M	N	O
1	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)	TRX_LAYER_ARRIVAL_TIME(µS)	NW_LAYER_ARRIVAL_TIME(µS)	MAC_LAYER_ARRIVAL_TIME(µS)	PHY_LAYER_ARRIVAL_TIME(µS)	QUEUING_DELAY	PHY_LAYER_START_
18	ROUTER-5	ROUTER-6	0	0	85.28	85.28	85.28	0	
21	ROUTER-5	ROUTER-6	533.333	533.333	617.653	617.653	617.653	0	
24	ROUTER-5	ROUTER-6	1066.667	1066.667	1150.987	1150.987	1150.987	0	
27	ROUTER-5	ROUTER-6	1600	1600	1684.32	1684.32	1684.32	0	
30	ROUTER-5	ROUTER-6	2133.333	2133.333	2217.653	2217.653	2217.653	0	
33	ROUTER-5	ROUTER-6	2666.667	2666.667	2750.987	2750.987	2750.987	0	
36	ROUTER-5	ROUTER-6	3200	3200	3284.32	3284.32	3284.32	0	
39	ROUTER-5	ROUTER-6	3733.333	3733.333	3817.653	3817.653	3817.653	0	
42	ROUTER-5	ROUTER-6	4266.666	4266.666	4350.986	4350.986	4350.986	0	
48	ROUTER-5	ROUTER-6	4800	4800	4884.32	4884.32	4884.32	0	
54	ROUTER-5	ROUTER-6	5333.333	5333.333	5417.653	5417.653	5417.653	0	
57	ROUTER-5	ROUTER-6	5866.666	5866.666	5950.986	5950.986	5950.986	0	
60	ROUTER-5	ROUTER-6	6400	6400	6484.32	6484.32	6484.32	0	
63	ROUTER-5	ROUTER-6	6933.333	6933.333	7017.653	7017.653	7017.653	0	
66	ROUTER-5	ROUTER-6	7466.666	7466.666	7550.986	7550.986	7550.986	0	
69	ROUTER-5	ROUTER-6	8000	8000	8084.32	8084.32	8084.32	0	
72	ROUTER-5	ROUTER-6	8533.333	8533.333	8617.653	8617.653	8617.653	0	
75	ROUTER-5	ROUTER-6	9066.666	9066.666	9150.986	9150.986	9150.986	0	
78	ROUTER-5	ROUTER-6	9599.999	9599.999	9684.319	9684.319	9684.319	0	
87	ROUTER-5	ROUTER-6	10133.333	10133.333	10217.653	10475.52	10475.52	257.867	
96	ROUTER-5	ROUTER-6	10666.666	10666.666	10750.986	11281.12	11281.12	530.134	
99	ROUTER-5	ROUTER-6	11199.999	11199.999	11284.319	11486.72	11486.72	202.401	
101	ROUTER-5	ROUTER-6	11733.333	11733.333	11817.653	11817.653	11817.653	0	
104	ROUTER-5	ROUTER-6	12266.666	12266.666	12350.986	12350.986	12350.986	0	
107	ROUTER-5	ROUTER-6	12799.999	12799.999	12884.319	12884.319	12884.319	0	
110	ROUTER-5	ROUTER-6	13333.333	13333.333	13417.653	13417.653	13417.653	0	
113	ROUTER-5	ROUTER-6	13866.666	13866.666	13950.986	13950.986	13950.986	0	
116	ROUTER-5	ROUTER-6	14399.999	14399.999	14484.319	14484.319	14484.319	0	

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

Figure 2-22: Queuing Delay

- Now, calculate the average queuing delay by taking the mean of the queueing delay of all the packets (see Figure 2-45)



	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)	TRX_LAYER_ARRIVAL_TIME(µS)	NW_LAYER_ARRIVAL_TIME(µS)	MAC_LAYER_ARRIVAL_TIME(µS)	PHY_LAYER_ARRIVAL_TIME(µS)	QUEUING_DELAY	PHY_LAYER_START_
18	ROUTER-5	ROUTER-6	0	0	85.28	85.28	85.28	0	
21	ROUTER-5	ROUTER-6	533.333	533.333	617.653	617.653	617.653	0	
24	ROUTER-5	ROUTER-6	1066.667	1066.667	1150.987	1150.987	1150.987	0	
27	ROUTER-5	ROUTER-6	1600	1600	1684.32	1684.32	1684.32	0	
30	ROUTER-5	ROUTER-6	2133.333	2133.333	2217.653	2217.653	2217.653	0	
33	ROUTER-5	ROUTER-6	2666.667	2666.667	2750.987	2750.987	2750.987	0	
36	ROUTER-5	ROUTER-6	3200	3200	3284.32	3284.32	3284.32	0	
39	ROUTER-5	ROUTER-6	3733.333	3733.333	3817.653	3817.653	3817.653	0	
42	ROUTER-5	ROUTER-6	4266.666	4266.666	4350.986	4350.986	4350.986	0	
48	ROUTER-5	ROUTER-6	4800	4800	4884.32	4884.32	4884.32	0	
54	ROUTER-5	ROUTER-6	5333.333	5333.333	5417.653	5417.653	5417.653	0	
57	ROUTER-5	ROUTER-6	5866.666	5866.666	5950.986	5950.986	5950.986	0	
60	ROUTER-5	ROUTER-6	6400	6400	6484.32	6484.32	6484.32	0	
63	ROUTER-5	ROUTER-6	6933.333	6933.333	7017.653	7017.653	7017.653	0	
66	ROUTER-5	ROUTER-6	7466.666	7466.666	7550.986	7550.986	7550.986	0	
69	ROUTER-5	ROUTER-6	8000	8000	8084.32	8084.32	8084.32	0	
72	ROUTER-5	ROUTER-6	8533.333	8533.333	8617.653	8617.653	8617.653	0	
75	ROUTER-5	ROUTER-6	9066.666	9066.666	9150.986	9150.986	9150.986	0	
78	ROUTER-5	ROUTER-6	9599.999	9599.999	9684.319	9684.319	9684.319	0	
87	ROUTER-5	ROUTER-6	10133.333	10133.333	10217.653	10475.52	10475.52	257.867	
96	ROUTER-5	ROUTER-6	10666.666	10666.666	10750.986	11281.12	11281.12	530.134	
99	ROUTER-5	ROUTER-6	11199.999	11199.999	11284.319	11486.72	11486.72	202.401	
101	ROUTER-5	ROUTER-6	11733.333	11733.333	11817.653	11817.653	11817.653	0	
104	ROUTER-5	ROUTER-6	12266.666	12266.666	12350.986	12350.986	12350.986	0	
107	ROUTER-5	ROUTER-6	12799.999	12799.999	12884.319	12884.319	12884.319	0	
110	ROUTER-5	ROUTER-6	13333.333	13333.333	13417.653	13417.653	13417.653	0	
113	ROUTER-5	ROUTER-6	13866.666	13866.666	13950.986	13950.986	13950.986	0	
116	ROUTER-5	ROUTER-6	14399.999	14399.999	14484.319	14484.319	14484.319	0	

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

Ready 187491 of 768020 records found | Average: 1836.820336 | Count: 187492 | Sum: 344387281.7

Figure 2-23: Average queuing delay

The average queuing delay obtained in the packet trace is 1836.82 μs which is same as that as mentioned in Table 2-3. Similarly, after computing queuing delay for link 2 and link 5 we get avg. queuing delay = 0. Hence the average queuing delay in the network is **1836.82 µs**.

Similar steps can be followed to obtain $Avg.Transmission\ Time\ (\mu s)$ and $Avg.Propagation\ Delay\ (\mu s)$.

2.1.4.2 The procedure for calculating $File\ Transfer\ Time(s)$

- To calculate the File Transfer Time, filter the data packets using the column **Control Packet Type/App Name** to option **App2 FTP**

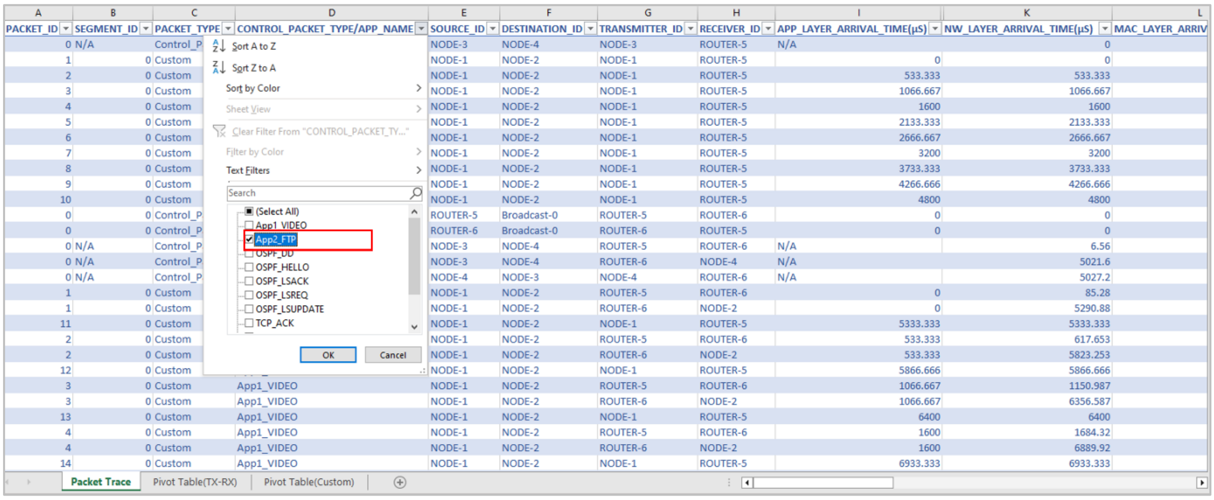


Figure 2-24: Filter the data packets in Packet Trace by selecting App2_FTP.

- Calculate the difference between the last value of **Phy layer end time** and first value of **App layer arrival time.**

$$FTP\ file\ transfer\ Time\ (s)\ = (Phy\ layer\ end\ time\ (\mu s) - App\ layer\ arrival\ time\ (\mu s))$$

$$= 16742892.32\ \mu s - 0\mu s = 16.74s$$

TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)	NW_LAYER_ARRIVAL_TIME(µS)	MAC_LAYER_ARRIVAL_TIME(µS)	PHY_LAYER_ARRIVAL_TIME(µS)	PHY_LAYER_START_TIME(µS)	PHY_LAYER_END_TIME(µS)
ROUTER-5	ROUTER-6	0	16719639.84	16730719.52	16730719.52	16731019.52	16736019.52
ROUTER-6	NODE-4	0	16736019.52	16736019.52	16736019.52	16736141.6	16736141.6
ROUTER-5	ROUTER-6	0	16720145.44	16731225.12	16731225.12	16731525.12	16736525.12
ROUTER-6	NODE-4	0	16736525.12	16736525.12	16736525.12	16736647.2	16736647.2
ROUTER-5	ROUTER-6	0	16720651.04	16731730.72	16731730.72	16732030.72	16737030.72
ROUTER-6	NODE-4	0	16737030.72	16737030.72	16737030.72	16737152.8	16737152.8
ROUTER-5	ROUTER-6	0	16721156.64	16732236.32	16732236.32	16732536.32	16737536.32
ROUTER-6	NODE-4	0	16737536.32	16737536.32	16737536.32	16737658.4	16737658.4
ROUTER-5	ROUTER-6	0	16721662.24	16732536.32	16732536.32	16732836.32	16737836.32
ROUTER-6	NODE-4	0	16737836.32	16737836.32	16737836.32	16737958.4	16737958.4
ROUTER-5	ROUTER-6	0	16722167.84	16733041.92	16733041.92	16733341.92	16738341.92
ROUTER-6	NODE-4	0	16738341.92	16738341.92	16738341.92	16738464	16738464
ROUTER-5	ROUTER-6	0	16722673.44	16733547.52	16733547.52	16733847.52	16738847.52
ROUTER-6	NODE-4	0	16738847.52	16738847.52	16738847.52	16738969.6	16738969.6
ROUTER-5	ROUTER-6	0	16723179.04	16734053.12	16734053.12	16734353.12	16739353.12
ROUTER-6	NODE-4	0	16739353.12	16739353.12	16739353.12	16739475.2	16739475.2
ROUTER-5	ROUTER-6	0	16723684.64	16734558.72	16734558.72	16734858.72	16739858.72
ROUTER-6	NODE-4	0	16739858.72	16739858.72	16739858.72	16739980.8	16739980.8
ROUTER-5	ROUTER-6	0	16724190.24	16735064.32	16735064.32	16735364.32	16740364.32
ROUTER-6	NODE-4	0	16740364.32	16740364.32	16740364.32	16740486.4	16740486.4
ROUTER-5	ROUTER-6	0	16724695.84	16735569.92	16735569.92	16735869.92	16740869.92
ROUTER-6	NODE-4	0	16740869.92	16740869.92	16740869.92	16740992	16740992
ROUTER-5	ROUTER-6	0	16725201.44	16736075.52	16736075.52	16736375.52	16741375.52
ROUTER-6	NODE-4	0	16741375.52	16741375.52	16741375.52	16741497.6	16741497.6
ROUTER-5	ROUTER-6	0	16725501.44	16736581.12	16736581.12	16736881.12	16741881.12
ROUTER-6	NODE-4	0	16741881.12	16741881.12	16741881.12	16742003.2	16742003.2
ROUTER-5	ROUTER-6	0	16726007.04	16737086.72	16737086.72	16737386.72	16742386.72
ROUTER-6	NODE-4	0	16742386.72	16742386.72	16742386.72	16742508.8	16742508.8
ROUTER-5	ROUTER-6	0	16726512.64	16737592.32	16737592.32	16737892.32	16742892.32

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

Figure 2-25: File Transfer Time

2.1.5 Exercises

1. Redo the experiment by using the following inputs.

Part-A:

a. Consider the application layer file size of 50 MB and modify the link speed as 120Mbps. Derive the theoretical File transfer time and compare against NetSim result.

b. For variations, consider different combinations of file sizes, such as 20 MB, 30MB, 40MB, 60MB, 70MB etc., and different link speeds such as 10 Mbps, 50 Mbps, 200 Mbps etc. Derive the theoretical File transfer time for each case and compare against NetSim result.

Part-B:

a. Redo the experiment mentioned in part-B with customized video traffic generation rate set to 20 Mbps with packet size = 1000B, IAT = 400µs. Derive the theoretical File transfer time for each case and compare against NetSim result obtained using the packet trace.

b. Redo the experiment mentioned in part-B and generate FTP traffic with 10 MB file size (application settings: File size = 10000000B, IAT = 5s) and a link speed of 50 Mbps. Derive the theoretical File transfer time for each case and compare against NetSim result obtained using the packet trace.

2.2 Simulating Link Failure (Level 1)

2.2.1 Objective

To model link failure, understand its impact on network performance

2.2.2 Theory

A link failure can occur due to a) faults in the physical link and b) failure of the connected port. When a link fails, packets cannot be transported. This also means that established routes to destinations may become unavailable. In such cases, the routing protocol must recompute an alternate path around the failure.

In NetSim, only WAN links (connecting two routers) can be failed. Click on a WAN link between two routers and the Link Properties Window is as shown below Figure 2-26.

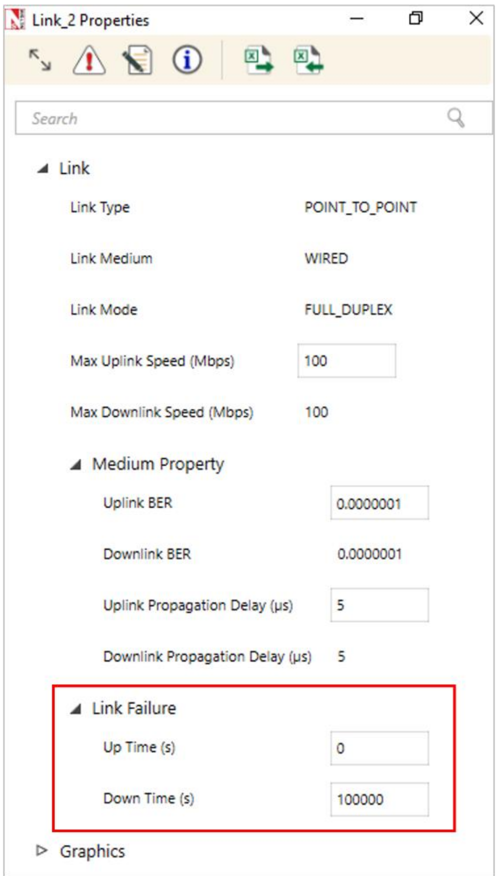


Figure 2-26: Wired Link Properties Window

Link up Time refers to the time(s) at which the link is functional and Link down time refers to the time (s) at which a link fails. Click on up time or down time to understand the configuration options.

NOTE: Link failure can be set only for “WAN Interfaces”.

2.2.3 Network Setup

Open NetSim and click on **Experiments> Internetworks> Network Performance> Advanced Simulating Link Failure** then click on the tile in the middle panel to load the example as shown in below Figure 2-27.

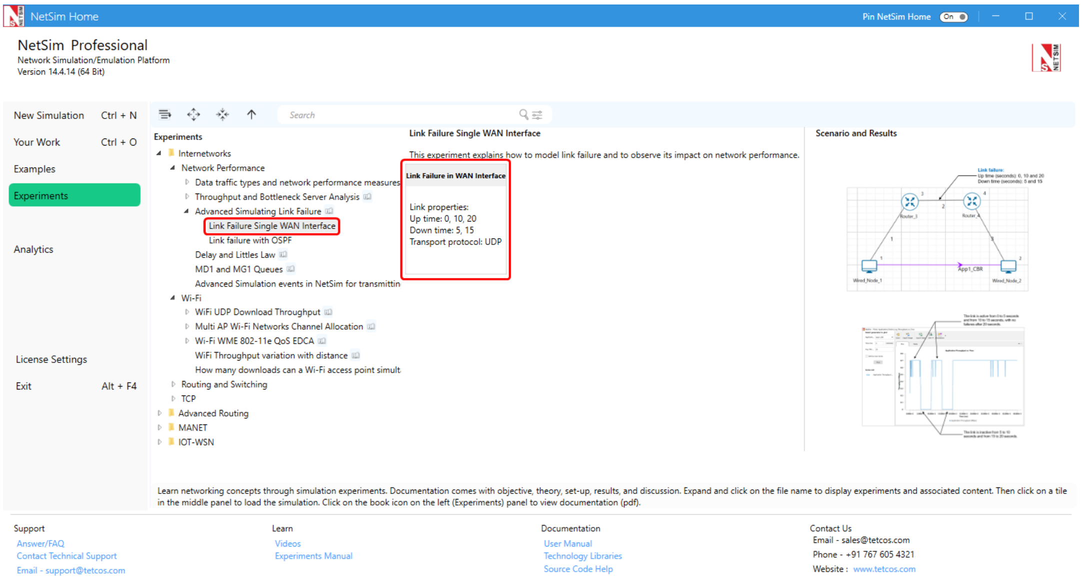


Figure 2-27: List of scenarios for the example of Advanced Simulating Link Failure

2.2.3.1 Link Failure Single WAN Interface

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 2-28.

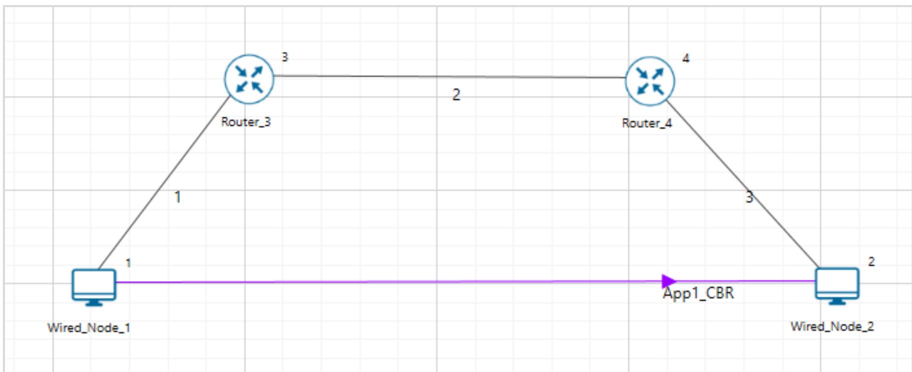


Figure 2-28: Network set up for studying the Link Failure Single WAN Interface

2.2.3.1.1 Procedure

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

Step 1: In the **"Internetworks"** library, and a network scenario is designed in NetSim comprising of 2 Wired Nodes and 2 Routers.

Step 2: By default, Link Failure **Up time** is set to 0,10,20 and **Down time** is set to 5,15. This means the link is up 0-5s, 10-15s and 20s onwards, and it is down 5-10s and 15-20s. This can set by clicking on WAN link between routers (link 2) and expanding the link property panel on right and changing the link up and link down as shown below.

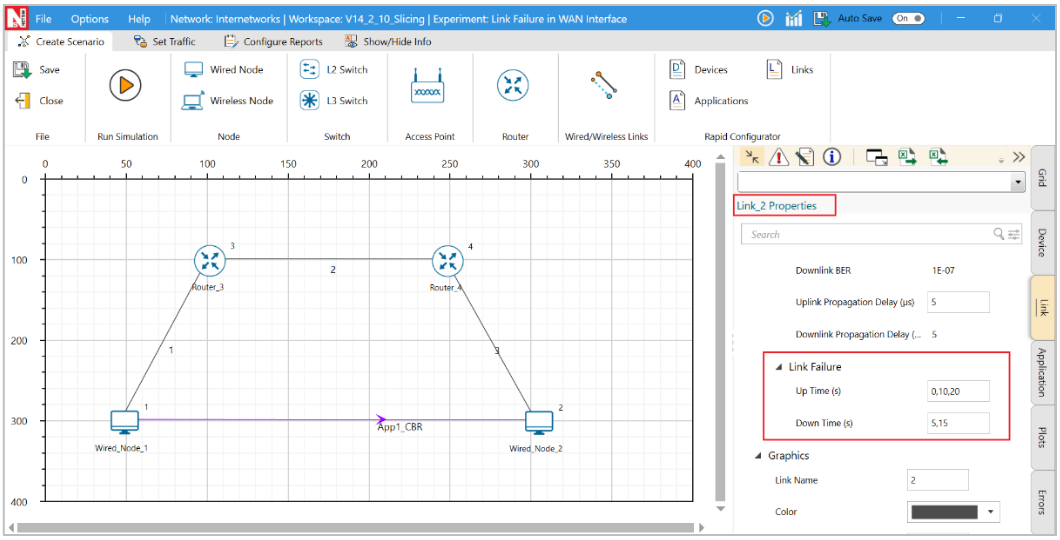


Figure 2-29: Link failure setting in WAN link

Step 3: Packet trace is enabled from Configure reports tab in the ribbon on the top. At the end of the simulation, a .csv file containing all the packet information is available for performing packet level analysis.

Step 4: Configure a CBR application between wired node 1 to wired node 2 by clicking on set traffic tab in the ribbon on the top. Click on the created application, expand the application property panel on the right, and set the transport layer to UDP while keeping the other properties as default.

Step 5: Enable the Link Throughput vs. Time plot by clicking on Configure reports and plots, then run the simulation for 50 seconds.

2.2.3.1.2 Output

Open the Link Throughput plot from the simulation results window, filter the link id to 2 and disable the accelerate plotting and change the average window size to 50ms, we can notice the following.

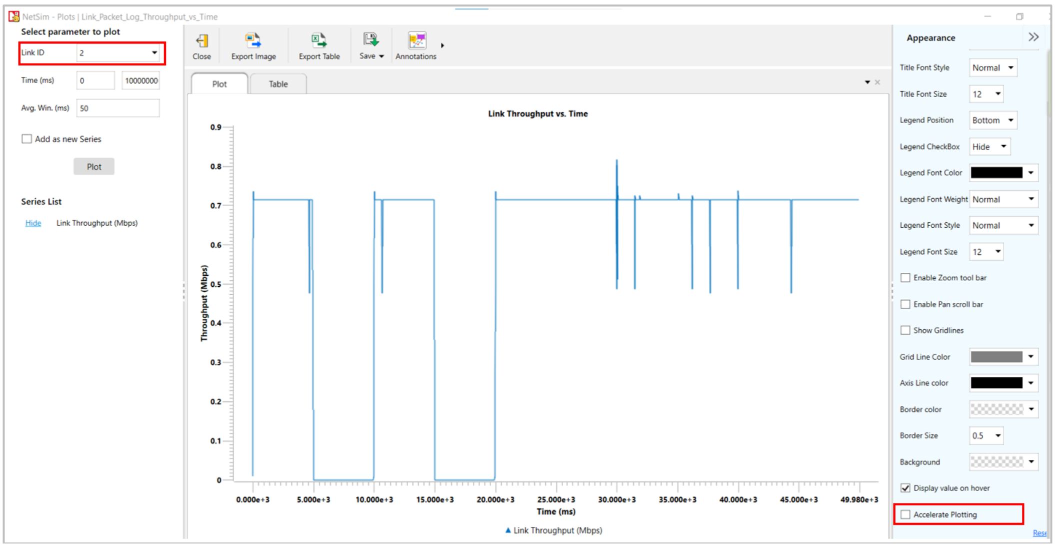
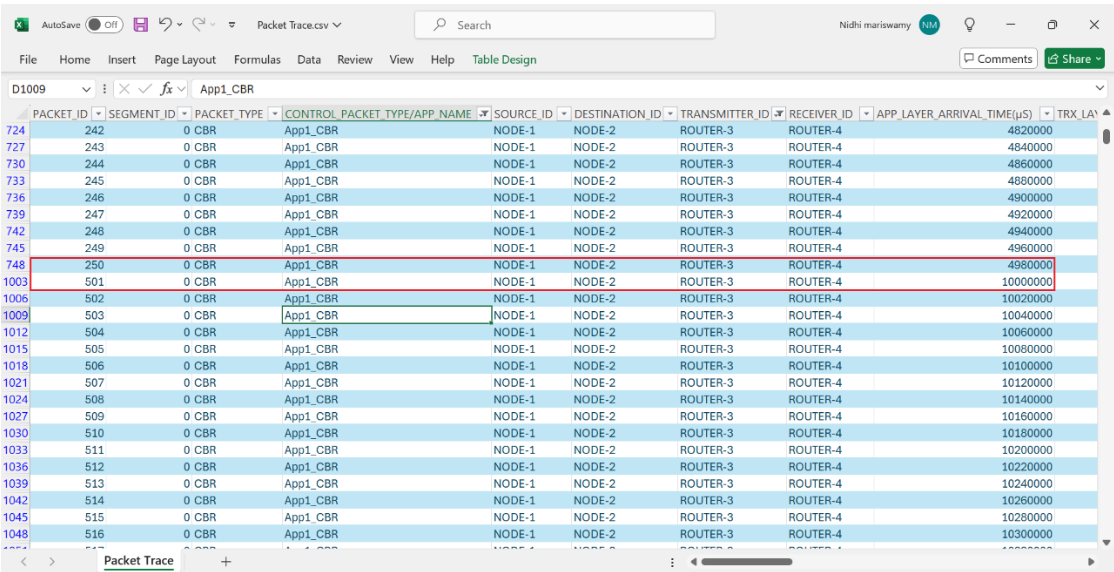


Figure 2-30: Link Throughput vs Time plot for Link 2.

1. The application starts at the 0th second between Router 3 and Router 4 and is initially active from 0 to 5 seconds. A throughput of 0.71 Mbps is observed during the intervals of 0-5 seconds, 10-15 seconds, and from 20 seconds onward.
2. The link fails in the intervals 5-10s and 15-20s. The throughput drops to 0 Mbps in these intervals.

Similarly, the same can be observed in the packet trace by filtering Control packet type/App name to App1 CBR, Transmitter Id to router 3, and Receiver Id to router 4. By observing the arrival times, you'll notice that no data transmission occurs during the link failure period.



	PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)
724	242	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	4820000
727	243	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	4840000
730	244	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	4860000
733	245	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	4880000
736	246	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	4900000
739	247	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	4920000
742	248	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	4940000
745	249	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	4960000
748	250	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	4980000
1003	501	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10000000
1006	502	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10020000
1009	503	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10040000
1012	504	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10060000
1015	505	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10080000
1018	506	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10100000
1021	507	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10120000
1024	508	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10140000
1027	509	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10160000
1030	510	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10180000
1033	511	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10200000
1036	512	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10220000
1039	513	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10240000
1042	514	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10260000
1045	515	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10280000
1048	516	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4	10300000

Figure 2-31: NetSim packet trace showing the link failure period

2.2.3.2 Link Failure with OSPF

NetSim UI displays the configuration file corresponding to this experiment as shown in Figure 2-32.

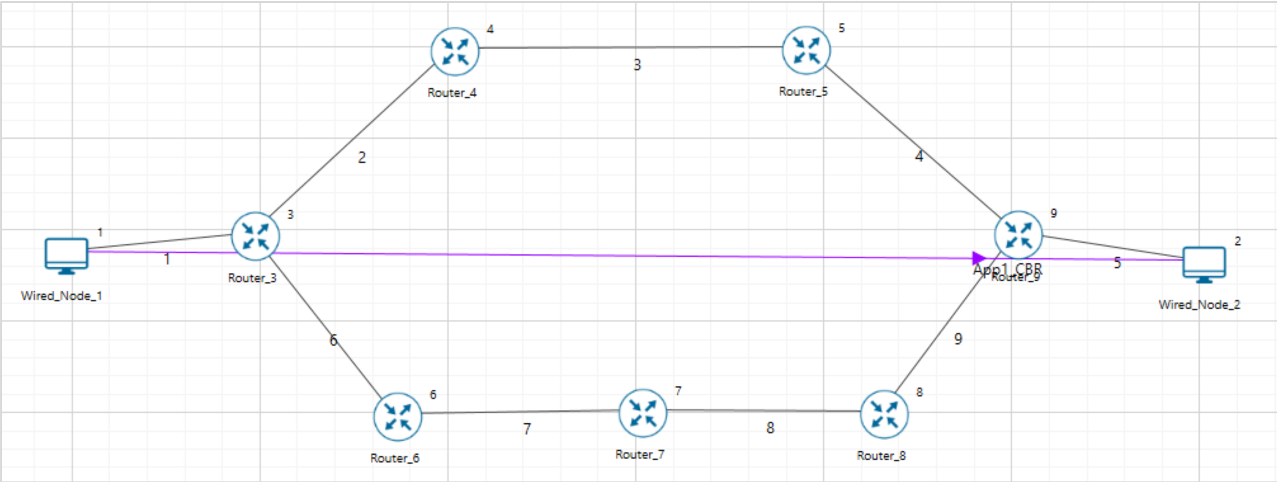


Figure 2-32: Network set up for studying the Link Failure with OSPF

2.2.3.2.1 Procedure

Without link failure: The following set of procedures were done to generate this sample:

Step 1: In the **"Internetworks"** library, and a network scenario is designed in NetSim comprising of 2 Wired Nodes and 7 Routers.

Step 2: By default, Link Failure **Up Time** is set to 0 and **Down Time** is set to 100000 for Link id 3.

Step 3: Packet Trace is enabled from configure reports tab in ribbon on the top. At the end of the simulation, a .csv file containing all the packet information is available for performing packet level analysis.

Step 4: Configure a CBR application between wired node 1 to Wired Node 2 by clicking on set traffic tab in the ribbon on the top. Click on the created application, expand the application property panel on the right, and set the transport layer to TCP.

Additionally, the **"Start Time(s)"** parameter is set to 30 s. This time is usually set to be greater than the time taken for OSPF convergence (i.e., exchange of OSPF information between all the routers), and it increases as the size of the network increases.

Step 5: Enable the Link Throughput vs. Time plot by clicking on Configure reports and Plots and run the simulation for 80 Seconds.

With link failure: The following changes in settings are done from the previous sample:

Step 1: In Link 3 Properties, Link Failure **Up Time** is set to 0 and **Down Time** is set to 50. This means that the link would fail at 50 Seconds.

Step 2: Enable the plots and run the simulation for 80 Seconds.

2.2.3.2.2 Output

Open Packet trace and observe the packet flow.

- Initially OSPF Control Packets are exchanged between all the routers.
- Once after the exchange of control packets, the data packets are sent from the source to the destination.
- The packets are routed to the Destination via, **N1 > R3 > R4 > R5 > R9 > N2** as shown below Figure 2-33.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1	ROUTER-3
1	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4
1	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-4	ROUTER-5
1	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-5	ROUTER-9
1	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-9	NODE-2
2	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1	ROUTER-3
2	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4
2	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-4	ROUTER-5
2	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-5	ROUTER-9
2	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-9	NODE-2
3	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1	ROUTER-3
3	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4
3	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-4	ROUTER-5
3	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-5	ROUTER-9
3	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-9	NODE-2
4	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1	ROUTER-3
4	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4
4	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-4	ROUTER-5
4	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-5	ROUTER-9
4	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-9	NODE-2
5	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1	ROUTER-3
5	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4

Figure 2-33: Packet Trace file for without link failure

With link failure

- We create a Link Failure in Link 3, between Router 4 and Router 5 at 50s.
- Since the packets are not able to reach the destination, the routing protocol recomputes an alternate path to the Destination.
- This can be observed in the Packet Trace.
- Go to the Results Dashboard and click on Packet Trace under traces and do the following:
- Filter Control Packet Type/App Name to APP1 CBR and Transmitter ID to Router 3.
- We can notice that packets are changing its route from, **N1 > R3 > R4 > R5 > R9 > N2** to **N1 > R3 > R6 > R7 > R8 > R9 > N2** at 50 s of simulation time, since the link between R4 and R5 fails at 50 s.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1	ROUTER-3
1	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-4
1	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-4	ROUTER-5
1	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-5	ROUTER-9
1	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-9	NODE-2

Figure 2-34: Packet Trace file for link failure before 50 secs

1001	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1	ROUTER-3		50000000
1001	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-3	ROUTER-6		50000000
1001	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-6	ROUTER-7		50000000
1001	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-7	ROUTER-8		50000000
1001	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-8	ROUTER-9		50000000
1001	0	CBR	App1_CBR	NODE-1	NODE-2	ROUTER-9	NODE-2		50000000

Figure 2-35: Packet Trace file for link failure after 50 secs

2.3 Delay and Little's Law (Level 2)

2.3.1 Introduction

Delay is another important measure of quality of a network, very relevant for real-time applications. The application processes concern over different types of delay - connection establishment delay, session response delay, end-to-end packet delay, jitter, etc. In this experiment, we will review the most basic and fundamental measure of delay, known as end-to-end packet delay in the network. The **end-to-end packet delay** denotes the sojourn time of a packet in the network and is computed as follows. Let a_i and d_i denote the time of arrival of packet i into the network (into the transport layer at the source node) and time of departure of the packet i from the network (from the transport layer at the destination node), respectively. Then, the sojourn time of the packet i is computed as $(d_i - a_i)$ seconds. A useful measure of delay of a flow is the average end-to-end delay of all the packets in the flow, and is computed as

$$\text{average packet delay} = \frac{1}{N}\sum_{i=1}^{N}(d_{i\,-}\, a_i)\ \text{secs}$$

where N is the count of packets in the flow.

A packet may encounter delay at different layers (and nodes) in the network. The transport layer at the end hosts may delay packets to control flow rate and congestion in the network. At the network layer (at the end hosts and at the intermediate routers), the packets may be delayed due to queues in the buffers. In every link (along the route), the packets see channel access delay and switching/forwarding delay at the data link layer, and packet transmission delay and propagation delay at the physical layer. In addition, the packets may encounter processing delay (due to hardware restrictions). It is a common practice to group the various components of the delay under the following four categories: **queueing delay** (caused due to congestion in the network), **transmission delay** (caused due to channel access and transmission over the channel), **propagation delay** and **processing delay**. We will assume zero processing delay and define packet delay as

$$\text{end to end packet delay } = \text{queueing delay} + \text{transmission delay } + \text{ propagation delay}$$

We would like to note that, in many scenarios, the propagation delay and transmission delay are relatively constant in comparison with the queueing delay. This permits us (including applications and algorithms) to use packet delay to estimate congestion (indicated by the queueing delay) in the network.

2.3.1.1 Little's Law

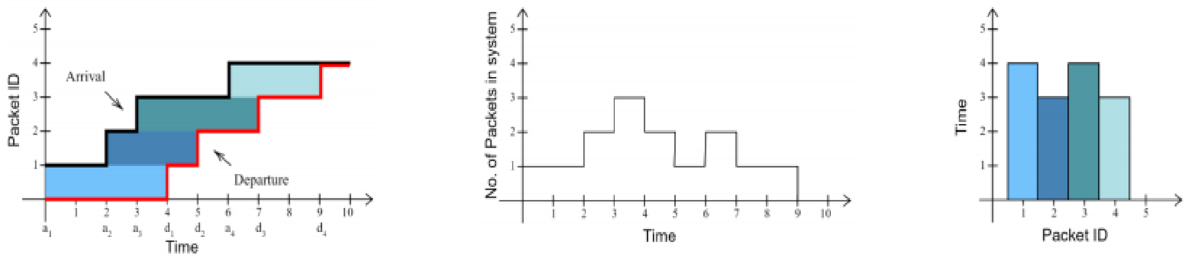
The average end-to-end packet delay in the network is related to the average number of packets in the network. **Little's law** states that the average number of packets in the network is equal to the average arrival rate of packets into the network multiplied by the average end-to-end delay in the network, i.e.,

$$\text{average number of packets in the network}$$
$$= \text{average arrival rate into the network} \times \text{average end to end delay in the network}$$

Likewise, the average queueing delay in a buffer is also related to the average number of packets in the queue via Little's law.

$$\text{average number of packets in queue} = \text{average arrival rate into the queue} \times \text{average delay in the queue}$$

The following figure illustrates the basic idea behind Little's law. In Figure 2-36a, we plot the arrival process $a(t)$ (thick black line) and the departure process $d(t)$ (thick red line) of a queue as a function of time. We have also indicated the time of arrivals (a_i) and time of departures (d_i) of the four packets in Figure 2-36a. In Figure 2-36b, we plot the queue process $q(t) = a(t) - d(t)$ as a function of time, and in Figure 2-36c, we plot the waiting time $(d_{i-}\, a_i)$ of the four packets in the network. From the figures, we can note that the area under the queue process is the same as the sum of the waiting time of the four packets. Now, the average number of packets in the queue ($\frac{14}{10}$) , if we consider a duration of ten seconds for the experiment) is equal to the product of the average arrival rate of packets $\left(\frac{4}{10}\right)$ and the average delay in the queue $\left(\frac{14}{4}\right)$.



(a) Arrival and Departure Processes (b) Queue Process (c) Waiting Time

Figure 2-36: Illustration of Little's law in a queue.

In Experiment 3 (Throughput and Bottleneck Server Analysis), we noted that bottleneck server analysis can provide tremendous insights on the flow and network performance. Using M/G/1 analysis of the bottleneck server and Little's law, we can analyze queueing delay at the bottleneck server and predict end-to-end packet delay as well (assuming constant transmission times and propagation delays).

2.3.2 NetSim Simulation Setup

Open NetSim and click on **Experiments> Internetworks> Network Performance> Delay and Littles Law** then click on the tile in the middle panel to load the example as shown below in Figure 2-37.

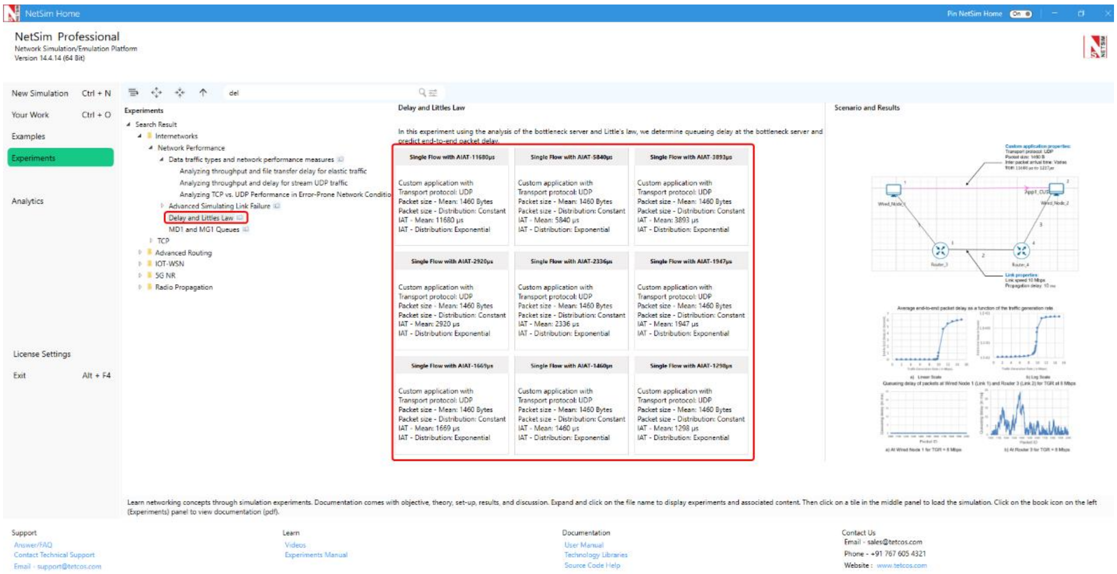


Figure 2-37: List of scenarios for the example of Delay and Littles Law

2.3.3 Part-1: A Single Flow Scenario

We will study a simple network setup with a single flow illustrated in Figure 2-38 to review end to end packet delay in a network as a function of network traffic. An application process at Wired Node 1 seeks to transfer data to an application process at Wired Node 2. We will consider a custom traffic generation process (at the application) that generates data packets of constant length (say, L bits) with IAT 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 communication overheads (including delay at the transport layer) and hence, will use UDP for data transfer between the application processes.

In this setup, we will vary the traffic generation rate $\left(\frac{L}{v}\right)$ by varying the average inter-arrival time (v), and review the average queue at the different links, average queueing delay at the different links and end-to-end packet delay.

2.3.3.1 Procedure

We will simulate the network setup illustrated in Figure 2-38 with the configuration parameters listed in detail in Table 2-6 to study the single flow scenario.

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

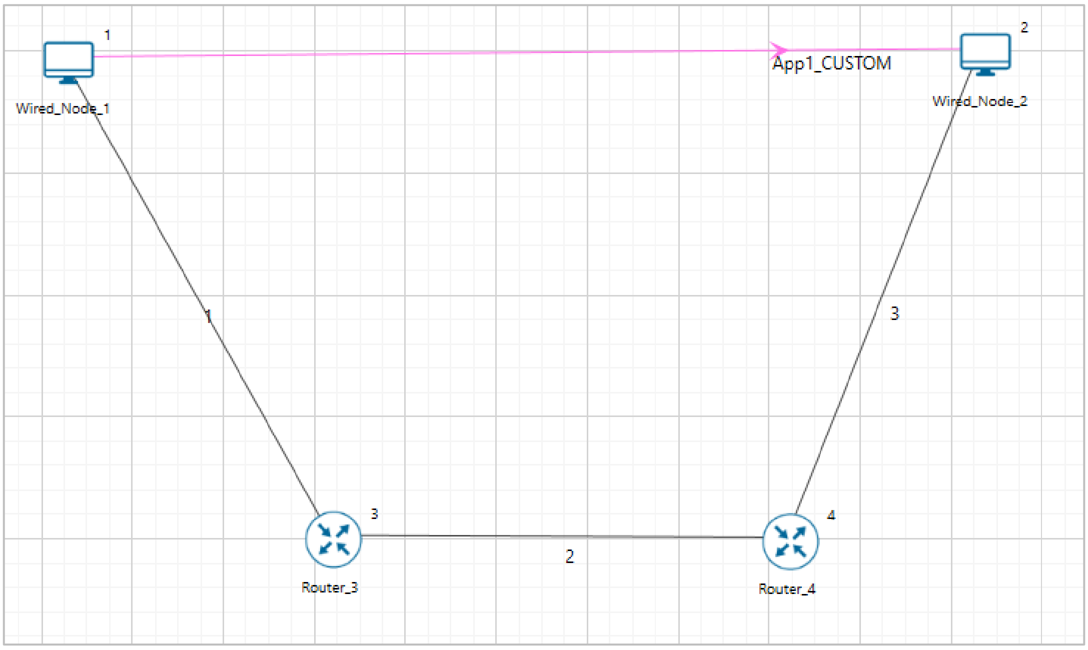


Figure 2-38: Network set up for studying a single flow

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 to wired links as shown in (See Figure 2-38).

Step 2: Configure an application between any two nodes by selecting a **Custom** application from the Set Traffic tab. Right click on the Application Flow **App1 CBR** and select Properties. In the **Application** configuration window (see Figure 2-39), select **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.

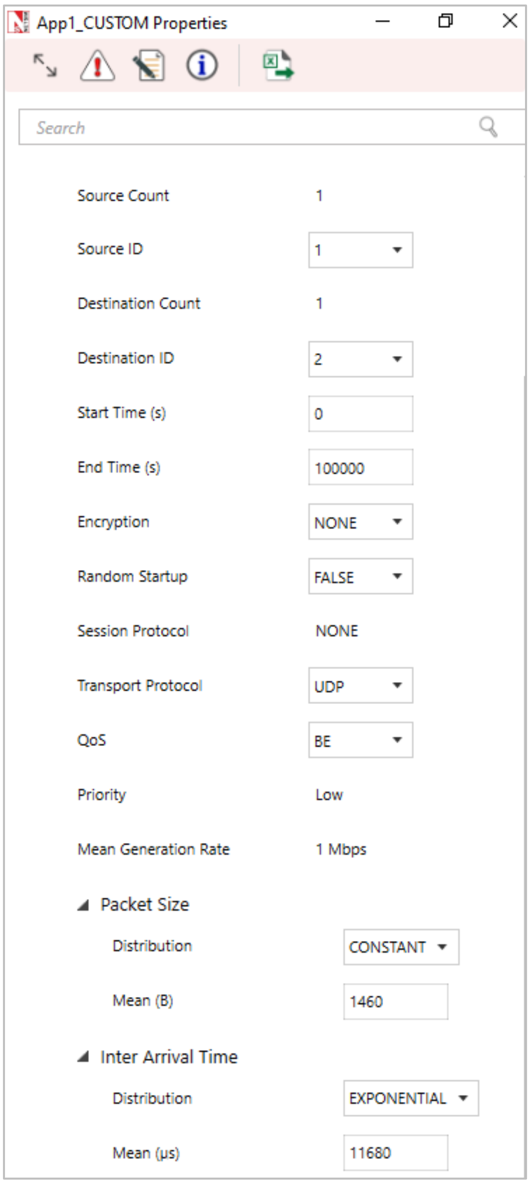


Figure 2-39: Application configuration window

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 window (see Figure 2-40). 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 10 milliseconds for the backbone link 2.

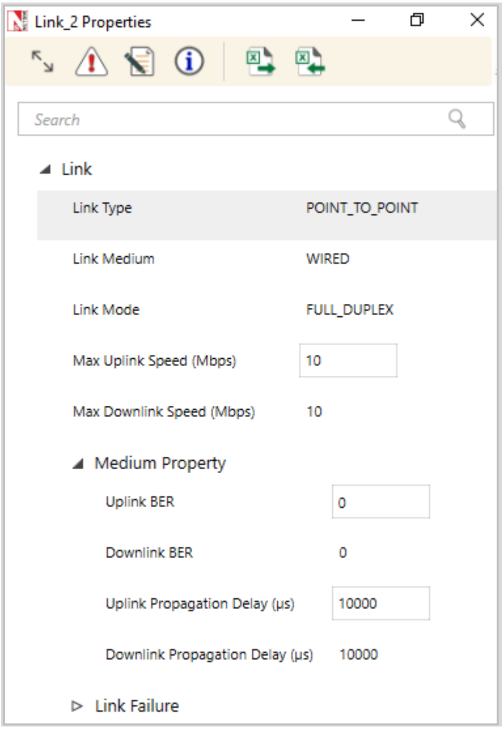


Figure 2-40: Link_ID_2 Properties window

Step 5: Right-click **Router 3** icon and select **Properties** to access the link's properties window (see Figure 2-41). In the **INTERFACE 2 (WAN)** tab, select the **NETWORK LAYER** properties, set **Buffer size (MB)** to **8.**

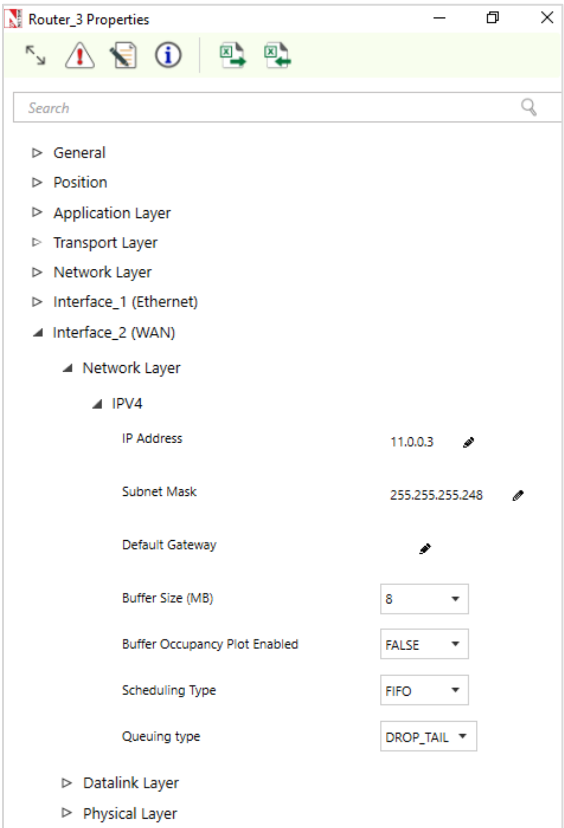


Figure 2-41: Router Properties window

Step 6: Enable Packet Trace check box. Packet Trace can be used for packet level analysis.

Step 7: Click on **Run** icon to access the Run Simulation window (see Figure 2-42) and set the **Simulation Time** to 100 seconds and click on **Run.**

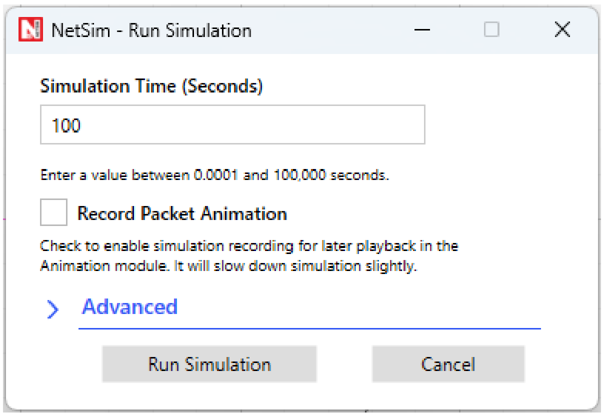


Figure 2-42: Run Simulation window

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 2-6).

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)	10 milliseconds
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 2-26
Inter Arrival Time – Distribution	Exponential
ROUTER PARAMETERS	
Buffer Size -Interface(WAN)	8
MISCELLANEOUS	
Simulation Time	100 Sec
Packet Trace	Enabled

Table 2-6: Detailed Network Parameters

2.3.3.2 Performance Measure

In Table 2-7 and Table 2-9, 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 and delay of packets in the network and in the buffers, and packet loss rate.

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

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

$$\text{PHY rate} = \frac{11680 + 54 \times 8}{v} = \frac{12112}{v} bps$$

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

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

where, T is the simulation time. And, let $N(u)$ denote the instantaneous number of packets in the network at time u. Then, the average number of packets in the network is computed as

$$average\ number\ of\ packet\ in\ the\ network\ \ = \frac{1}{T}\int_0^T N(u)\ \ du\ bits$$

Let $a_{i,l}$ and $d_{i,l}$ denote the time of arrival of a packet i into the link l (the corresponding router) and the time of departure of the packet i from the link l (the corresponding router), respectively. Then, the average queueing delay at the link l (the corresponding router) is computed as

$$\text{average queueing delay at link l} = \frac{1}{N}\sum_{i=1}^{N}(d_{i,l-}\ a_{i,l})$$

where N is the count of packets in the flow. Let ai and di denote the time of arrival of a packet i into the network (into the transport layer at the source node) and time of departure of the packet i from the network (from the transport layer at the destination node), respectively. Then, the end-to-end delay of the packet i is computed as $(d_{i-}\ a_i)$ seconds, and the average end to end delay of the packets in the flow is computed as

$$\text{average end to end packet delay} = \frac{1}{N}\sum_{i=1}^{N}(d_{i-}\ a_i)$$

2.3.3.2.1 Average Queue Computation from Packet Trace

- Open Packet Trace file using the **Open Packet Trace** option available in the Simulation Results window.
- In the Packet Trace, filter the data packets using the column **CONTROL PACKET TYPE/APP NAME** and the option **App1 CUSTOM** (see Figure 2-43).

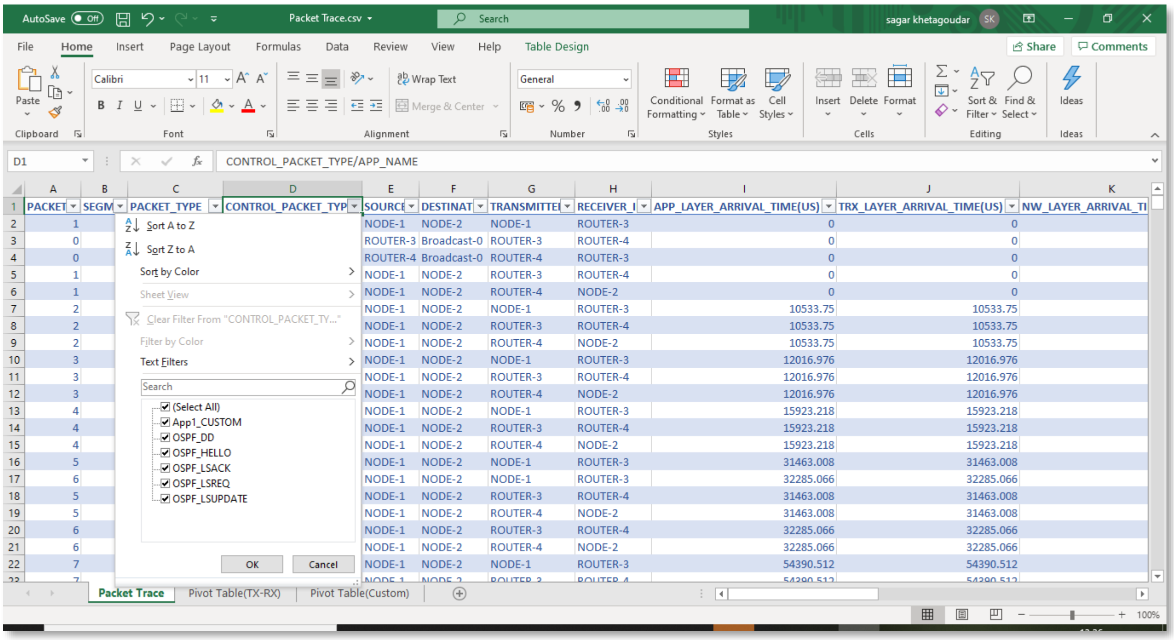


Figure 2-43: Filter the data packets in Packet Trace by selecting App1 CUSTOM.

- Now, to compute the average queue in Link 2, we will select **TRANSMITTER ID** as **ROUTER-3** and **RECEIVER ID** as **ROUTER-4**. This filters all the successful packets from Router 3 to Router 4.
- The columns **NW LAYER ARRIVAL TIME(US)** and **PHY LAYER ARRIVAL TIME(US)** correspond to the arrival time and departure time of the packets in the buffer at Link 2, respectively (see Figure 2-44).
- You may now count the number of packets arrivals (departures) into (from) the buffer upto time t using the NW LAYER ARRIVAL TIME(US) (PHY LAYER ARRIVAL TIME(US)) column. The difference between the number of arrivals and the number of departures gives us the number of packets in the queue at any time.

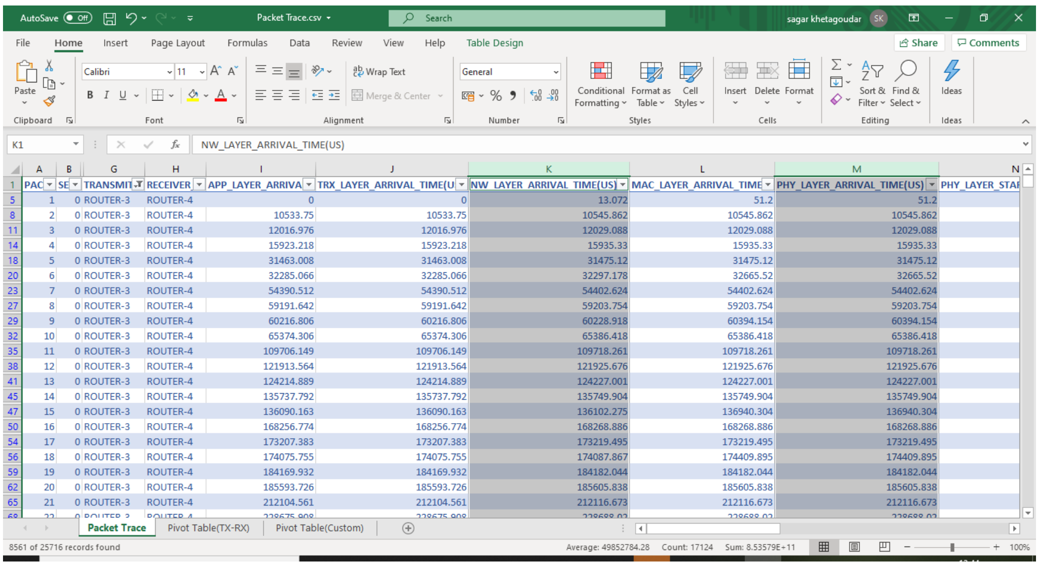


Figure 2-44: Packet arrival and departure times in the link buffer

- 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 PHY LAYER ARRIVAL TIME(US) and the NW LAYER ARRIVAL TIME(US) will give us the delay of a packet in the link (see Figure 2-45).

$$\text{Queuing Delay} = \text{PHY LAYER ARRIVAL TIME(US)} - \text{NW LAYER ARRIVAL TIME(US)}$$

N7 =[@[PHY_LAYER_ARRIVAL_TIME(US)]]-[@[NW_LAYER_ARRIVAL_TIME(US)]]

	I	J	K	L	M	N
1	APP_LAYER_ARRIVAL_TIME(US)	TRX_LAYER_ARRIVAL_TIME(US)	NW_LAYER_ARRIVAL_TIME(US)	MAC_LAYER_ARRIVAL_TIME(US)	PHY_LAYER_ARRIVAL_TIME(US)	Queuing Delay
7	0	0	13.072	51.2	51.2	38.128
10	1053.375	1053.375	1065.487	1241.6	1241.6	[illegible]
14	1201.698	1201.698	1213.81	2432	2432	1218.19
16	1592.322	1592.322	1604.434	3622.4	3622.4	2017.966
21	3146.301	3146.301	3158.413	4812.8	4812.8	1654.387
24	3228.507	3228.507	3240.619	6003.2	6003.2	2762.581
26	5439.051	5439.051	5451.163	7193.6	7193.6	1742.437
28	5919.164	5919.164	5931.276	8384	8384	2452.724
30	6021.681	6021.681	6033.793	9574.4	9574.4	3540.607
33	6537.431	6537.431	6549.543	10764.8	10764.8	4215.257
37	10970.615	10970.615	10982.727	11955.2	11955.2	972.473
41	12191.356	12191.356	12203.468	13145.6	13145.6	942.132
43	12421.489	12421.489	12433.601	14336	14336	1902.399
45	13573.779	13573.779	13585.891	15526.4	15526.4	1940.509
50	13609.016	13609.016	13621.128	16716.8	16716.8	3095.672
54	16825.677	16825.677	16837.789	17907.2	17907.2	1069.411
56	17320.738	17320.738	17332.85	19097.6	19097.6	1764.75
59	17407.576	17407.576	17419.688	20288	20288	2868.312
61	18416.993	18416.993	18429.105	21478.4	21478.4	3049.295
64	18559.373	18559.373	18571.485	22668.8	22668.8	4097.315
68	21210.456	21210.456	21222.568	23859.2	23859.2	2636.632
70	22867.591	22867.591	22879.703	25049.6	25049.6	2169.897

Figure 2-45: Queuing Delay

- Now, calculate the average queuing delay by taking the mean of the queueing delay of all the packets (see Figure 2-45)

2.3.3.2.2 Network Delay Computation from Packet Trace

- Open Packet Trace file using the **Open Packet Trace** option available in the Simulation Results window
- In the Packet Trace, filter the data packets using the column **CONTROL PACKET TYPE/APP NAME** and the option **App1 CUSTOM** (see Figure 2-43**)**.
- Now, we will select the **RECEIVER ID** as **NODE-2**. This filters all the successful packets in the network that reached Wired Node 2
- The columns **APP LAYER ARRIVAL TIME(US)** and **PHY LAYER END TIME(US)** correspond to the arrival time and departure time of the packets in the network respectively.
- You may now count the number of arrivals (departures) into (from) the network upto time t using the APP LAYER ARRIVAL TIME(US) (PHY LAYER END TIME(US)) column. The difference between the number of arrivals and the number of departures gives us the number of packets in the network at any time.
- Calculate the average number of packets in the network by taking the mean of the number of packets in network at every time interval during the simulation.
- Packet Delay at a per packet level can be calculated using the columns **Application Layer Arrival Time** and **Physical Layer End Time** in the packet trace as:

  End-to-End Delay = PHY LAYER END TIME(US) – APP LAYER ARRIVAL TIME(US)
- Calculate the average end-to-end packet delay by taking the mean of the difference between Phy Layer End Time and App Layer Arrival Time columns.

NOTE: To calculate average number of packets in queue refer the experiment on Throughput and Bottleneck Server Analysis.

2.3.3.3 Results

In Table 2-7, we report the flow average inter-arrival time (AIAT) and the corresponding application traffic generation rate (TGR), input flow rate (at the physical layer), average number of packets in the system, end-to-end packet delay in the network and packet loss rate.

AIAT v (in μs)	App layer traffic gen rate, $\frac{L}{v}$ (in Mbps)	PHY Rate with overheads (in Mbps)	Arrival Rate (in Pkts/sec)	End-to-End Packet Delay (in µs)	Avg no of pkts in system
11680	1	1.037	86	11282.27	0.97
5840	2	2.074	171	11368.05	1.95
3893	3.0003	3.1112	257	11474.12	2.95
2920	4	4.1479	342	11620.76	3.98
2336	5	5.1849	428	11834.14	5.06
1947	5.999	6.2209	514	12142.87	6.24
1669	6.9982	7.257	599	12663.59	7.59
1460	8	8.2959	685	13846.6	9.48
1298	8.9985	9.3313	770	17848.49	13.73
1284	9.0966	9.433	779	18941.64	14.76
1270	9.1969	9.537	787	20296.04	15.98
1256	9.2994	9.6433	796	22319.08	17.77
1243	9.3966	9.7442	805	25232.27	20.31
1229	9.5037	9.8552	814	31604.84	26
1217	9.5974	9.9523	822	42744.24	35.14

Table 2-7: Packet arrival rate, average number of packets in the system, end-to-end delay and packet loss rate. In this table “Average number of packets in the system” is calculated as “Arrival rate” times “End-to-end Delay”.

We can infer the following from Table 2-7,

- The average end-to-end packet delay (between the source and the destination) is bounded below by the sum of the packet transmission durations and the propagation delays of the constituent links. This value is equal to $2 \times 12\ \mu s$ (transmission time in the node–router links) + $1211 \mu s$ (transmission time in the router-router link) + $10000\ \mu s$ (propagation delay in the router-router link) which is $11{,}235\ \mu s$.
- As the input flow rate increases, the packet delay increases as well (due to congestion and queueing in the intermediate routers). As the input flow rate matches or exceeds the bottleneck link capacity, the end-to-end packet delay increases unbounded (limited by the buffer size).
- The average number of packets in the network can be found to be equal to the product of the average end-to-end packet delay and the average input flow rate into the network.

This is a validation of Little's law. In cases where the packet loss rate is positive, the arrival rate is to be multiplied by (1 - packet loss rate).

2.3.4 Independent verification of Little's law

We know that from Little's law

$$Average\ number\ of\ packets\ in\ the\ system = Arrival\ rate\ \times Delay$$

Additionally, we know that in simulation

$$Average\ number\ of\ packets\ in\ the\ system = \ Packet\ generated - packet\ received$$

We verify that the Average number of packets in the system computed from these two independent methods matches.

Since Little's law deals with averages, a good match between theory and simulation is improbable from any one trial. We therefore run multiple simulations for each scenario by changing the seed for the random number generator, and at the end of each simulation obtain (i) Packets Generated, and (ii) Packets received. We then subtract packets received from packets generated for each simulation run and then average over all simulation runs. This data is then compared against the Average number of packets in the system obtained theoretically.

Consider the case where AIAT = $1229\ \mu s$. We run 10 simulations per the table shown below.

RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Simulation)
1	2	814	30795.90699	25.06	81461	81430	31
2	3	814	27154.86007	22.10	81078	81056	22
3	4	814	34090.20559	27.74	81776	81749	27
4	5	814	28166.82512	22.92	81399	81377	22
5	6	814	26717.70032	21.74	81309	81279	30
6	7	814	32580.07104	26.51	81498	81456	42
7	8	814	28673.19279	23.33	81431	81426	5
8	9	814	31786.77442	25.86	81873	81860	13
9	10	814	28730.26673	23.38	81241	81231	10
10	11	814	29707.92242	24.17	81537	81515	22
Avg No. of pkts in the system (Theory)				**24.28**	**Avg No. of pkts in system (Simulation)**		**22.4**

Table 2-8: Results for validating the number of packets in systems for $1229\ \mu s$ IAT.

Observe that we are generating $\approx 80{,}000$ packets and the results from simulation and theory falls within a margin of 2 packets which shows good accuracy. Results for each of the average inter-arrival times (AIATs) are provided in the Appendix.

2.3.5 Average queue length and the average queuing delay

In Table 2-9, we report the average queue and average queueing delay at the intermediate routers (Wired Node 1, Router 3 and Router 4) and the average end-to-end packet delay as a function of the input flow rate.

Input Flow Rate (in Mbps)	Arrival Rate (in Pkts/sec)	Avg no of pkts in Queue			Average Queueing Delay (in µs)			End-to-End Packet Delay (in µs)
		Node 1	Router 3	Router 4	Node 1	Router 3	Router 4	
1.037	86	0	0	0	0.008	67.55	0	11270.07
2.074	171	0	0	0	0.015	153.26	0	11355.79
3.1112	257	0	0.08	0	0.021	259.47	0	11462.00
4.1479	342	0	0.13	0	0.029	406.54	0	11609.08
5.1849	428	0	0.26	0	0.035	619.21	0	11821.76
6.2209	514	0	0.45	0	0.046	928.11	0	12130.67
7.257	599	0	0.98	0	0.054	1488.91	0	12651.48
8.2959	685	0	1.93	0	0.062	2632.03	0	13834.61
9.3313	770	0	5.34	0	0.070	6625.60	0	17828.18
9.433	779	0	6.83	0	0.070	7691.11	0	18893.69
9.537	787	0	7.82	0	0.071	9095.84	0	20288.42
9.6433	796	0	7.82	0	0.071	11086.29	0	22288.88
9.7442	805	0	11.18	0	0.073	14040.96	0	25255.66
9.8552	814	0	16.44	0	0.073	20390.13	0	31604.83
9.9523	822	0	25.7	0	0.073	31451.06	0	42665.76
10.0598	831	0	43.28	0	0.074	51441.43	0	62660.26
10.1611	839	0	93.14	0	0.075	112374.10	0	123595.75
10.2644	847	0	437.92	0	0.076	518145.71	0	528204.59
10.3699	856	0	856.67	0	0.077	1010102.2	0	1021304.82
11.4049	942	0	3873.8	0	0.085	4614884.9	0	4626087.56
12.4481	1028	0	4593.1	0	0.093	5468670.4	0	5479885.16
13.4878	1114	0	4859.1	0	0.099	5786635.5	0	5797838.10
14.5228	1199	0	5185.6	0	0.106	5953385.9	0	5964588.26
15.5481	1284	0	5275.6	0	0.113	6055075.2	0	6066277.82

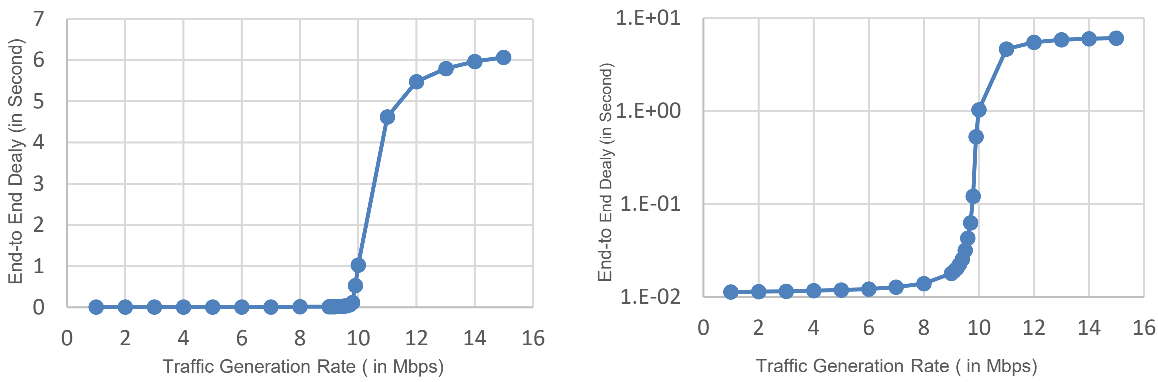
Table 2-9: Average queue and average queueing delay in the intermediate buffers and end-to-end packet delay.

We can infer the following from Table 2-9 .

- There is queue buildup as well as queueing delay at Router_3 (Link 2) as the input flow rate increases. Clearly, link 2 is the bottleneck link where the packets see large queueing delay.
- As the input flow rate matches or exceeds the bottleneck link capacity, the average queueing delay at the (bottleneck) server increases unbounded. Here, we note that the maximum queueing delay is limited by the buffer size (8 MB) and link capacity (10 Mbps), and an upper bounded is $8 \times 1024 \times 1024 \times 8\ 107\ =\ 6.7\ \text{seconds}.$

- The average number of packets in a queue can be found to be equal to the product of the average queueing delay and the average input flow rate into the network. This is again a validation of the Little's law. In cases where the packet loss rate is positive, the arrival rate is to be multiplied by $(1 - \text{packet loss rate})$.
- The average end-to-end packet delay can be found to be equal to the sum of the packet transmission delays (12.112μs (link 1), 1211μs (link 2), 12.112 μs (link3)), propagation delay (10000 μs) and the average queueing delay in the three links.

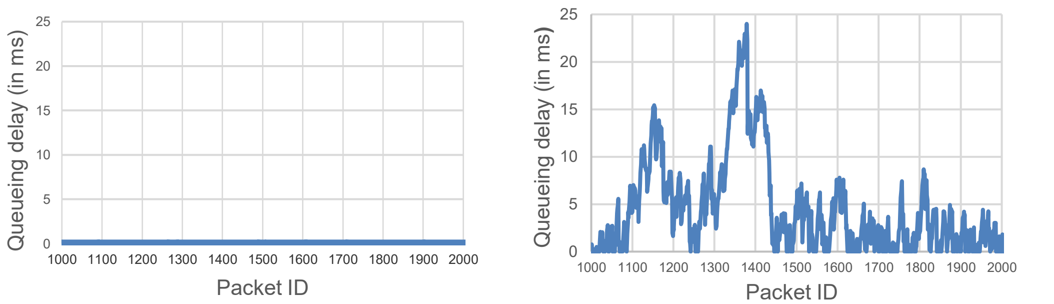
For the sake of the readers, we have made the following plots for clarity. In Figure 2-46, we plot the average end-to-end packet delay as a function of the traffic generation rate. We note that the average packet delay increases unbounded as the traffic generation rate matches or exceeds the bottleneck link capacity.



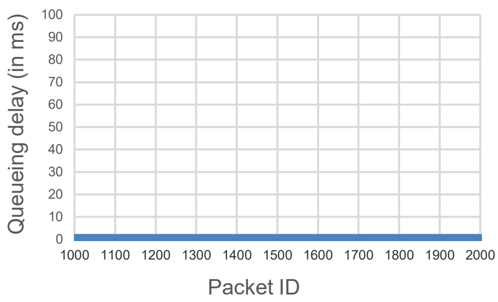
a) Linear Scale b) Log Scale

Figure 2-46: Average end-to-end packet delay as a function of the traffic generation rate.

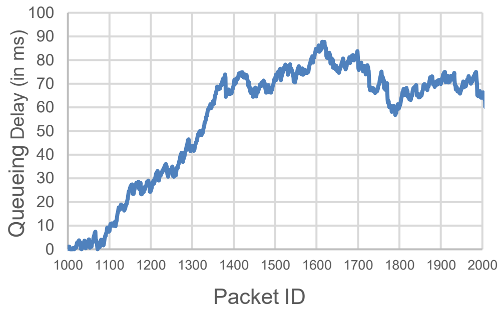
In Figure 2-47, we plot the queueing delay experienced by few packets at the buffers of Links 1 and 2 for two different input flow rates. We note that the packet delay is a stochastic process and is a function of the input flow rate and the link capacity as well.



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 2-47: Queueing Delay of packets at Wired_Node_1 (Link 1) and Router_3 (Link 2) for two different traffic generation rates

2.3.5.1.1 Bottleneck Server Analysis as M/G/1 Queue

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 queueing delay at the link buffer (assuming large buffer size) must be.

$$\text{average queueing delay } = \frac{1}{\mu} + \frac{1}{2\mu}\frac{\rho}{1-\rho} = \lambda \times \text{ average queue}$$

where ρ is the offered load to the link, λ is the input flow rate in packet arrivals per second and μ is the service rate of the link in packets served per second. Notice that the average queueing delay increases unbounded as $\rho \rightarrow 1$.

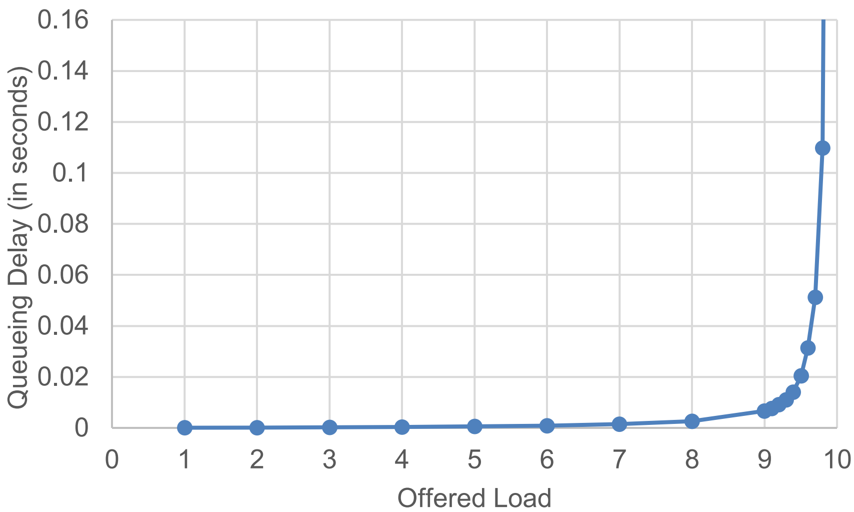


Figure 2-48: Average queueing delay (in seconds) at the bottleneck link 2 (at Router 3) Average queueing delay (in seconds) at the bottleneck link 2 (at Router 3) as a function of the offered load.

In Figure 2-48, we plot the average queueing delay (from simulation) and from (1) (from the bottleneck analysis) as a function of offered load ρ. Clearly, the bottleneck link analysis predicts the average queue (from simulation) very well. Also, we note from (1) that the network performance depends on λ and μ as $\frac{\lambda}{\mu} = \rho$ only.

2.3.6 Appendix

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
11680	1	2	86	11283.56	0.97	8623	8622	1
11680	2	3	86	11278.78	0.97	8535	8535	0
11680	3	4	86	11280.88	0.97	8566	8566	0
11680	4	5	86	11285.21	0.97	8538	8538	0
11680	5	6	86	11277.80	0.97	8514	8514	0
11680	6	7	86	11283.66	0.97	8583	8583	0
11680	7	8	86	11279.83	0.97	8577	8576	1
11680	8	9	86	11283.53	0.97	8562	8561	1
11680	9	10	86	11282.39	0.97	8442	8442	0
11680	10	11	86	11283.05	0.97	8472	8472	0
11680	**Avg No. of pkts in the system (Theory)**				**0.97**	**Avg No. of pkts in system (Simulation)**		**0.3**

Table 2-10: Results for validating the number of packets in systems for $11680\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
5840	1	2	171	11364.89	1.95	17202	17200	2
5840	2	3	171	11366.71	1.95	17076	17073	3
5840	3	4	171	11363.02	1.95	17161	17156	5
5840	4	5	171	11362.85	1.95	17155	17155	0
5840	5	6	171	11364.89	1.95	16997	16996	1
5840	6	7	171	11371.79	1.95	17155	17150	5
5840	7	8	171	11367.94	1.95	17112	17110	2
5840	8	9	171	11370.50	1.95	17178	17177	1
5840	9	10	171	11365.97	1.95	17002	17000	2
5840	10	11	171	11366.97	1.95	17077	17076	1
5840	**Avg No. of pkts in the system (Theory)**				**1.95**	**Avg No. of pkts in system (Simulation)**		**2.2**

Table 2-11: Results for validating the number of packets in systems for $5840\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
3893	1	2	257	11475.81	2.95	25860	25854	6
3893	2	3	257	11477.72	2.95	25556	25555	1
3893	3	4	257	11472.91	2.95	25680	25676	4
3893	4	5	257	11479.34	2.95	25737	25735	2

3893	5	6	257	11481.69	2.95	25714	25710	4
3893	6	7	257	11480.16	2.95	25785	25782	3
3893	7	8	257	11480.76	2.95	25687	25683	4
3893	8	9	257	11475.06	2.95	25736	25732	4
3893	9	10	257	11470.14	2.95	25522	25519	3
3893	10	11	257	11478.44	2.95	25629	25625	4
3893	**Avg No. of pkts in the system (Theory)**				**2.95**	**Avg No. of pkts in system (Simulation)**		**3.5**

Table 2-12: Results for validating the number of packets in systems for $3893\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
2920	1	2	342	11623.20	3.98	34593	34589	4
2920	2	3	342	11626.28	3.98	34061	34059	2
2920	3	4	342	11614.69	3.98	34236	34228	8
2920	4	5	342	11625.14	3.98	34245	34240	5
2920	5	6	342	11629.86	3.98	34215	34209	6
2920	6	7	342	11631.24	3.98	34349	34344	5
2920	7	8	342	11627.24	3.98	34320	34318	2
2920	8	9	342	11625.62	3.98	34390	34387	3
2920	9	10	342	11617.11	3.98	34033	34030	3
2920	10	11	342	11624.00	3.98	34251	34243	8
2920	**Avg No. of pkts in the system (Theory)**				**3.98**	**Avg No. of pkts in system (Simulation)**		**4.6**

Table 2-13: Results for validating the number of packets in systems for $2920\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
2336	1	2	428	11838.59	5.07	43145	43139	6
2336	2	3	428	11816.87	5.06	42445	42440	5
2336	3	4	428	11816.53	5.06	42801	42794	7
2336	4	5	428	11840.76	5.07	42820	42813	7
2336	5	6	428	11827.88	5.06	42798	42793	5
2336	6	7	428	11847.27	5.07	42986	42982	4
2336	7	8	428	11835.97	5.07	42862	42853	9
2336	8	9	428	11836.49	5.07	42978	42974	4
2336	9	10	428	11826.71	5.06	42664	42659	5
2336	10	11	428	11825.99	5.06	42792	42787	5
2336	**Avg No. of pkts in the system (Theory)**				**5.06**	**Avg No. of pkts in system (Simulation)**		**5.7**

Table 2-14: Results for validating the number of packets in systems for $2336\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pks Generated	Pkts Received	No. of pkts in the system (Sim)
1947	1	2	514	12149.61	6.24	51581	51575	6

1947	2	3	514	12132.72	6.23	51159	51147	12
1947	3	4	514	12118.78	6.22	51388	51383	5
1947	4	5	514	12158.55	6.24	51387	51378	9
1947	5	6	514	12146.59	6.24	51338	51334	4
1947	6	7	514	12156.33	6.24	51515	51510	5
1947	7	8	514	12146.09	6.24	51347	51339	8
1947	8	9	514	12168.86	6.25	51777	51772	5
1947	9	10	514	12150.55	6.24	51271	51265	6
1947	10	11	514	12141.11	6.24	51355	51351	4
1947	**Avg No. of pkts in the system (Theory)**				**6.24**	**Avg No. of pkts in system (Simulation)**		**6.4**

Table 2-15: Results for validating the number of packets in systems for $1947\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
1669	1	2	599	12694.47	7.61	60146	60138	8
1669	2	3	599	12659.47	7.59	59736	59728	8
1669	3	4	599	12652.29	7.58	60020	60011	9
1669	4	5	599	12712.53	7.62	59986	59975	11
1669	5	6	599	12687.75	7.60	59890	59878	12
1669	6	7	599	12704.78	7.61	60242	60230	12
1669	7	8	599	12680.35	7.60	60047	60038	9
1669	8	9	599	12728.32	7.63	60458	60456	2
1669	9	10	599	12676.02	7.59	59870	59858	12
1669	10	11	599	12687.70	7.60	60020	60015	5
1669	**Avg # of pkts in the system (Theory)**				**7.60**	**Avg # of pkts in system (Simulation)**		**8.8**

Table 2-16: Results for validating the number of packets in systems for $1669\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
1460	1	2	685	13841.40	9.48	68752	68734	18
1460	2	3	685	13703.41	9.39	68303	68295	8
1460	3	4	685	13720.77	9.40	68604	68597	7
1460	4	5	685	13871.45	9.50	68522	68514	8
1460	5	6	685	13825.66	9.47	68390	68380	10
1460	6	7	685	13880.92	9.51	68731	68719	12
1460	7	8	685	13840.10	9.48	68524	68508	16
1460	8	9	685	13928.14	9.54	68992	68985	7
1460	9	10	685	13771.70	9.43	68422	68411	11
1460	10	11	685	13842.15	9.48	68587	68571	16
1460	**Avg No. of pkts in the system (Theory)**				**9.47**	**Avg No. of pkts in system (Simulation)**		**11.3**

Table 2-17: Results for validating the number of packets in systems for$1460\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
1298	1	2	770	17642.09	13.59	77110	77094	16
1298	2	3	770	17284.05	13.32	76765	76743	22
1298	3	4	770	17906.68	13.80	77332	77316	16
1298	4	5	770	17942.55	13.82	77100	77089	11
1298	5	6	770	17330.87	13.35	76921	76913	8
1298	6	7	770	17980.07	13.85	77213	77201	12
1298	7	8	770	17793.99	13.71	77121	77099	22
1298	8	9	770	18201.85	14.02	77572	77566	6
1298	9	10	770	17616.95	13.57	76862	76856	6
1298	10	11	770	17972.09	13.85	77092	77083	9
1298	**Avg No. of pkts in the system (Theory)**				**13.69**	**Avg # of pkts in system (Simulation)**		**12.8**

Table 2-18: Results for validating the number of packets in systems for $1298\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
1284	1	2	779	18677.06	14.55	77980	77947	33
1284	2	3	779	18228.55	14.20	77602	77594	8
1284	3	4	779	19086.73	14.87	78191	78171	20
1284	4	5	779	18916.42	14.73	77924	77903	21
1284	5	6	779	18205.63	14.18	77778	77757	21
1284	6	7	779	19228.28	14.98	78067	78045	22
1284	7	8	779	18854.90	14.68	77967	77951	16
1284	8	9	779	19222.07	14.97	78393	78373	20
1284	9	10	779	18531.55	14.43	77697	77687	10
1284	10	11	779	19097.05	14.87	77984	77974	10
1284	**Avg No. of pkts in the system (Theory)**				**14.65**	**Avg No. of pkts in system (Simulation)**		**18.1**

Table 2-19: Results for validating the number of packets in systems for $1284\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
1270	1	2	787	20051.08	15.79	78768	78761	7
1270	2	3	787	19407.29	15.28	78449	78429	20
1270	3	4	787	20760.65	16.35	79125	79091	34
1270	4	5	787	20133.15	15.85	78788	78766	22
1270	5	6	787	19314.91	15.21	78692	78666	26
1270	6	7	787	20993.89	16.53	78860	78849	11
1270	7	8	787	20085.40	15.82	78800	78791	9
1270	8	9	787	20607.46	16.23	79223	79209	14
1270	9	10	787	19836.88	15.62	78579	78562	17

1270	10	11	787	20480.04	16.13	78868	78847	21
1270	**Avg No. of pkts in the system (Theory)**				**15.88**	**Avg No. of pkts in system (Simulation)**		**18.1**

Table 2-20: Results for validating the number of packets in systems for $1270\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
1256	1	2	796	22133.73	17.62	79679	79649	30
1256	2	3	796	21074.87	16.78	79286	79272	14
1256	3	4	796	23681.15	18.85	80019	79998	21
1256	4	5	796	21731.11	17.30	79610	79601	9
1256	5	6	796	20840.28	16.59	79586	79562	24
1256	6	7	796	23481.11	18.70	79751	79728	23
1256	7	8	796	21819.70	17.37	79687	79664	23
1256	8	9	796	22823.29	18.17	80158	80130	28
1256	9	10	796	21818.70	17.37	79493	79466	27
1256	10	11	796	22451.04	17.88	79797	79776	21
1256	**Avg No. of pkts in the system (Theory)**				**17.66**	**Avg No. of pkts in system (Simulation)**		**22**

Table 2-21: Results for validating the number of packets in systems for $1256\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
1243	1	2	805	25180.05	20.26	80517	80498	19
1243	2	3	805	23621.91	19.00	80152	80138	14
1243	3	4	805	27685.64	22.27	80894	80844	50
1243	4	5	805	23932.33	19.25	80479	80447	32
1243	5	6	805	22965.12	18.48	80394	80378	16
1243	6	7	805	27007.94	21.73	80551	80542	9
1243	7	8	805	24249.05	19.51	80528	80500	28
1243	8	9	805	26040.47	20.95	80995	80959	36
1243	9	10	805	24433.37	19.66	80356	80336	20
1243	10	11	805	25106.62	20.20	80599	80591	8
1243	**Avg No. of pkts in the system (Theory)**				**20.13**	**Avg No. of pkts in system (Simulation)**		**23.2**

Table 2-22: Results for validating the number of packets in systems for $1243\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
1229	1	2	814	30795.91	25.06	81461	81430	31
1229	2	3	814	27154.86	22.10	81078	81056	22
1229	3	4	814	34090.21	27.74	81776	81749	27
1229	4	5	814	28166.83	22.92	81399	81377	22
1229	5	6	814	26717.70	21.74	81309	81279	30
1229	6	7	814	32580.07	26.51	81498	81456	42
1229	7	8	814	28673.19	23.33	81431	81426	5

1229	8	9	814	31786.77	25.86	81873	81860	13
1229	9	10	814	28730.27	23.38	81241	81231	10
1229	10	11	814	29707.92	24.17	81537	81515	22
1229	**Avg No. of pkts in the system (Theory)**				**24.28**	**Avg No. of pkts in system (Simulation)**		**22.4**

Table 2-23: Results for validating the number of packets in systems for $1229\ \mu s$ IAT.

IAT	RNG Seed 1	RNG Seed 2	Arrival Rate (Pkts/Sec)	End to End Delay (μs)	No of pkts in the system (Theory)	Pkts Generated	Pkts Received	No. of pkts in the system (Sim)
1217	1	2	822	41953.86	34.47	82287	82246	41
1217	2	3	822	33117.16	27.21	81893	81870	23
1217	3	4	822	44215.58	36.33	82578	82519	59
1217	4	5	822	34847.11	28.63	82199	82176	23
1217	5	6	822	33705.32	27.70	82163	82095	68
1217	6	7	822	40737.94	33.47	82248	82240	8
1217	7	8	822	35997.53	29.58	82189	82176	13
1217	8	9	822	49519.81	40.69	82676	82667	9
1217	9	10	822	37348.67	30.69	81989	81979	10
1217	10	11	822	36543.17	30.03	82318	82305	13
1217	**Avg No. of pkts in the system (Theory)**				**31.88**	**Avg No. of pkts in system (Simulation)**		**26.7**

Table 2-24: Results for validating the number of packets in systems for $1217\ \mu s$ IAT.

2.4 Throughput and Bottleneck Server Analysis (Level 2)

2.4.1 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

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

2.4.1.1 A Single Flow Scenario

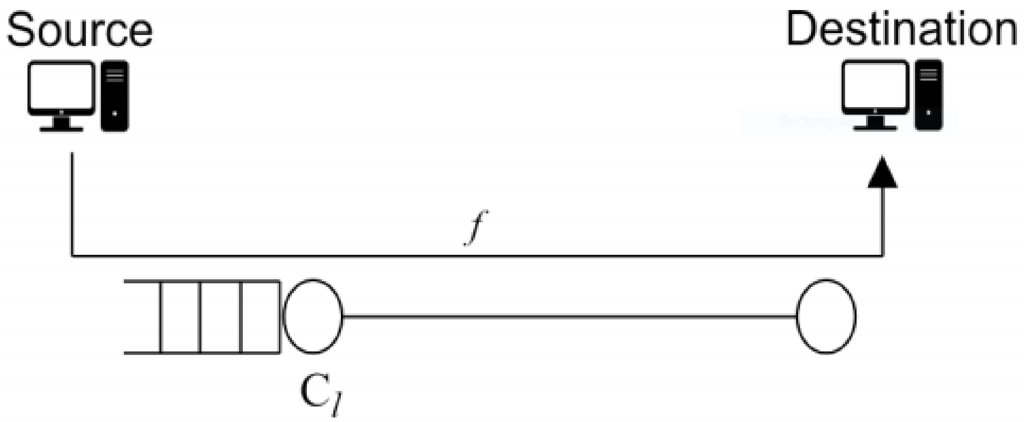


Figure 2-49: A flow $\boldsymbol{f}$ passing through a link $\boldsymbol{l}$ of fixed capacity $\boldsymbol{C_l}$.

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.

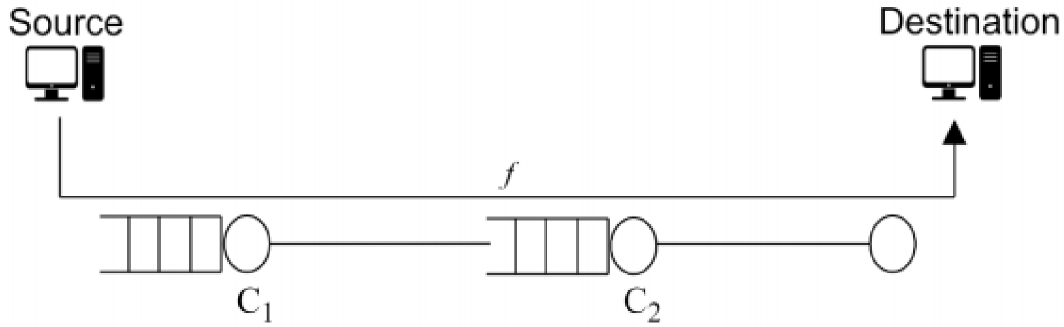


Figure 2-50: A single flow $\boldsymbol{f}$ passing through a series of links. The link with the least capacity will be identified as the bottleneck link for the flow $\boldsymbol{f}$

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 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.

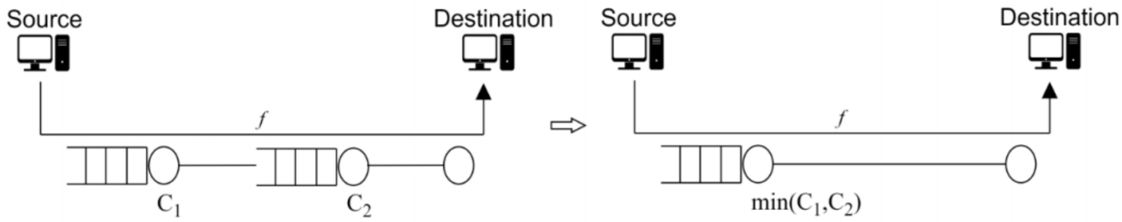


Figure 2-51: Approximation of a network using bottleneck server technique.

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 2-51. 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.

2.4.1.2 Multiple Flow Scenario

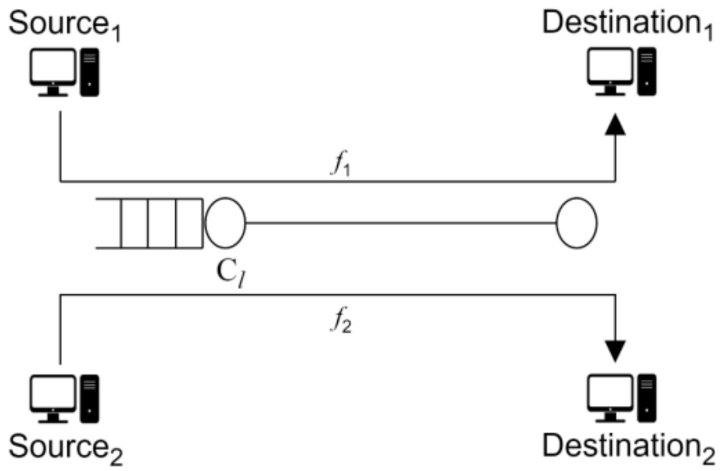


Figure 2-52: Two flows f_1 and f_2 passing through a link l of capacity C_l

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)

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

2.4.2 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 2-53.

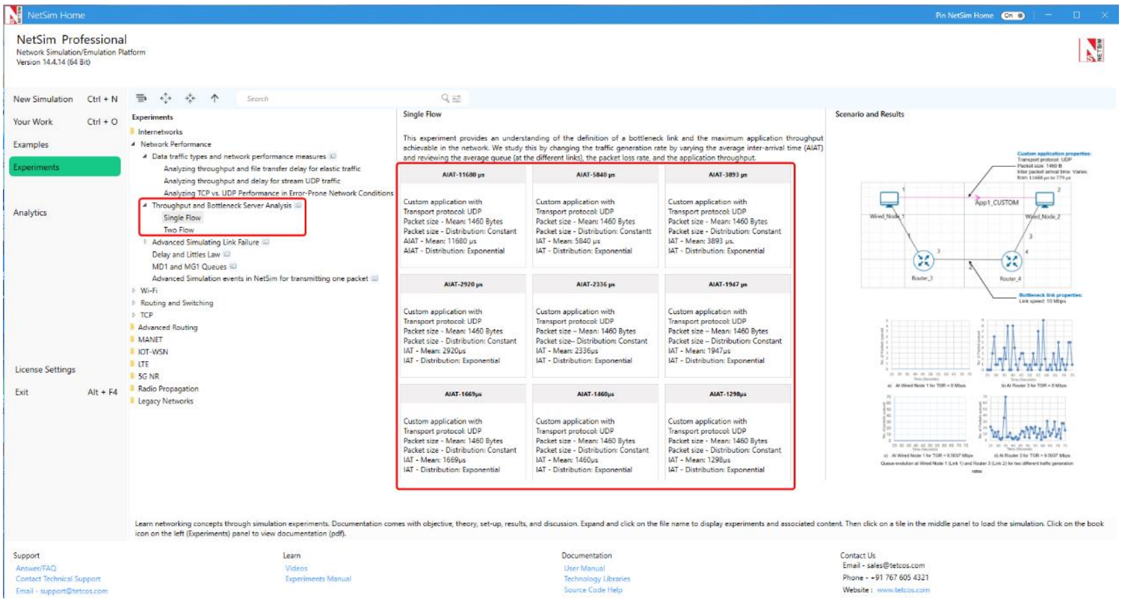


Figure 2-53: List of scenarios for the example of Throughput and Bottleneck Server Analysis

2.4.3 Part - 1: A Single Flow Scenarios

We will study a simple network setup with a single flow illustrated in Figure 2-54 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.

2.4.3.1 Procedure

We will simulate the network setup illustrated in Figure 2-54 with the configuration parameters listed in detail in Table 2-25 to study the single flow scenario.

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

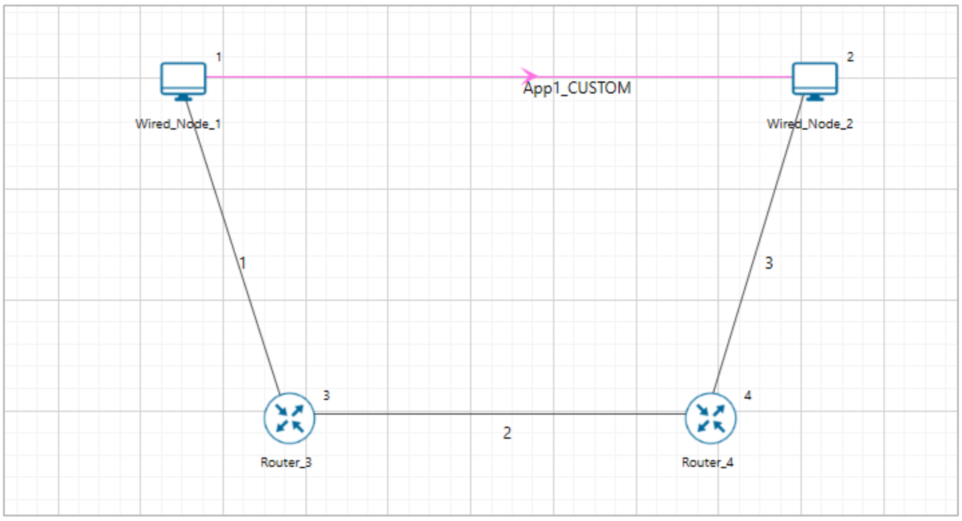


Figure 2-54: Network set up for studying a single flow

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 2-54).

Step 2: Click on the 'Set Traffic' tab in the ribbon at the top and configure a custom application between two wired nodes. To set the application properties as shown in below figure, click on the created application, expand the property panel on the right, and set the **transport protocol** to **UDP**. In the Packet Size tab, select **Distribution** as **CONSTANT** and set the **value** to **1460 bytes**. In the Inter-Arrival Time tab, select **Distribution** as **EXPONENTIAL** and set the **mean** to **11680** microseconds.

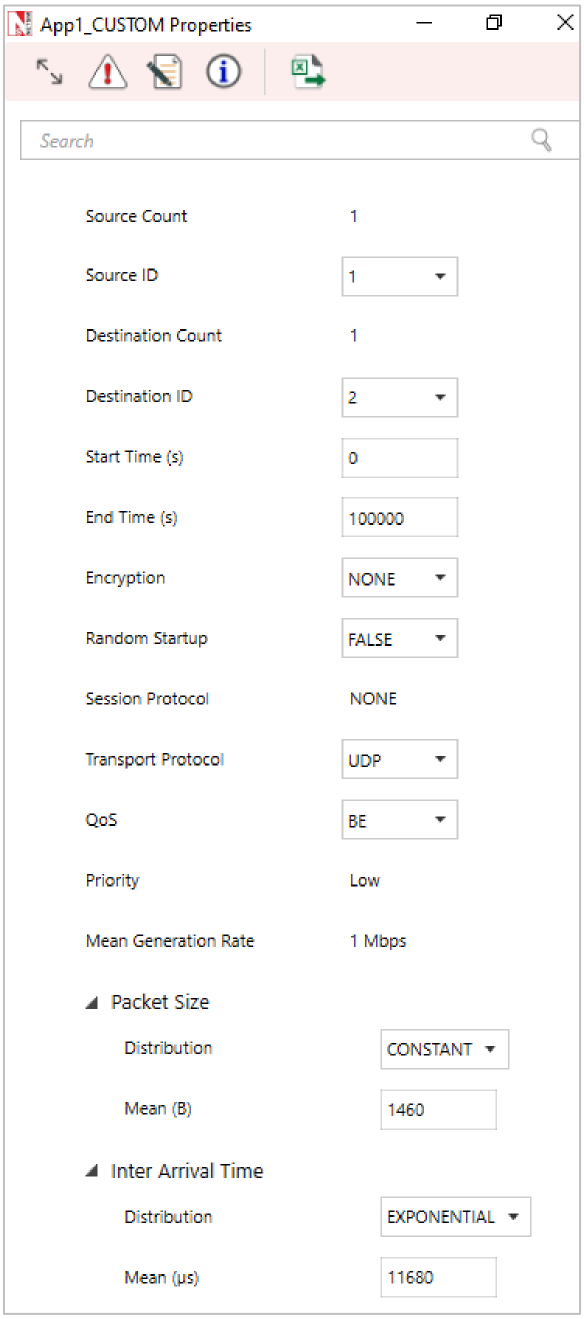


Figure 2-55: Application configuration window

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

Step 4: Click the link ID (of a wired link) and expand the property panel on right (see Figure 2-56). 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.

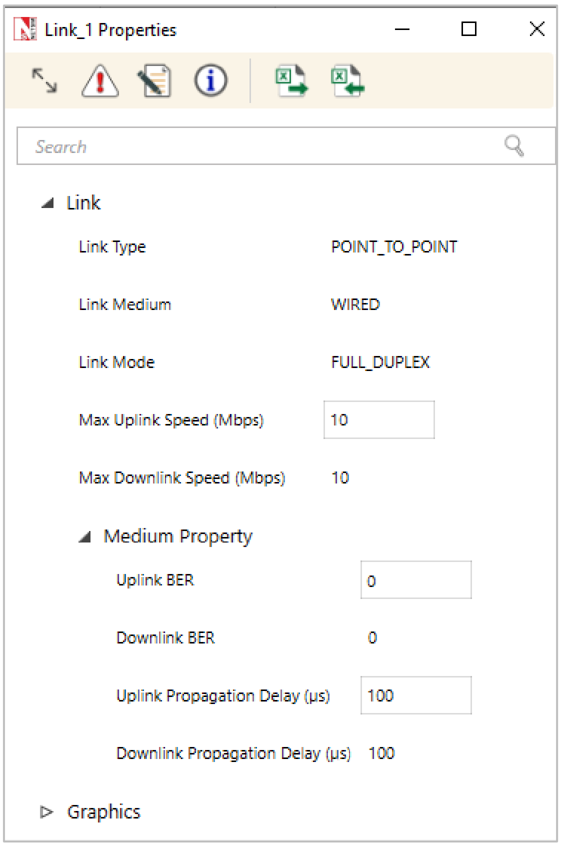


Figure 2-56: Link Properties window

Step 5: Click on Router 3, expand the property panel on right and set the Buffer size to 8 MB in network layer of Interface 2 (WAN) as shown in Figure 2-57.

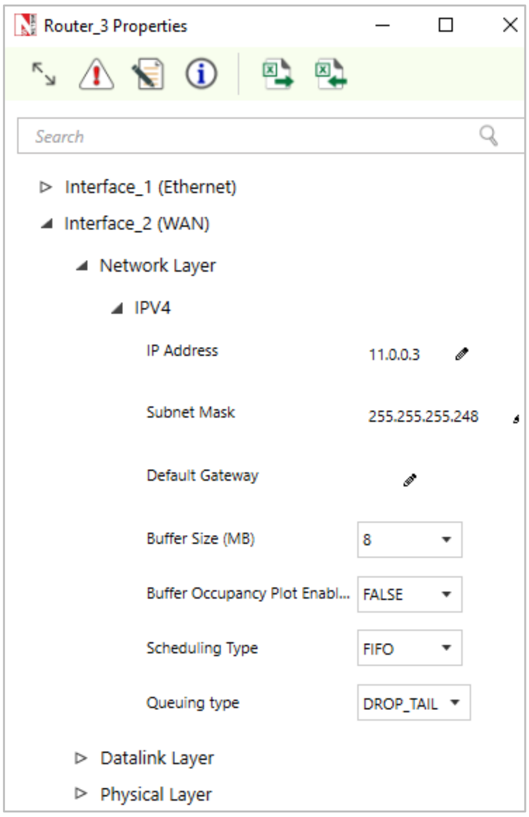


Figure 2-57: Router Properties window

Step 6: Click on Configure reports tab in ribbon on the top and enable packet trace. Packet Trace can be used for packet level analysis.

Step 7: Click on **Run** icon to access the Run Simulation window (see Figure 2-58) and set the **Simulation Time** to 100 seconds and click on **Run**.

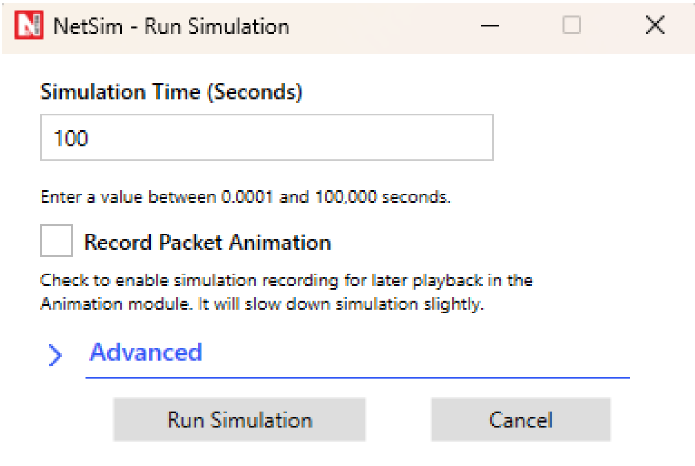


Figure 2-58: Run Simulation window

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 2-25).

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)
Inter Arrival Time – Distribution	Exponential
ROUTER PARAMETERS	
Buffer Size	8
MISCELLANEOUS	
Simulation Time	100 Sec
Packet Trace	Enabled

Table 2-25: Detailed Network Parameters

2.4.3.2 Performance Measure

In Table 2-26, 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).

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

2.4.3.2.1 Average Queue Computation from Packet Trace

- Open Packet Trace from the Simulation Results window as shown below

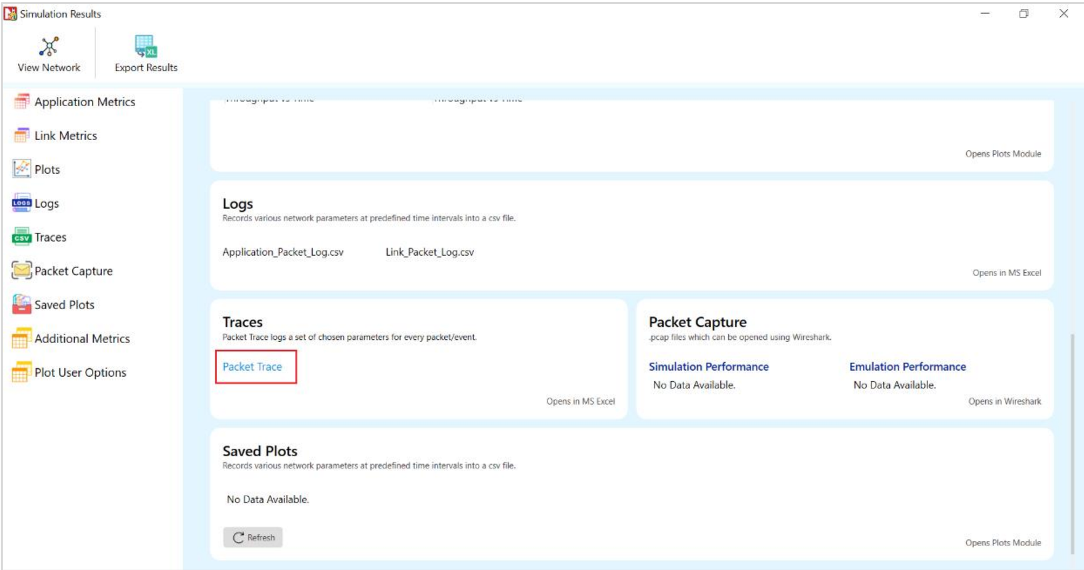


Figure 2-59: Opening Packet trace from Simulation results window

- In packet trace, click on Insert on Top ribbon → Select Pivot Table as shown in below figure

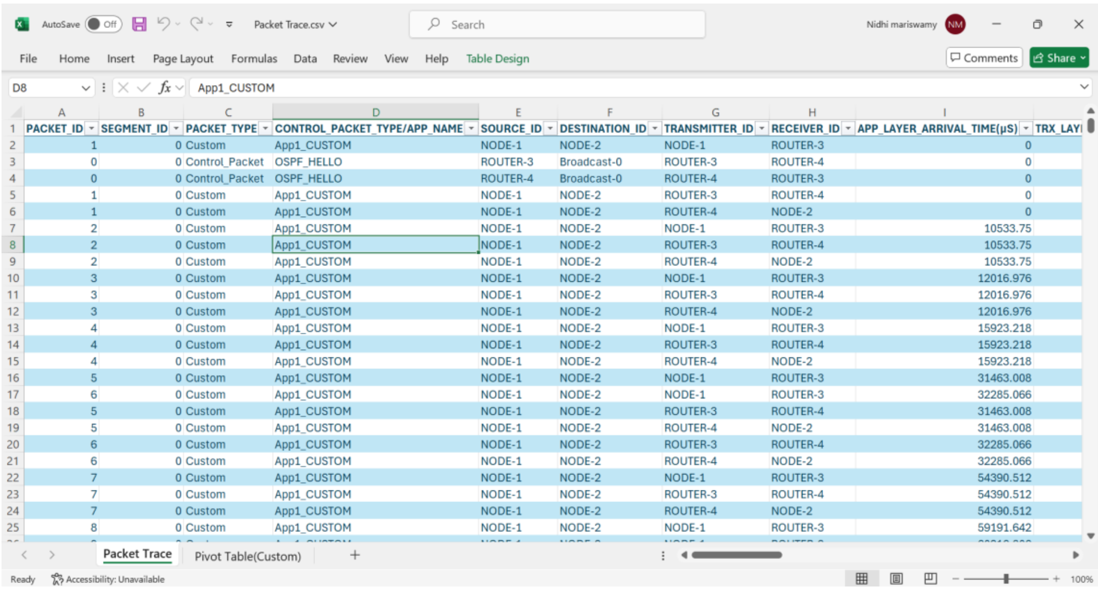


Figure 2-60: Creating pivot table from packet trace

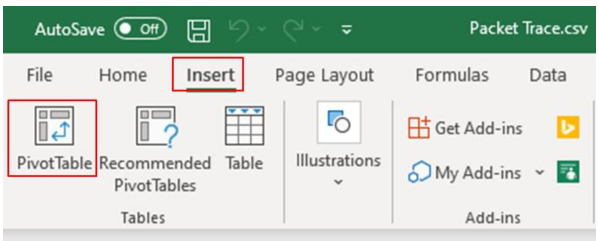


Figure 2-61: Top Ribbon

- The packet trace will be selected and Table 1 is mentioned as defualt, just click on OK.

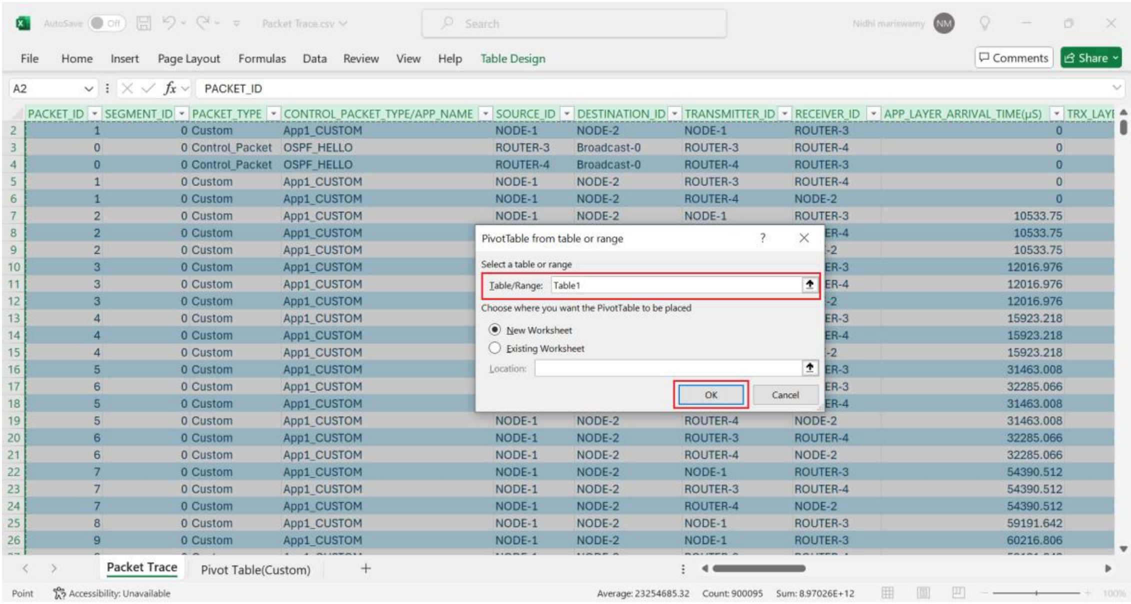


Figure 2-62: Packet Trace Pivot Table

- Then we will get blank Pivot table.

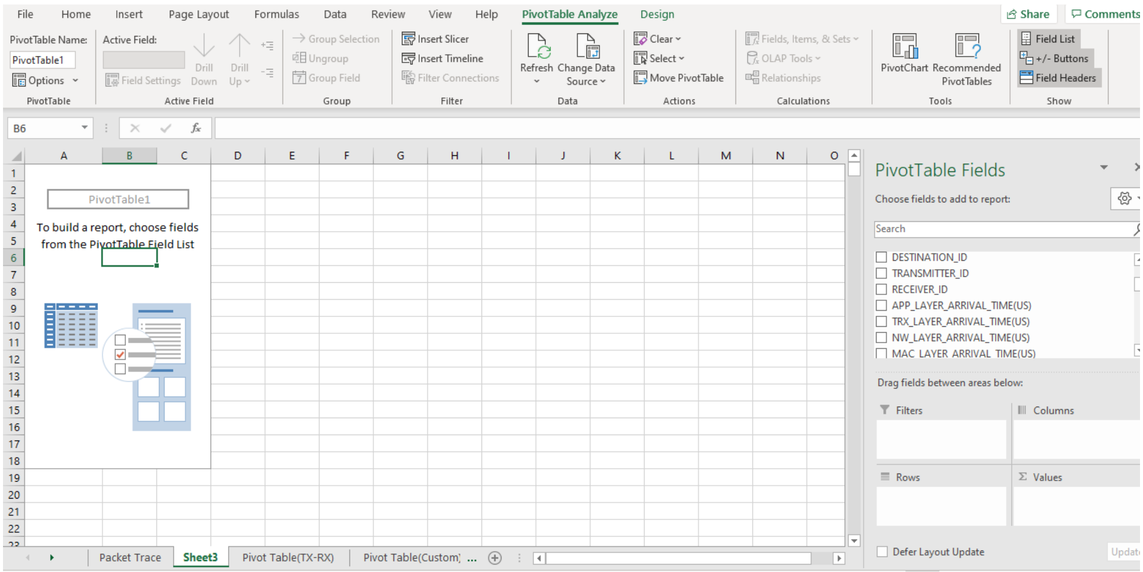
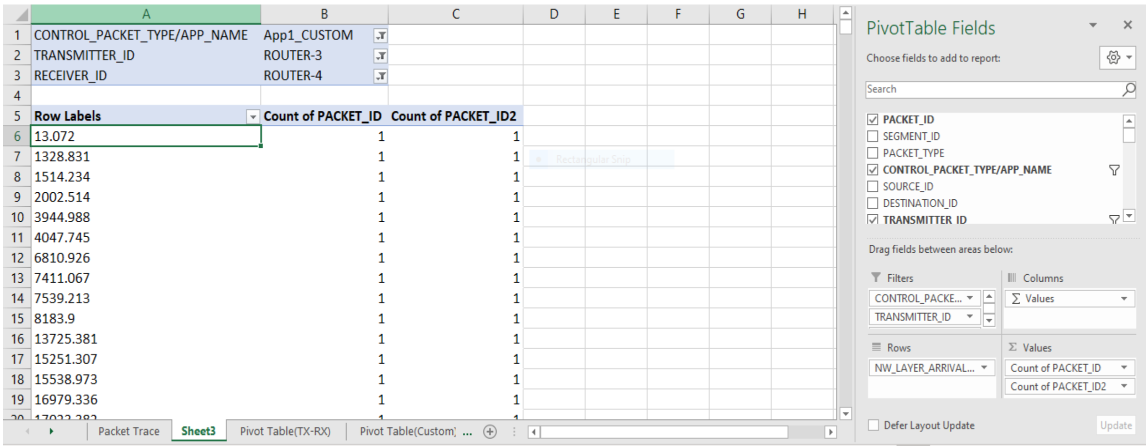


Figure 2-63: Blank Pivot Table

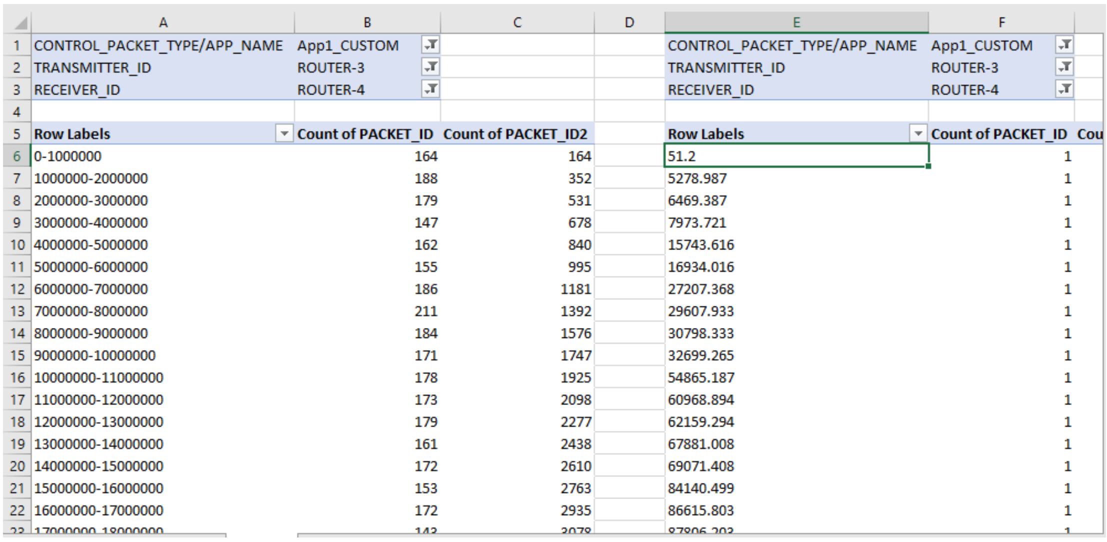
- **Packet ID** drag and drop into **Values field** for 2 times, **CONTROL PACKET TYPE or APP NAME, TRANSMITTER ID, RECEIVER ID** into **Filter field, NW LAYER ARRIVAL TIME (US)** to **Rows field** see Figure 2-64.
- Change **Sum of PACKET ID** -> Values Field Settings **->**Select **Count -> ok** for both Values field**, CONTROL PACKET TYPE** to **APP1 CUSTOM, TRANSMITTER ID** to **Router 3 and RECEIVER ID to Router 4**



	A	B	C
1	CONTROL_PACKET_TYPE/APP_NAME	App1_CUSTOM	
2	TRANSMITTER_ID	ROUTER-3	
3	RECEIVER_ID	ROUTER-4	
4			
5	Row Labels	Count of PACKET_ID	Count of PACKET_ID2
6	13.072	1	1
7	1328.831	1	1
8	1514.234	1	1
9	2002.514	1	1
10	3944.988	1	1
11	4047.745	1	1
12	6810.926	1	1
13	7411.067	1	1
14	7539.213	1	1
15	8183.9	1	1
16	13725.381	1	1
17	15251.307	1	1
18	15538.973	1	1
19	16979.336	1	1

Figure 2-64: Adding fields into Filter, Columns, Rows and Values

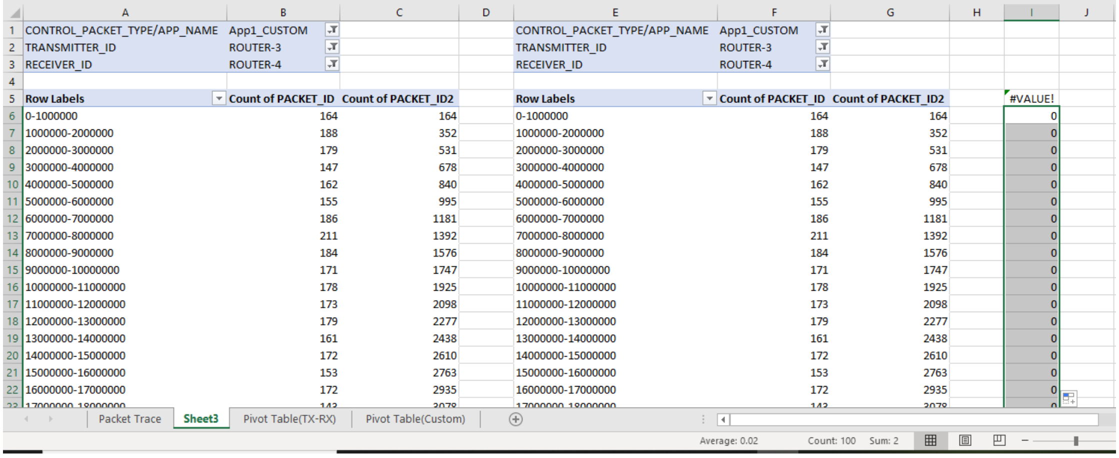
- Right click on first value of Row Labels ->Group ->Select **By** value as 1000000.
- Go to Values field under left click on Count of PACKET ID2 ->Values Field Settings-> click on show values as -> **Running total in**-> click on OK.
- Again, create one more Pivot Table, Click on Insert on Top ribbon → Select Pivot Table.
- Then select packet trace and press Ctrl + A → Select ok
- Then we will get blank Pivot table see Figure 2-65.
- **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 2-65,
- 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.



	A	B	C	D	E	F
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
6	0-1000000	164	164		51.2	1
7	1000000-2000000	188	352		5278.987	1
8	2000000-3000000	179	531		6469.387	1
9	3000000-4000000	147	678		7973.721	1
10	4000000-5000000	162	840		15743.616	1
11	5000000-6000000	155	995		16934.016	1
12	6000000-7000000	186	1181		27207.368	1
13	7000000-8000000	211	1392		29607.933	1
14	8000000-9000000	184	1576		30798.333	1
15	9000000-10000000	171	1747		32699.265	1
16	10000000-11000000	178	1925		54865.187	1
17	11000000-12000000	173	2098		60968.894	1
18	12000000-13000000	179	2277		62159.294	1
19	13000000-14000000	161	2438		67881.008	1
20	14000000-15000000	172	2610		69071.408	1
21	15000000-16000000	153	2763		84140.499	1
22	16000000-17000000	172	2935		86615.803	1

Figure 2-65: 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 2-66



	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

Packet Trace | Sheet3 | Pivot Table(TX-RX) | Pivot Table(Custom)

Average: 0.02 Count: 100 Sum: 2

Figure 2-66: Average Packets in Queue

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

2.4.3.3 Results

In Table 2-26, 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)			Average Throughput (in Mbps)	Packet Loss Rate (in percent)
			Wired Node1 (Link 1)	Router 3 (Link 2)	Router4 (Link 3)		
11680	1	1.037	0	0	0	0.999925	0
5840	2	2.074	0	0.03	0	1.998565	0
3893	3.0003	3.1112	0	0.04	0	2.999541	0
2920	4	4.1479	0	0.11	0	3.996546	0
2336	5	5.1849	0	0.32	0	5.010136	0
1947	5.999	6.2209	0	0.63	0	6.000133	0.01
1669	6.9982	7.257	0	0.89	0	7.002744	0
1460	8	8.2959	0	1.67	0	8.028482	0
1298	8.9985	9.3313	0	5.56	0	9.009952	0.01
1284	9.0966	9.433	0	6.82	0	9.106429	0.01
1270	9.1969	9.537	0	7.93	0	9.210147	0.01
1256	9.2994	9.6433	0	7.24	0	9.314099	0
1243	9.3966	9.7442	0	10.64	0	9.415014	0.01
1229	9.5037	9.8552	0	15.97	0	9.521886	0.02
1217	9.5974	9.9523	0	26.15	0	9.61591	0.01
1204	9.701	10.0598	0	42.96	0	9.71776	0.05
1192	9.7987	10.1611	0	89.42	0	9.796366	0.26
1180	9.8983	10.2644	0	440.24	0	9.807813	1.15
1168	10	10.3699	0	855.38	0	9.809098	2.09
1062	10.9981	11.4049	0	3989.26	0	9.811667	11.00

973	12.0041	12.4481	0	4752.75	0	9.811667	18.53
898	13.0067	13.4878	0	5035.93	0	9.811667	24.75
834	14.0048	14.5228	0	5185.56	0	9.811667	30.09
779	14.9936	15.5481	0	5275.62	0	9.811667	34.75

Table 2-26: Average queue, throughput and loss rate as a function of traffic generation rate.

We can infer the following from Table 2-26,

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 2 60, 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).

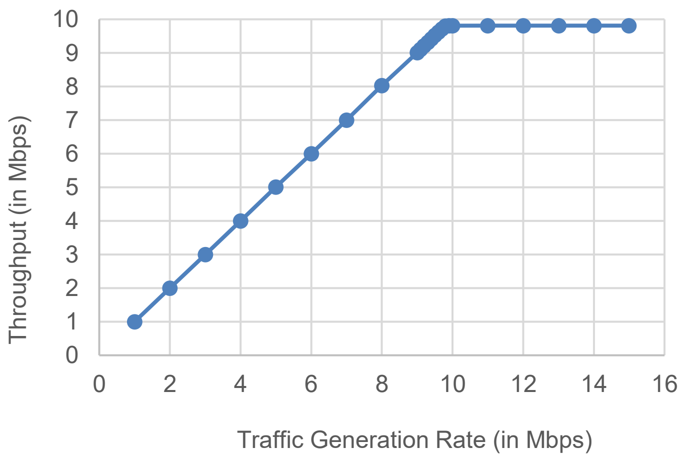
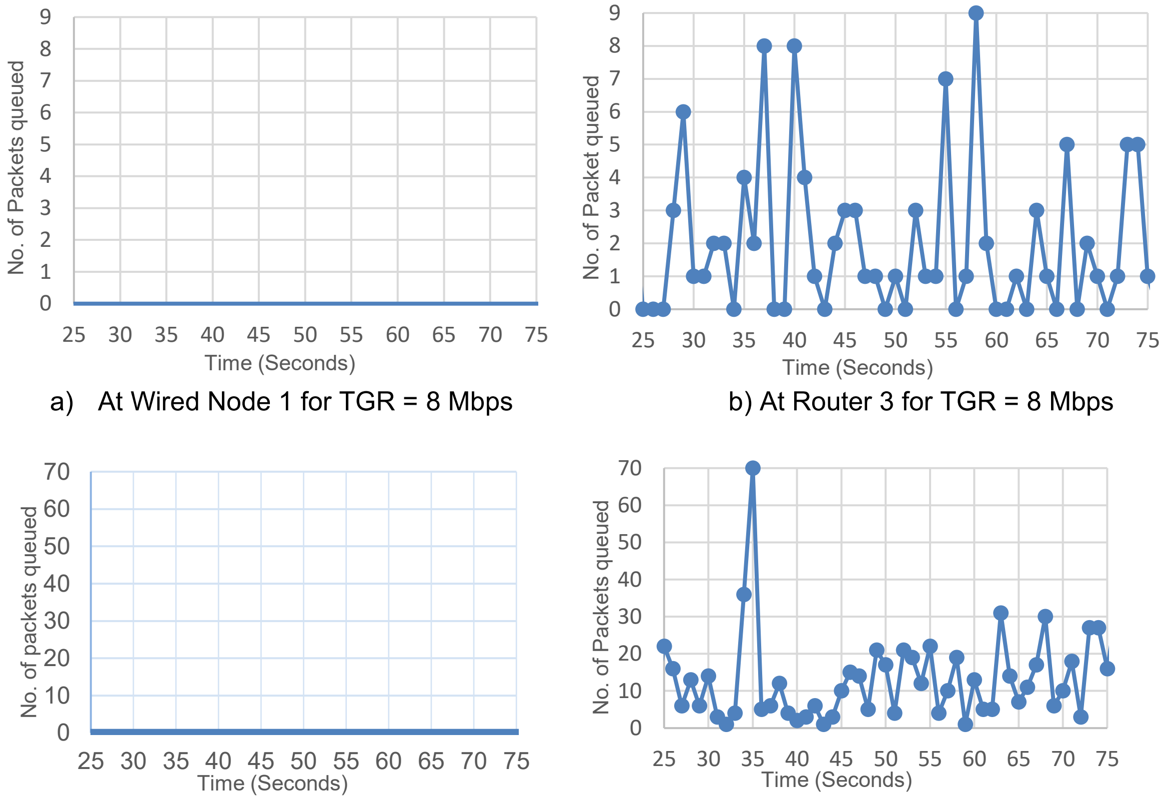


Figure 2-67: Application throughput as a function of the traffic generation rate.

Figure 2-64, 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 2-68: Queue evolution at Wired Node 1 (Link 1) and Router 3 (Link 2) for two different traffic generation rates

In Figure 2-69, 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.

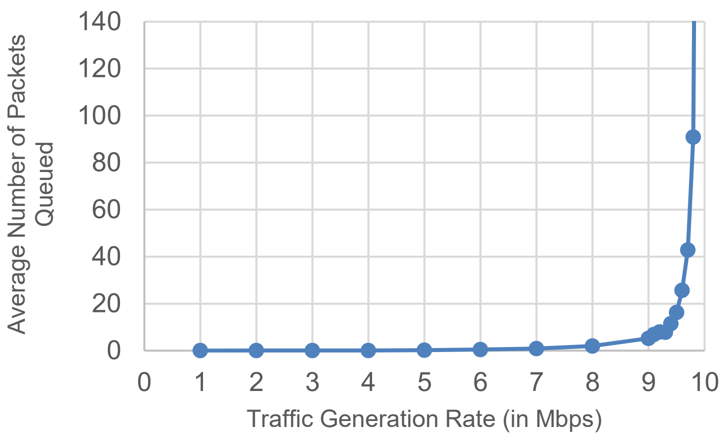


Figure 2-69: Average queue (in packets) at the bottleneck link 2 (at Router 3) as a function of the traffic generation rate

2.4.3.3.1 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 2-69, 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 $\rho = \frac{\lambda}{\mu}$.

2.4.4 Part - 2: Two Flow Scenario

We will consider a simple network setup with two flows illustrated in Figure 2-70 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).

2.4.4.1 Procedure

We will simulate the network setup illustrated in Figure 2-70 with the configuration parameters listed in detail in Table 2-25 to study the two-flow scenario. We will assume identical configuration parameters for the access links and the two application processes.

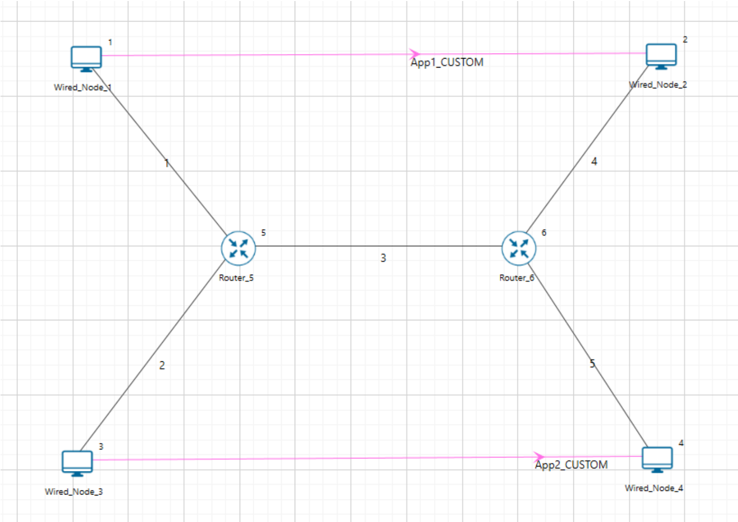


Figure 2-70: Network set up for studying two flows

Step 1: Right-click the link ID (of a wired link) and select Properties to access the link's properties window. 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 Packet trace in NetSim GUI.

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

2.4.4.2 Results

In Table 2-27, 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)	
				WN1	WN2	Router	App1	App2	App1	App2
11680	1.000	1.037	2.074	0	0	0.02	0.999	1.002	0	0
5840	2.000	2.074	4.148	0	0	0.21	1.998	2.006	0	0
3893	3.000	3.111	6.222	0	0	0.39	2.999	3.001	0	0

2920	4.000	4.147	8.295	0	0	1.88	3.996	4.018	0	0
2336	5.000	5.184	10.369	0	0	857.64	4.903	4.907	2.12	2.10
1947	5.999	6.220	12.441	0	0	4768.78	4.895	4.917	18.42	18.35
1669	6.998	7.257	14.514	0	0	5195.27	4.884	4.928	30.27	29.82
1460	8.000	8.295	16.591	0	0	5345.82	4.903	4.908	38.92	38.75
1298	8.998	9.331	18.662	0	0	5422.26	4.907	4.904	45.53	45.55
1168	10	10.369	20.738	0	0	5468.07	4.919	4.892	50.90	51.15

Table 2-27: Average queue, throughput(s) packet loss rate(s) as a function of the traffic generation

We can infer the following.

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).

2.4.5 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).

3 Routing & Switching

3.1 The OSPF weight setting problem and the performance comparison of the OSPF vs. RIP

3.1.1 Part A: The OSPF Weight Setting Problem

Routing is the task of finding a path from a sender to a desired destination. Routing is complex in large networks because of the many potential intermediate destinations a packet might traverse before reaching its destination. A routing table instructs the router how to forward packets. The routing protocols employ different operations to analyse different incoming update messages to produce their routing tables. Given a packet with an IP destination address in its header, the router performs a routing table lookup which returns the IP address of the packet's next hop

OSPF, or Open Shortest Path First, is a routing protocol used in IP networks. OSPF requires routers to exchange routing information with all other routers in the network[2]. Complete network topology knowledge, i.e. the arrangement of all routers and links in the network, is required. Because each router knows the complete topology, each router can compute all needed shortest paths.

OSPF calculates routes as follows. Each link is assigned a dimensionless metric, called cost or weight. This integer cost ranges from 1 to $65{,}535\ (= 2^{16} - 1)$. The cost of a path in the directed graph is the sum of the link costs. Using Dijkstra's shortest path algorithm, OSPF mandates that each router computes a tree of shortest paths with itself as the root.[3]

The link weights are assigned by the network operator. The lower the weight, the greater the chance that traffic will get routed on that link. One approach is to assign the OSPF metric as the inverse of the link bandwidth. On the other hand, if each link cost is set to 1, the cost of a path is equal to the number of links (hops) in the path [1].

The OSPF weight setting problem seeks a set of weights that optimizes network performance. Finding the right set of weights can greatly improve network performance. To see the importance of good OSPF weight setting, we consider the example [2] given.

[2] A network is usually divided into smaller domains. Considering each domain individually makes the network more manageable. Routing domains in today's Internet are called autonomous systems (AS). In NetSim a "network" designed is assumed to be within one AS.

[3] In the case of multiple shortest paths, OSPF will use load balancing and split the traffic flow over these multiple shortest paths. Load balancing is not currently implemented in NetSim.

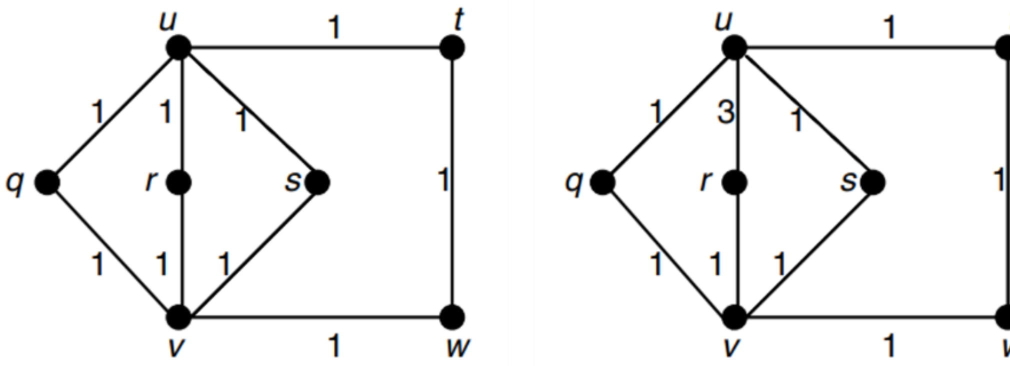


Figure 3-1: A seven-node, nine-link network is shown. All the links are assumed to be bidirectional. The capacity of each link is 100 Mbps. Nodes q, r, s, and w generate 50 Mbps of traffic each destined for node t. The weights assigned to each link are also shown.

Nodes q, r, s, and w generate 50 Mbps traffic each, for node t. The weights assigned to each link are shown beside the links. Two different weight assignments are considered. We will simulate these two scenarios in NetSim and compare their network performance.

3.1.1.1 Network Set up

Open NetSim and click on **Experiments> Internetworks> Routing and Switching> The OSPF weight setting problem and the performance comparison of the OSPF vs. RIP** then click on the tile in the middle panel to load the example as shown below.

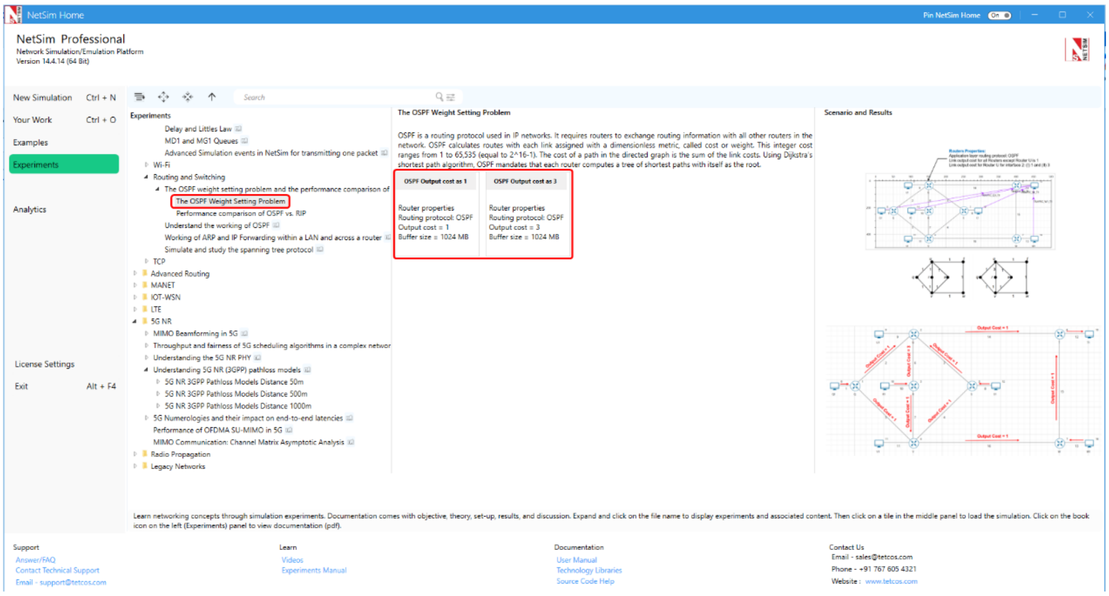


Figure 3-2: List of scenarios for the example of OSPF weight setting problem.

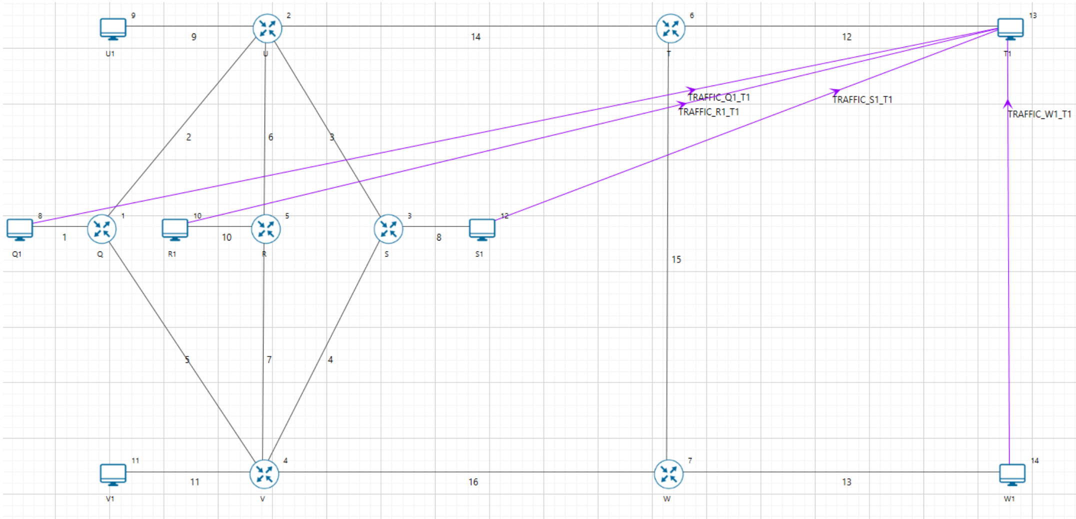


Figure 3-3: The same scenario is replicated in NetSim. Each node from the earlier figure is represented in NetSim by one router and one host. Traffic is generated from Q1, R1, S1 and W1 to T1 at a rate of 50 Mbps and all link capacities are 100 Mbps.

3.1.1.2 Procedure

Step 1: A network scenario is designed in the NetSim GUI comprising of 7 Wired Nodes and 7 Routers.

Step 2: Go to Router Properties. In the Application Layer, Routing Protocol is set as OSPF

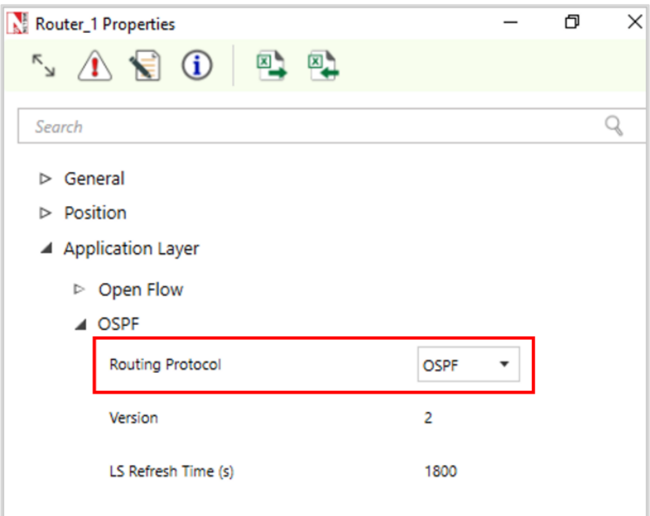


Figure 3-4: Application Layer Window - Routing Protocol is set as OSPF.

The Router Configuration Window shown above indicates the Routing Protocol set as OSPF along with its associated parameters. The “Routing Protocol” parameter is Global. i.e., changing in one Router will affect all the other Routers. So, in all the Routers, the Routing Protocol is now set as OSPF.

Step 3: Go to Router Properties > Application Layer > OSPF > Expand the Interfaces. In all the routers WAN Outgoing Interfaces, set the Output Cost is set to 1.

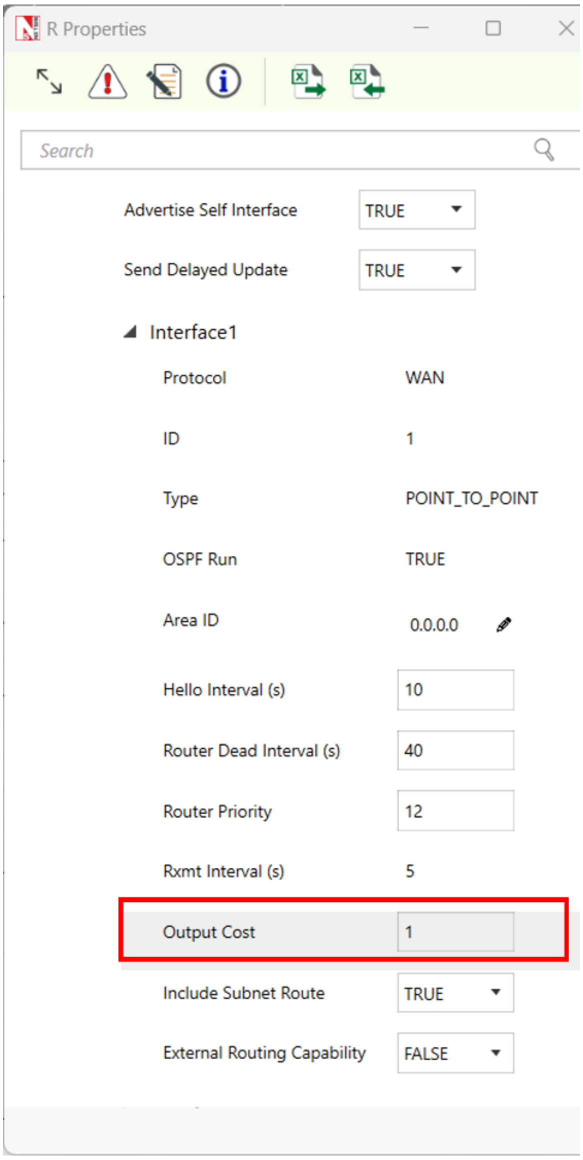


Figure 3-5: WAN Interfaces - Output Cost is set to 1.

The “Output Cost” parameter in the WAN Interface > Application Layer of a router indicates the cost of sending a data packet on that interface and is expressed in the link state metric.

Step 4: Go to Router Properties > All WAN Interfaces > Network Layer > set the Buffer Size (MB) parameter to 1024.

Step 5: Go to Link Properties, set the properties as below.

Parameter	Parameter Value
Uplink / Downlink Speed (Mbps)	100
Uplink / Downlink BER	0
Uplink / Downlink Propagation Delay (µs)	0

Table 3-1: Wired Link Properties.

Step 6: Go to the Link ID 12 Properties (Link Between Router T and Node T1) > Set Uplink and Downlink Speed to 1000 Mbps.

Step 7: To Configure an application between any two nodes by selecting a CBR application from the Set Traffic tab connect between source and destination. Right click on the Application Flow and select transport protocol to UDP.

Here, the CBR Application is generated from Q1, R1, S1 and W1 to T1 with 50Mbps of generation rate by setting the Packet Size of 1460 Bytes and Inter Arrival Time remaining $233.6\ \mu s$.

Additionally, the “Start Time (s)” parameter is set to 10s. This time is typically chosen to be greater than the time required for OSPF convergence (i.e., Exchange of OSPF information between all the routers), and it increases as the size of the network increases.

Step 8: Packet Trace is enabled, and post simulation we can observe the route which the packets have chosen to reach the destination based on the Open Shortest Path First Routing Protocol.

Step 9: Run the Simulation for 20 Seconds.

Step 10: Note down the Throughput and Observe the flow of Packet using Packet Trace.

3.1.1.3 Case 2: OSPF Output cost as 3. Network configuration:

For the same case, Change the Router R Properties > Application Layer > OSPF > Expand the Interface 1> Set the Output Cost is set to 3 and run the simulation for 20 seconds.

3.1.1.4 Results

Case-1: OSPF Output cost as 1

All link weights are set as 1.

- Shortest Path from Q1 to T1 is Q1 > Q > U > T > T1. Total cost is 2.
- Shortest Path from R1 to T1 is R1 > R > U > T > T1. Total cost is 2.
- Shortest Path from S1 to T1 is S1 > S > U > T > T1. Total cost is 2.
- Shortest Path from W1 to T1 is W1 > W > T > T1. Total cost is 1.

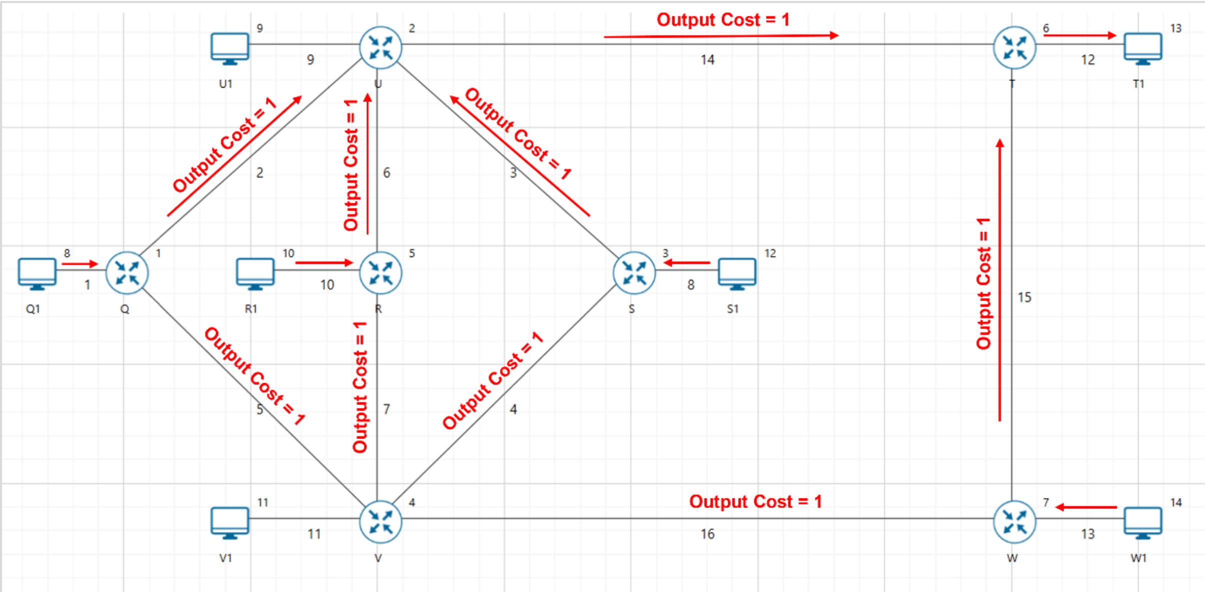
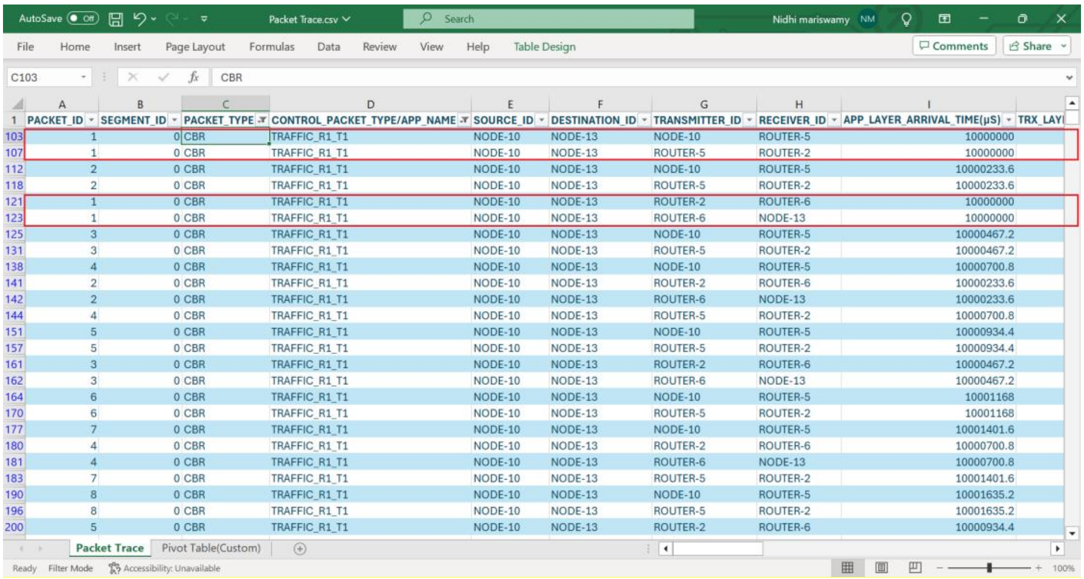


Figure 3-6: OSPF Packet Flow for Case 1, Output cost set to 1.

- The shortest paths from nodes Q1, R1, S1, and W1 to node T1 are determined based on OSPF's calculation of total cost.
- The traffic from nodes Q1, R1, and S1 is routed through node U due to its shorter path compared to the direct path to T1. The traffic from node W1 takes the direct single-hop path to T1 due to its lower cost. The packet flow described above, choosing the shortest paths, can be observed in the Packet Trace

After simulation, open packet trace, filter the PACKET TYPE column to CBR and further filter the CONTROL PACKET TYPE/APP NAME to respective traffic flow and observe the packet flow. Here we have filtered the CONTROL PACKET TYPE/APP NAME as TRAFFIC_R1_T1



1	PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)
103	1	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10000000
107	1	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-2	10000000
112	2	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10000233.6
118	2	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-2	10000233.6
121	1	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-2	ROUTER-6	10000000
123	1	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-6	NODE-13	10000000
125	3	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10000467.2
131	3	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-2	10000467.2
138	4	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10000700.8
141	2	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-2	ROUTER-6	10000233.6
142	2	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-6	NODE-13	10000233.6
144	4	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-2	10000700.8
151	5	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10000934.4
157	5	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-2	10000934.4
161	3	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-2	ROUTER-6	10000467.2
162	3	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-6	NODE-13	10000467.2
164	6	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10001168
170	6	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-2	10001168
177	7	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10001401.6
180	4	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-2	ROUTER-6	10000700.8
181	4	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-6	NODE-13	10000700.8
183	7	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-2	10001401.6
190	8	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10001635.2
196	8	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-2	10001635.2
200	5	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-2	ROUTER-6	10000934.4

Figure 3-7: Packet flow in packet trace for OSPF output cost as 1.

Case-2: OSPF Output cost as 3

Link-6 cost is set to 3, all other link weights are 1.

- Increasing the output cost on link R→ U to 3 results in different shortest paths chosen by OSPF for some nodes compared to Case 1.
- For instance, in Case 2, node R1 chooses a different path through nodes V and W to reach T1 due to the increased cost on the direct link R → U.
- This weight setting aims to balance the load on links by redistributing traffic, ensuring de-congestion of link U → T.

Users can observe the shortest paths from NetSim Packet Trace.

- Shortest Path from Q1 to T1 is Q1 > Q > U > T > T1. Total cost is 2.
- Shortest Path from R1 to T1 is R1 > V > W > T > T1. Total cost is 3.
- Shortest Path from S1 to T1 is S1 > S > U → T > T1. Total cost is 2.
- Shortest Path from W1 to T1 is W1 > W > T > T1. Total cost is 1.

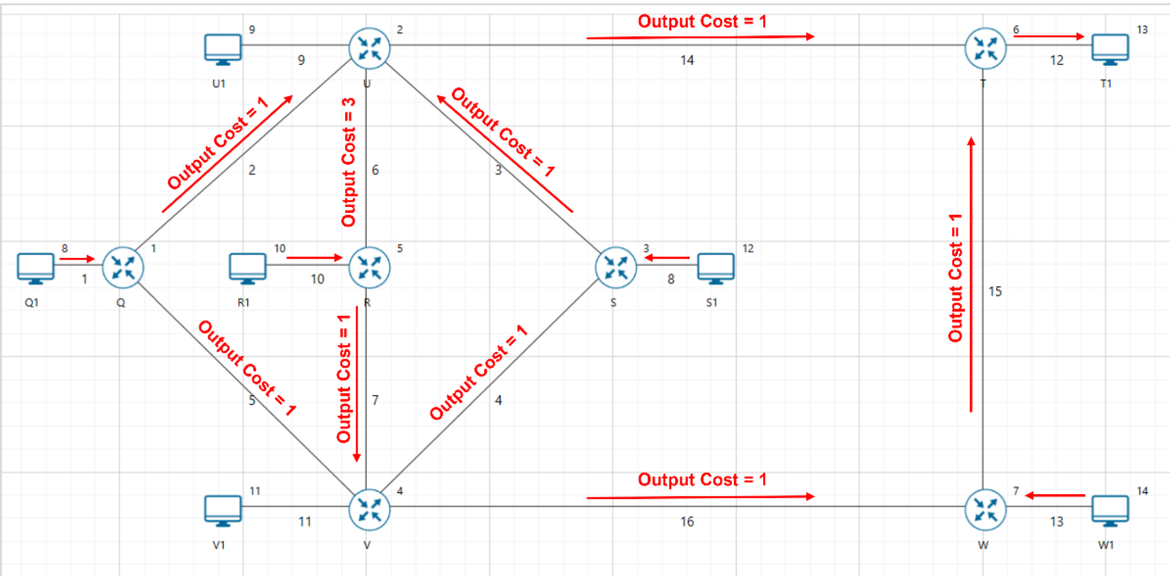
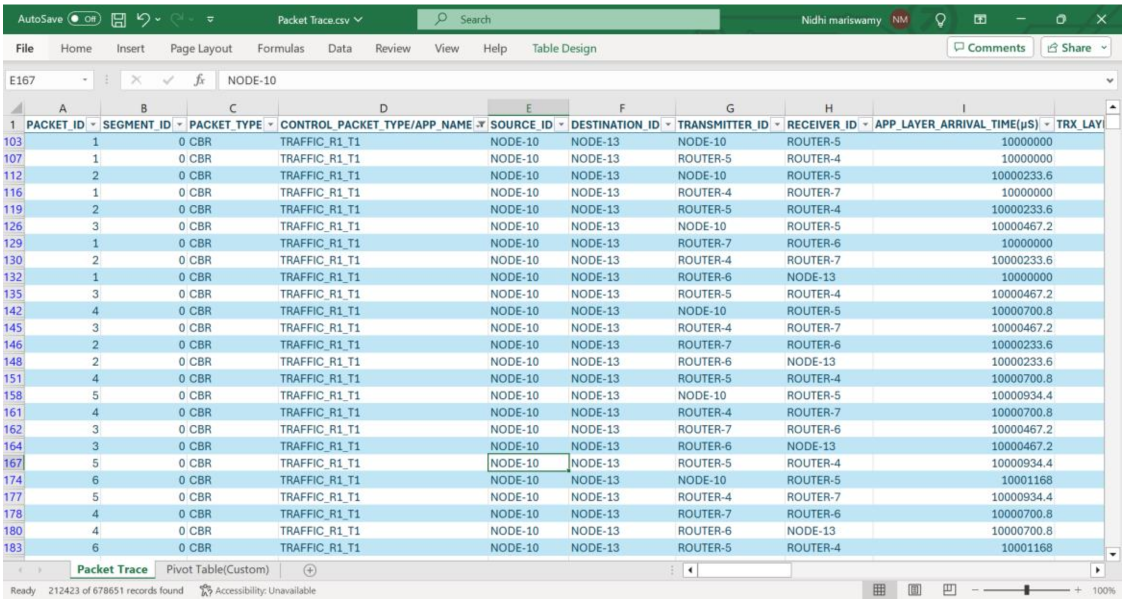


Figure 3-8: OSPF Packet Flow for Case 2, Output cost set to 3 in Interface 1 of Router R.

After simulation, open packet trace, filter the CONTROL PACKET TYPE/APP NAME as TRAFFIC_R1_T1 and observe the change the packet flow due to changes in output cost.



	PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)	TRX_LAYI
103	1	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10000000	
107	1	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-4	10000000	
112	2	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10000233.6	
116	1	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-4	ROUTER-7	10000000	
119	2	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-4	10000233.6	
126	3	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10000467.2	
129	1	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-7	ROUTER-6	10000000	
130	2	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-4	ROUTER-7	10000233.6	
132	1	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-6	NODE-13	10000000	
135	3	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-4	10000467.2	
142	4	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10000700.8	
145	3	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-4	ROUTER-7	10000467.2	
146	2	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-7	ROUTER-6	10000233.6	
148	2	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-6	NODE-13	10000233.6	
151	4	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-4	10000700.8	
158	5	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10000934.4	
161	4	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-4	ROUTER-7	10000700.8	
162	3	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-7	ROUTER-6	10000467.2	
164	3	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-6	NODE-13	10000467.2	
167	5	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-4	10000934.4	
174	6	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	NODE-10	ROUTER-5	10001168	
177	5	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-4	ROUTER-7	10000934.4	
178	4	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-7	ROUTER-6	10000700.8	
180	4	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-6	NODE-13	10000700.8	
183	6	0	CBR	TRAFFIC_R1_T1	NODE-10	NODE-13	ROUTER-5	ROUTER-4	10001168	

Figure 3-9: Packet flow in packet trace for OSPF output cost as 3.

3.1.1.5 Discussion

Scenario	Throughput in Mbps (Q1 -T1)	Throughput in Mbps (R1 -T1)	Throughput in Mbps (S1 -T1)	Throughput in Mbps (W1 -T1)
Case 1	32.71	32.71	32.70	49.99
Case 2	49.06	49.06	49.06	49.06

Table 3-2: NetSim simulation outputs from both cases. Recall that the traffic demand (generation rate) at each node is ≈50 Mbps. In case #1 the three flows Q1-T1, R1-T1 and S1-T1 obtain only 32.71 Mbps. In case #2 with the right OPSF weight settings, each traffic.

The traffic generation rate at each node was approximately 50 Mbps. In case #1, the three flows—Q1-T1, R1-T1, and S1-T1—achieved a throughput of only 32.71 Mbps. This indicates a significant underperformance relative to the traffic demand. This happens because the link U → T has a capacity of 100 Mbps while the input rate is 150 Mbps. The available bandwidth is split equally between the three flows with each flow getting approximately $\frac{100}{3}$ Mbps.

In case #2, after optimizing OSPF weight configurations, traffic from R is routed to T through V and W. The link U → T now carries two flows totaling 100 Mbps and the link W → T carries the other two flows also totaling 100 Mbps. The network is now able to handle the traffic demands. This shows how setting the right link weights in OSPF can greatly improve network performance.

3.1.2 Part-B: Performance comparison of OSPF vs. RIP

Routing information protocol (RIP) is a routing protocol based on the Bellman-Ford (or distance vector) algorithm. This algorithm has been used for routing computations in computer networks since the early days of the ARPANET. Distance vector algorithms are based on a table in each router listing the best route to every destination in the system. Of course, in order to define which route is best, we have to have some way of measuring goodness. This is referred to as the "metric". RIP uses a metric that simply counts how many routers a message must go through i.e., to get the metric of a complete route, one just adds up the costs of the individual hops that make up the route [3]. The best route is the route with the minimum hops; RIP therefore is a minimum hop routing algorithm.

The advantage of min hop routing is that a demand from a source to the corresponding destination can be routed through the network while consuming the least amount of total bandwidth resources. If a demand requires an amount of bandwidth d, then the total bandwidth consumed on a route is $d \times H$ where H is the number of hops on the chosen route. If H_{min} is the number of hops on the shortest path, then $H_{min} \leq H$, and it follows that min hop routing consumes the least resources. However, the amount of resources consumed in routing a demand is often not the only important criterion to be considered. We show this through the example below and compare the performance of RIP with OSPF.

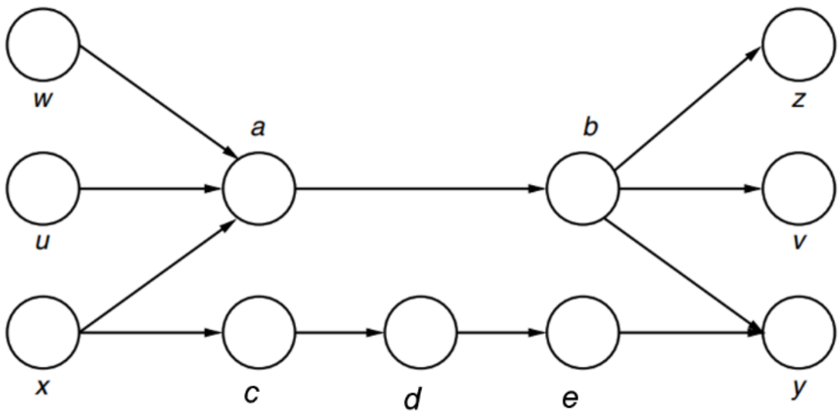


Figure 3-10: Example showing possible problems with min hop routing. Suppose each link has a bandwidth of 100 Mbps unit. Let us say W is sending 50 Mbps of traffic to z, and u is sending 50 Mbps of data to v. We now have a traffic demand between x and y, asking for 50 unit of bandwidth. Rather than route the packet through c – d – e, RIP uses min hop to route the traffic, i.e., traffic is sent through the link a – b, which is already congested. This problem can be overcome in OSPF by setting the right link weights (which we discussed in Part A).Network setup.

3.1.2.1 Network setup

Open NetSim and click on **Experiments> Internetworks> Routing and Switching> The OSPF weight setting problem and the performance comparison of the OSPF vs. RIP** then click on the tile in the middle panel to load the example as shown below.

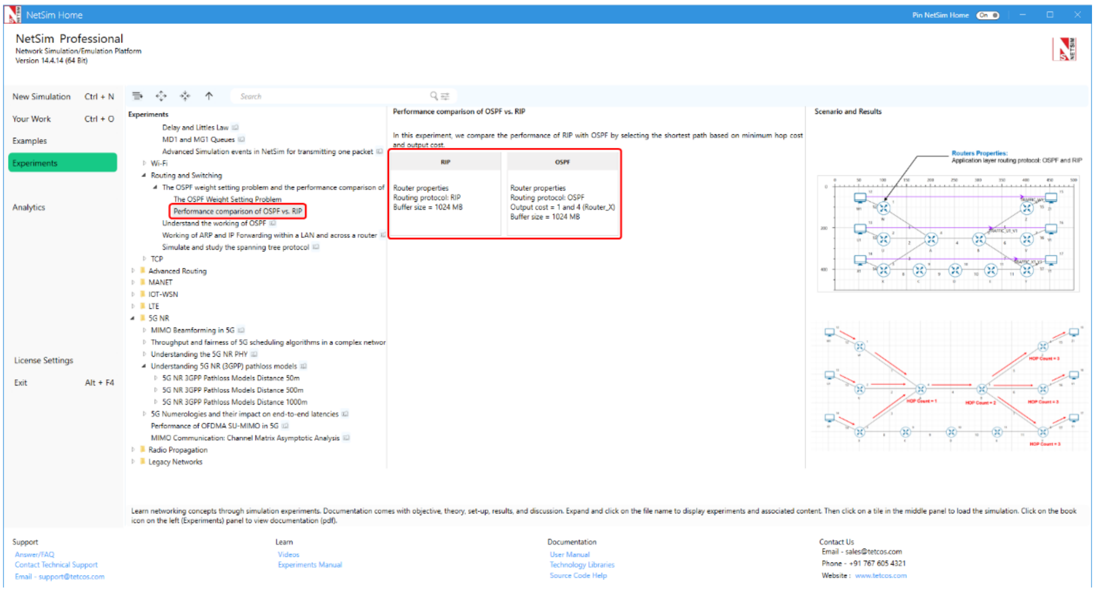


Figure 3-11: List of scenarios for the example the performance comparison of the OSPF vs. RIP

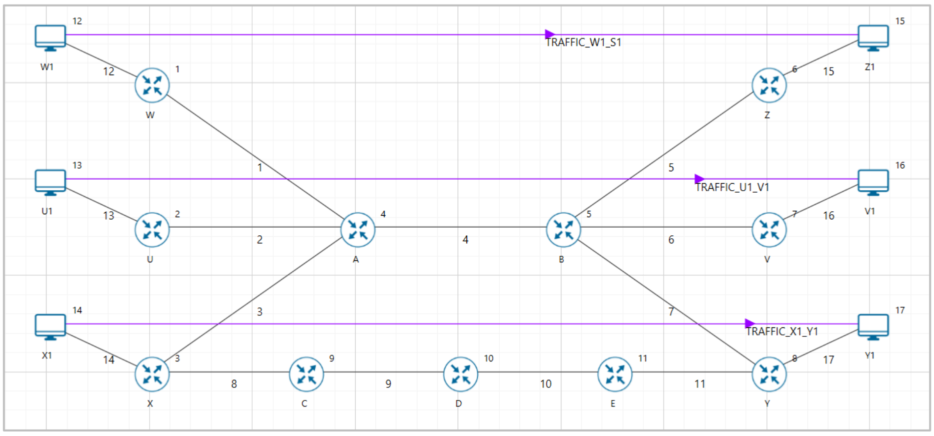


Figure 3-12: NetSim scenario consists of 11 Routers and 6 Wired Nodes. CBR Traffic is generated from W1 to Z1, U1 to V1, X1 to Y1 at 50 Mbps rate and link capacity is 100 Mbps.

3.1.2.2 Case 1: RIP. Network configuration

Step 1: Design a network in the NetSim GUI comprising of 6 Wired Nodes and 11 Routers

Step 2: Go to Router Properties. In the Application Layer, Routing Protocol is set as RIP.

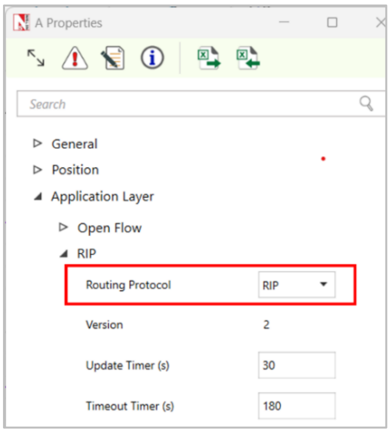


Figure 3-13: Application Layer Window - Routing Protocol is set as RIP.

The “Routing Protocol” parameter is Global. i.e., changing in one Router will update this parameter in all other Routers.

Step 3: Go to Router Properties > All WAN Interfaces > Network Layer > set the Buffer Size (MB) parameter to 1024.

Step 4: Go to Link Properties, set the properties as below.

Parameter	Parameter Value
Uplink / Downlink Speed (Mbps)	100
Uplink / Downlink BER	0
Uplink / Downlink Propagation Delay (µs)	0

Table 3-3: Wired Link Properties.

Step 5: To Configure an application between any two nodes by selecting a CBR application from the Set Traffic tab connect between source and destination. Right click on the Application Flow and select transport protocol to UDP.

The CBR Application is created with a packet size of 1460 bytes and an inter-arrival time of $233.6\ \mu s$. Additionally, the "Start Time (s)" parameter is set to 10 while configuring the application. This time is typically set to be greater than the time taken for route convergence, and it increases as the size of the network increases.

Step 6: Packet Trace is enabled, and post-simulation, we can observe the route that the packets have chosen to reach the destination based on the Routing Protocol set.

Step 7: Run the Simulation for 20 Seconds.

Step 8: Note down the Throughput and observe the flow of packets using Packet Trace.

3.1.2.3 Case 2: OSPF. Network configuration

For the same case, change the setting as follows

Step 1: Go to Router Properties. In the Application Layer, Routing Protocol is set as OSPF.

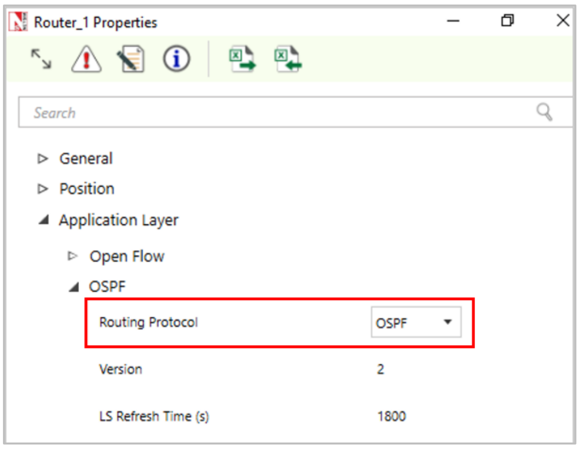


Figure 3-14: Application Layer Window - Routing Protocol is set as OSPF.

Step 2: Go to Router Properties > Application Layer > OSPF > Expand the Interfaces. In all the routers WAN Outgoing Interfaces, set the "Output Cost" is set to 1 as shown below.

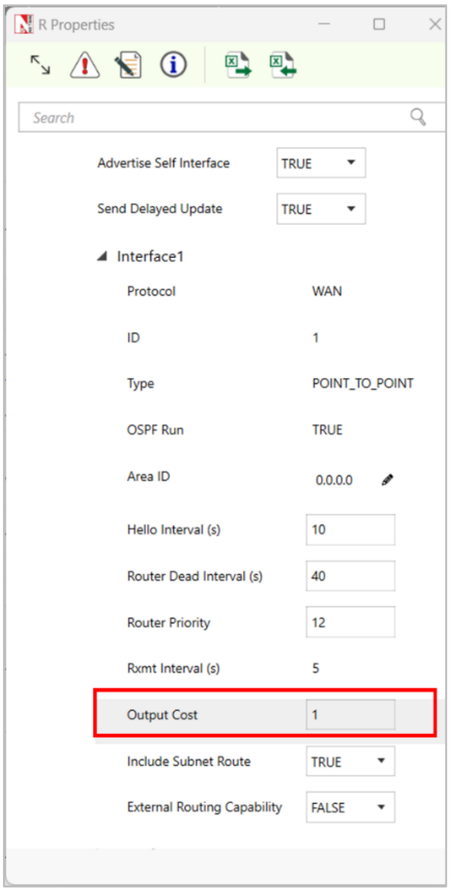


Figure 3-15: WAN Interfaces - Output Cost is set to 1.

Step 3: Go to Router X Properties > Application Layer > OSPF > Expand the Interface 1 > Set the Output Cost is set to 4.

Step 4: Run the Simulation for 20 Seconds.

Step 5: Note down the Throughput and observe the flow of packets using Packet Trace.

3.1.2.4 Results

Case-1: RIP

Users can observe from the packet trace that:

- Shortest Path from W1 to Z1 is W1 > W > A > B > Z > Z1. Hop count is 3.
- Shortest Path from U1 to V1 is U1 > U > A > B > V > V1. Hop count is 3.
- Shortest Path from X1 to Y1 is X1 > X > A > B > Y > Y1. Hop count is 3.

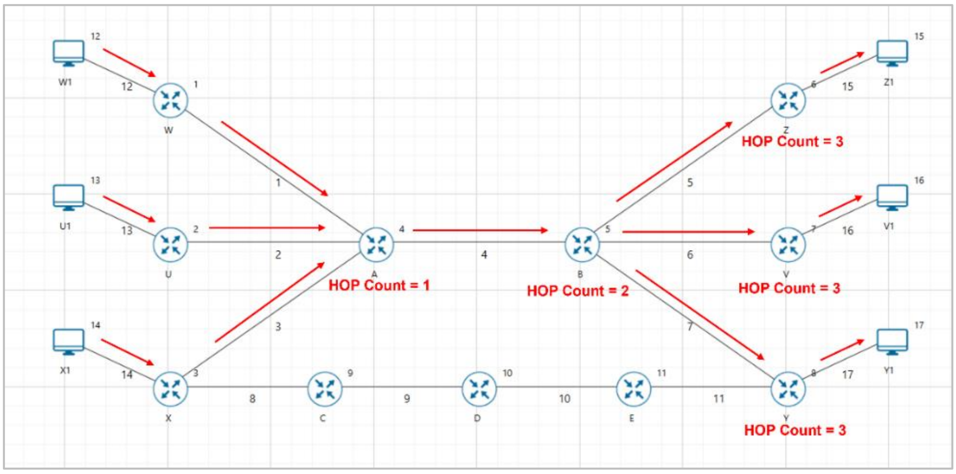
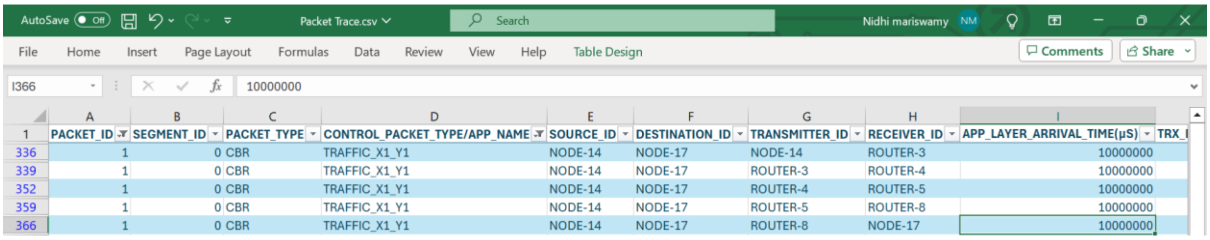


Figure 3-16: Packet flow when routing protocol is set to RIP.

- The shortest paths from nodes W1, U1 and X1 to nodes Z1, V1, and Y1 are determined based on hop-count. This is because the protocol in operation is RIP.
- Min hop routing would use the three-hop path X → A → B → Y. Then the link A → B is completely full, and, as far as servicing future demands is concerned, the network becomes partitioned.
- It would have been better to route the demand on the four-hop path even though it consumes more resources. The point is that resources on the links that constitute the four-hop path between x and y cannot be used because of HOP Count limitations in RIP protocol.

After simulation, open packet trace, filter the PACKET TYPE column to CBR and further filter the CONTROL PACKET TYPE/APP NAME to respective traffic flow and observe the packet flow. Here we have filtered the CONTROL PACKET TYPE/APP NAME as TRAFFIC_X1_Y1 and PACKET ID to 1.



	PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)	TRX_I
336	1	0	CBR	TRAFFIC_X1_Y1	NODE-14	NODE-17	NODE-14	ROUTER-3	10000000	
339	1	0	CBR	TRAFFIC_X1_Y1	NODE-14	NODE-17	ROUTER-3	ROUTER-4	10000000	
352	1	0	CBR	TRAFFIC_X1_Y1	NODE-14	NODE-17	ROUTER-4	ROUTER-5	10000000	
359	1	0	CBR	TRAFFIC_X1_Y1	NODE-14	NODE-17	ROUTER-5	ROUTER-8	10000000	
366	1	0	CBR	TRAFFIC_X1_Y1	NODE-14	NODE-17	ROUTER-8	NODE-17	10000000	

Figure 3-17: Shows the packet flow for Traffic X1-Y1 in the packet trace after applying the filters for RIP case

Case 2: OSPF

Users can observe from the Packet Trace that:

- Shortest Path from W1 to Z1 is W1 > W > A > B > Z > Z1. Total cost is 3.
- Shortest Path from U1 to V1 is U1 > U > A > B > V > V1. Total cost is 3.
- Shortest Path from X1 to Y1 is X1 > X > C > D > E > Y > Y1. Total cost is 4. Here, Packet flow from X1 to Y1 is not happening via X1 > X > A> B > Y > Y1 because the total output cost for is 6. Hence, it chooses the path with least output cost.

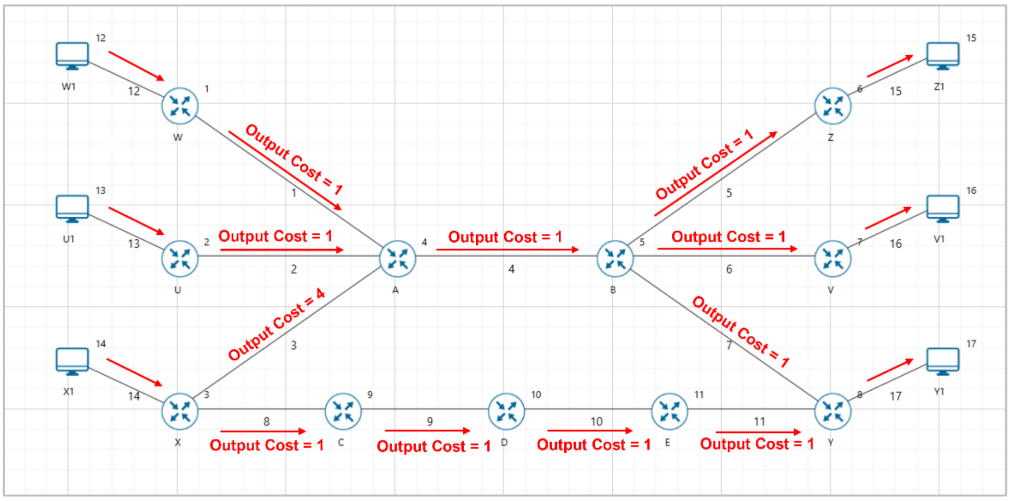
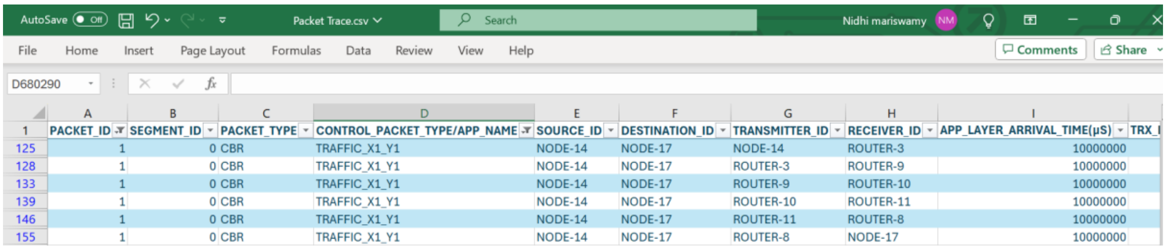


Figure 3-18: Packet flow when routing protocol is set to OSPF.

- The OSPF link output cost is set to 4 between X > A.
- The shortest paths from nodes W1, U1,and X1 to nodes Z1, V1, and Y1 are determined based on OSPF's calculation of total cost, which considers factors like link weights.

After simulation, open packet trace, filter the PACKET TYPE column to CBR and further filter the CONTROL PACKET TYPE/APP NAME to respective traffic flow and observe the packet flow. Here we have filtered the CONTROL PACKET TYPE/APP NAME as TRAFFIC_X1_Y1 and PACKET ID to 1.



PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)
1	0	CBR	TRAFFIC_X1_Y1	NODE-14	NODE-17	NODE-14	ROUTER-3	10000000
1	0	CBR	TRAFFIC_X1_Y1	NODE-14	NODE-17	ROUTER-3	ROUTER-9	10000000
1	0	CBR	TRAFFIC_X1_Y1	NODE-14	NODE-17	ROUTER-9	ROUTER-10	10000000
1	0	CBR	TRAFFIC_X1_Y1	NODE-14	NODE-17	ROUTER-10	ROUTER-11	10000000
1	0	CBR	TRAFFIC_X1_Y1	NODE-14	NODE-17	ROUTER-11	ROUTER-8	10000000
1	0	CBR	TRAFFIC_X1_Y1	NODE-14	NODE-17	ROUTER-8	NODE-17	10000000

Figure 3-19: Shows the packet flow for Traffic X1-Y1 in the packet trace after applying the filters for OSPF case.

3.1.2.5 Discussion

Scenario	Throughput in Mbps (W1-Z1)	Throughput in Mbps (U1 -V1)	Throughput in Mbps (X1 -Y1)
Case 1 (RIP)	32.71	32.70	32.70
Case 2 (OSPF)	49.06	49.06	49.99

Table 3-4: NetSim simulation outputs from both cases. Recall that the traffic demand (generation rate) at each node is ≈50 Mbps. In case #1, with RIP, the three flows W1-S1, U1-V1 and X1-Y1 obtain a throughput of only 32.70 Mbps. In case #2, with OPSF, each traffic demand gets the required throughput of ≈ 50 Mbps.

Here again, the traffic generation rate at each node was ≈ 50 Mbps. In case #1 with RIP, the three flows—W1-Z1, U1-V1, and X1-Y1—achieve a throughput of only 32.70 Mbps. This shows a significant underperformance relative to the traffic demand. Conversely, in case #2, with OSPF each flow successfully meets its expected throughput demand.

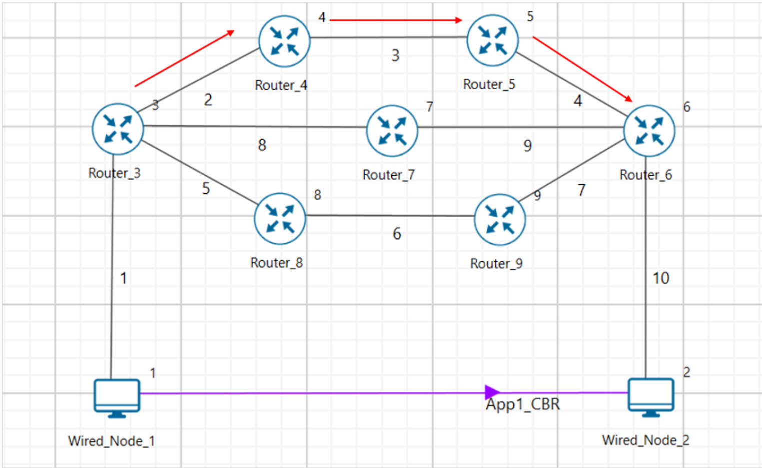
The reason for this difference is that RIP uses hop count as a metric forcing the X1–V1 flow to use the same congested A-B link. However, OSPF provides flexibility in setting the link weights. When we set the X-A link weight as 4, the X1-V1 traffic demand flows through the uncongested path X-C-D-E-Y. Thus the 100 Mbps A-B link is able to provide ≈ 50 Mbps of throughput to W1-Z1 and U1-V1 flows, and the other path (X-C-D-E-Y) provides ≈ 50 Mbps of throughput to the X1-Y1 flow.

3.1.2.6 Exercises

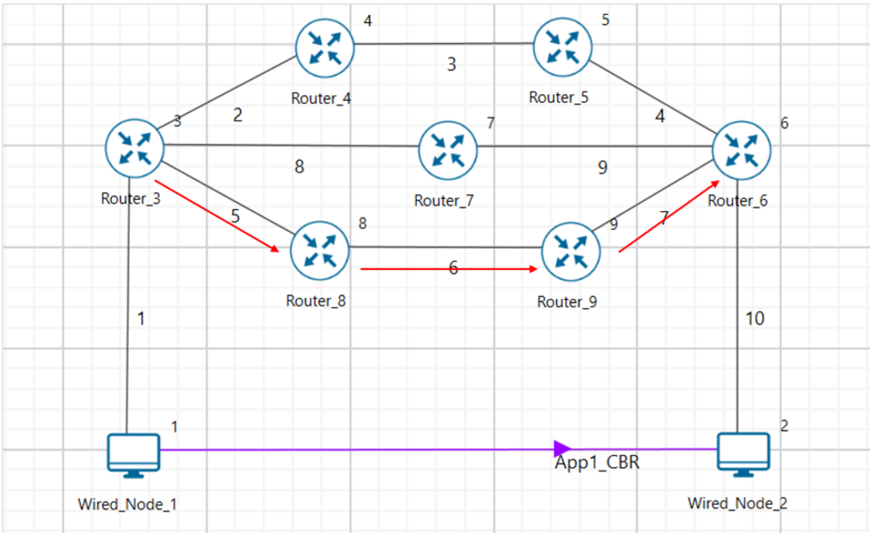
1. Effect of Output Cost on OSPF Routing Path Selection

Construct the network as shown below, using OSPF as the application layer routing protocol with the default output cost set to 100. Set the application start time to 10 seconds and enable packet trace to observe the data flow in the trace file after the simulation. Analyse how modifying the output cost in routers influences OSPF's path selection for data transmission

Case a: In the scenario below, OSPF by default, select the Router 3 > Router 7 > Router 6 path for data transmission due to its lower cost. Your task is to modify the OSPF link costs in router so that the data follows the Router 3 > Router 4 > Router 5 > Router 6 for data transmission

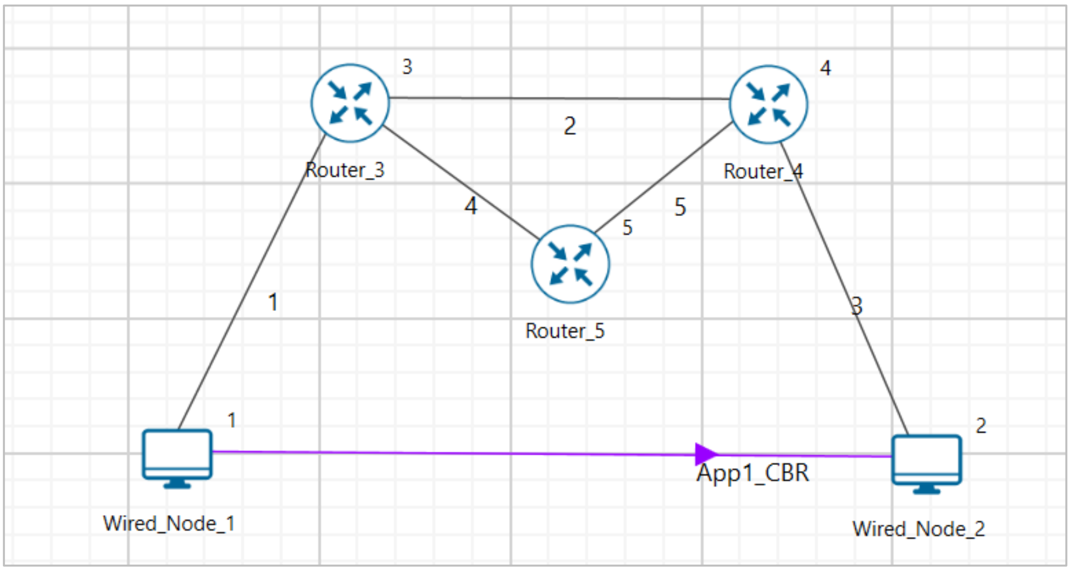


Case b: Similarly, configure the output cost in router link such that, the data should consider the path as shown below: Router 3 > Router 8 > Router 9 > Router 6.



2. **Understanding OSPF Rerouting After Link Failure**

   Construct the network scenario as shown below and configure OSPF as the default routing protocol. Introduce a link failure at 25 seconds on Link 2, and analyse how OSPF detects the failure, recomputes the routes, and selects an alternate path for data transmission. Enable packet trace prior to simulation and explain your observations with relevant screenshots.



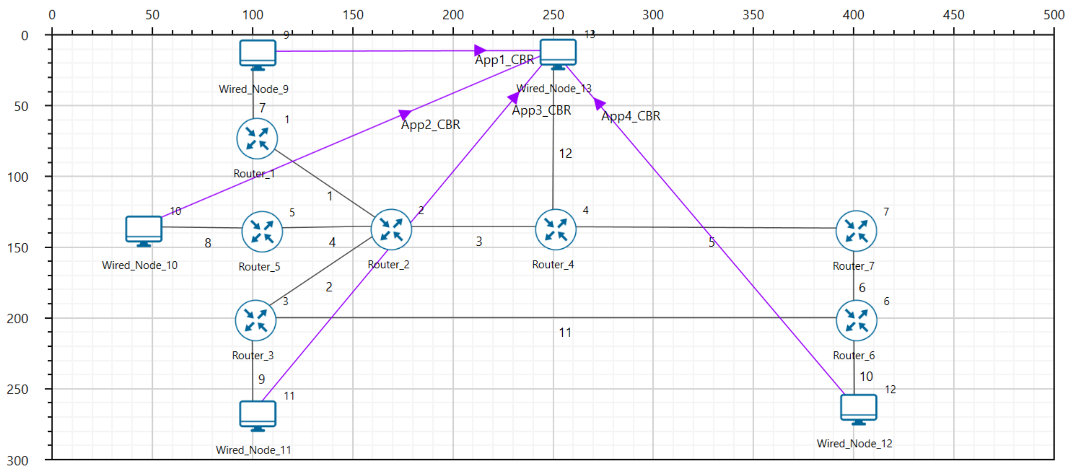
Network settings	
Router Properties: Router > Application layer	
Routing protocol	OSPF
Wired link properties: Link 2	
Up time (sec)	0
Down time (sec)	25
Application properties	
Start time	15 sec
Run time	
Simulation time	50 seconds

3. Understanding OSPF weight setting problem

Construct the network scenario as shown below, using OSPF as the routing protocol at the application layer. In this setup, generate 50 Mbps of traffic from Nodes 9, 10, 11, and 12 to Node 13. Set the output cost for all router links to 100, and follow the additional settings provided in the table below.

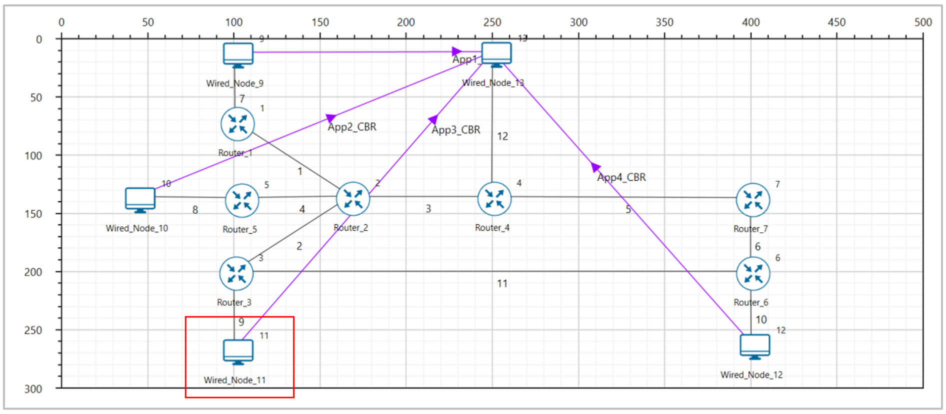
Case a: In this scenario, Nodes 9, 10, and 11 are using the same path for data transmission, leading to network congestion. Your tasks are to: (i) obtain the transmission path for each application from the packet trace, (ii) tabulate the throughput for each application, and (iii) highlight the congested links in the network.

Enable packet trace prior to simulation and explain your observations with relevant screenshots



Network settings	
Router Properties: Router > Application layer	
Application layer routing protocol	OSPF
Output costs for all routers	100
Router > Interface (WAN) > Network layer	
Buffer size	1024
All Link Properties	
Uplink / Downlink Speed (Mbps)	100 (Link 12: 1000)
Uplink / Downlink BER	0
Uplink / Downlink Propagation Delay (µs)	0
Application properties	
Start time (seconds)	10
Packet size (Bytes)	1460
Inter arrival time (µs)	233.6
Transport layer protocol	UDP
Run time	
Simulation time (seconds)	30

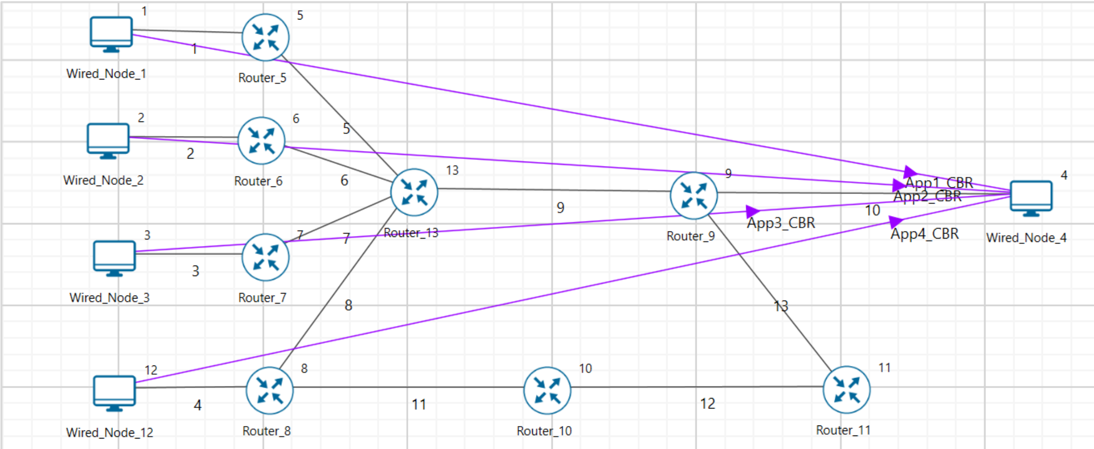
Case b: For the same scenario, adjust the weight (output cost) so that Node 11 takes an alternate path for data transmission, reducing the traffic load on the original path. The throughput should improve for all nodes, and network congestion should be minimized. (i) Tabulate the throughput obtained for each node and compare it with Case a. (ii) Explain the alternate path taken Node 11 with relevant screenshots from packet trace.



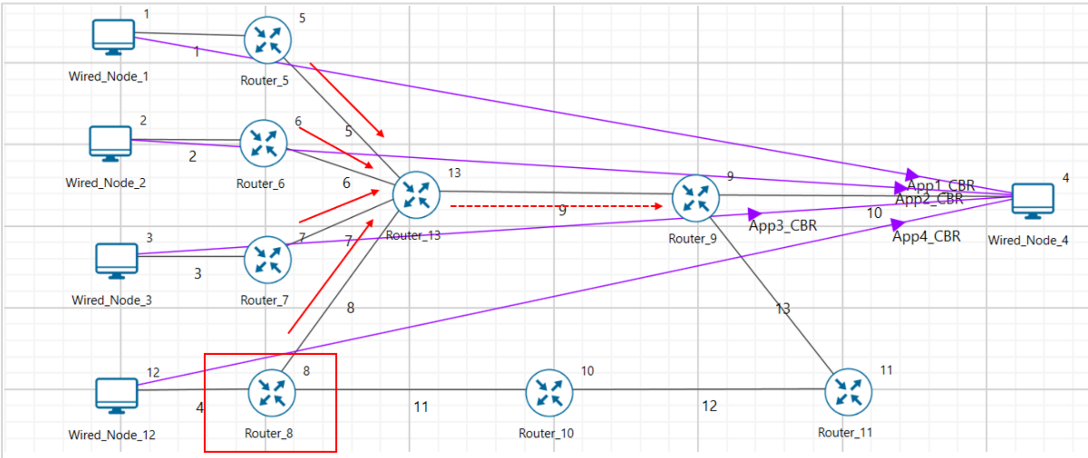
4. Understanding performance evaluation between OSPF and RIP protocol.

Construct the scenario as shown below and set the application layer routing protocol to RIP and Generate traffic from each node to Node 4 at a rate of 50 Mbps, set start time to 10 sec and simulate it for 30 seconds. Enable packet trace prior to simulation and explain your observations with relevant screenshots

Case a: In this case, configure the network setting as mentioned, tabulate the throughput for each application, analyse any congestion point in network.



Case b: Consider OSPF protocol and set the output cost for all routers to 100, adjust the weight (Output cost) for Router 8, in such way that it considers alternate path for data transmission and avoiding the congestion path. Explain how the OSPF has flexibility improving network performance by weight setting over RIP protocol.



3.1.2.7 References

[1] M. Ericsson, M. G. C. Resende and P. M. Pardalos, “A genetic algorithm for the weight setting problem in OSPF routing,” *Journal of Combinatorial Optimization* , vol. 6, p. 299–333, 2002.

[2] A. Kumar, D. Manjunath and J. Kuri, Communication Networking, ISBN: 0-12-428751-4, 2004.

[3] “RFC 2543: RIP Version 2,” IETF, [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2453#page-3.

3.2 Understand working of ARP and IP Forwarding within a LAN and across a router (Level 1)

3.2.1 Theory

In network architecture different layers have their own addressing scheme. This helps the different layers in being largely independent. Application layer uses host names, network layer uses IP addresses, and the link layer uses MAC addresses. Whenever a source node wants to send an IP datagram to a destination node, it needs to know the address of the destination. Since there are both IP addresses and MAC addresses, there needs to be a translation between them. This translation is handled by the Address Resolution Protocol (ARP). In IP network, IP routing involves the determination of suitable path for a network packet from a source to its destination. If the destination address is not on the local network, routers forward the packets to the next adjacent network.

(**Reference:** A good reference for this topic is Section 5.4.1: Link Layer Addressing and ARP, of the book, Computer Networking, A Top-Down Approach, 6th Edition by Kurose and Ross)

3.2.2 ARP protocol Description

1. ARP module in the sending host takes any IP address as input and returns the corresponding MAC address.
2. First the sender constructs a special packet called an ARP packet, which contains several fields including the sending and receiving IP and MAC addresses.
3. Both ARP request and response packets have the same format.
4. The purpose of the ARP request packet is to query all the other hosts and routers on the subnet to determine the MAC address corresponding to the IP address that is being resolved.
5. The sender broadcasts the ARP request packet, which is received by all the hosts in the subnet.
6. Each node checks if its IP address matches the destination IP address in the ARP packet.
7. The one with the match sends back to the querying host a response ARP packet with the desired mapping.
8. Each host and router have an ARP table in its memory, which contains mapping of IP addresses to MAC addresses.
9. The ARP table also contains a Time-to-live (TTL) value, which indicates when each mapping will be deleted from the table.

3.2.3 ARP Frame Format

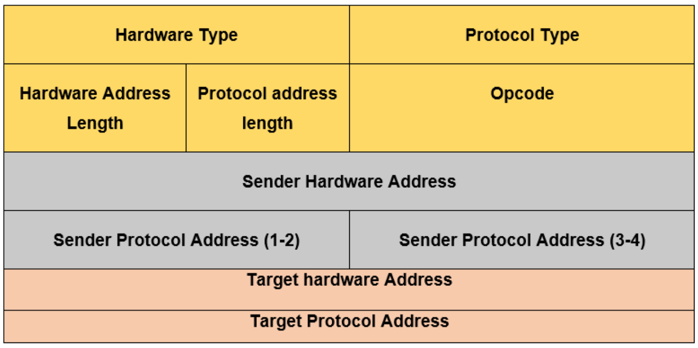


Figure 3-20: ARP Frame Format

The ARP message format is designed to accommodate layer two and layer three addresses of various sizes. This diagram shows the most common implementation, which uses 32 bits for the layer three ("Protocol") addresses, and 48 bits for the layer two hardware addresses.

3.2.4 IP Forwarding Description

1. Every router has a forwarding table that maps the destination addresses (or portions of the destination addresses) to that router's outbound links.
2. A router forwards a packet by examining the value of a field in the arriving packet's header, and then using this header value to index into the router's forwarding table.
3. The value stored in the forwarding table entry for that header indicates the router's outgoing link interface to which that packet is to be forwarded.
4. Depending on the network-layer protocol, the header value could be the destination address of the packet or an indication of the connection to which the packet belongs.
5. ARP operates when a host wants to send a datagram to another host on the same subnet.
6. When sending a Datagram off the subnet, the datagram must first be sent to the first-hop router on the path to the final destination. The MAC address of the router interface is acquired using ARP.
7. The router determines the interface on which the datagram is to be forwarded by consulting its forwarding table.
8. Router obtains the MAC address of the destination node using ARP.
9. The router sends the packet into the respective subnet from the interface that was identified using the forwarding table.

3.2.5 Network Set up

Open NetSim and click on **Experiments> Internetworks> Routing and Switching > Working of ARP and IP Forwarding within a LAN and across a router** then click on the tile in the middle panel to load the as shown in example see Figure 3-21.

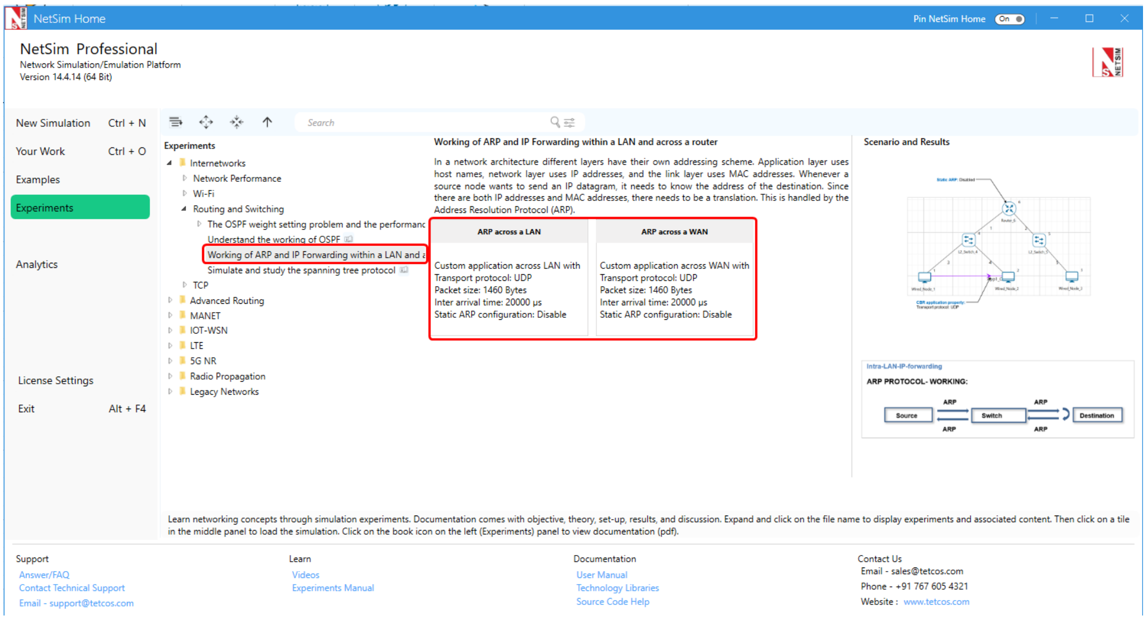


Figure 3-21: List of scenarios for the example of Working of ARP and IP Forwarding within a LAN and across a router.

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 3-22.

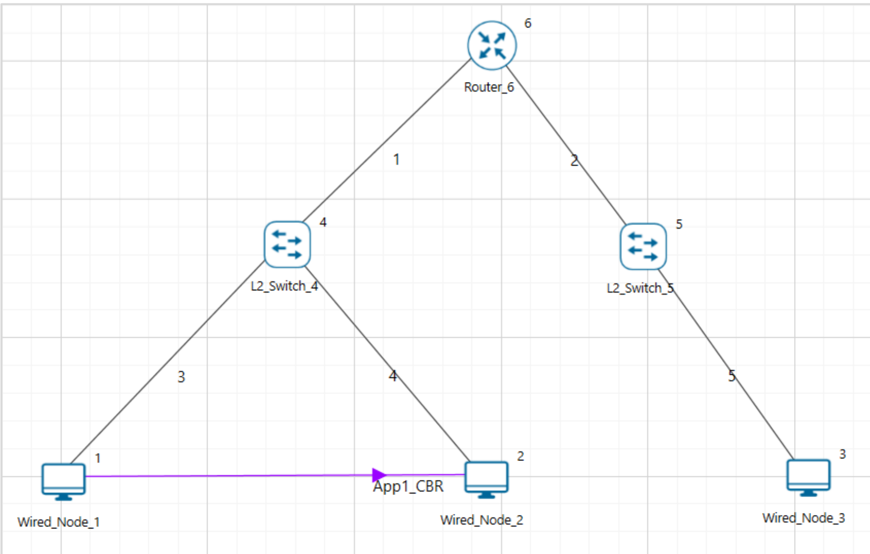


Figure 3-22: Network set up for studying the ARP across a LAN

3.2.6 Procedure

ARP across a LAN

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

Step 1: A network scenario is designed in NetSim GUI comprising of 3 Wired Nodes, 2 L2 Switches, and 1 Router in the **"Internetworks"** Network Library.

Step 2: Configure an application between any two nodes by selecting a CBR application from the Set Traffic tab. Click on the created application, expand the application property panel on the right, and set the transport protocol to UDP instead of TCP by keeping other properties as default.

If set to TCP, the ARP table will get updated due to the transmission of TCP control packets thereby eliminating the need for ARP to resolve addresses.

Step 3: Packet Trace is enabled from Configure report tab, and hence we can view the ARP Request and ARP Reply packets exchanged initially, before transmission of the data packets.

Step 4: Click on Run simulation. The simulation time is set to 10 seconds. In the **"Static ARP Configuration"** tab, Static ARP is set to disable see Figure 3-23.

Step 5: Under Options, the "Static ARP" tab, Static ARP is set to disable see Figure 3-23.

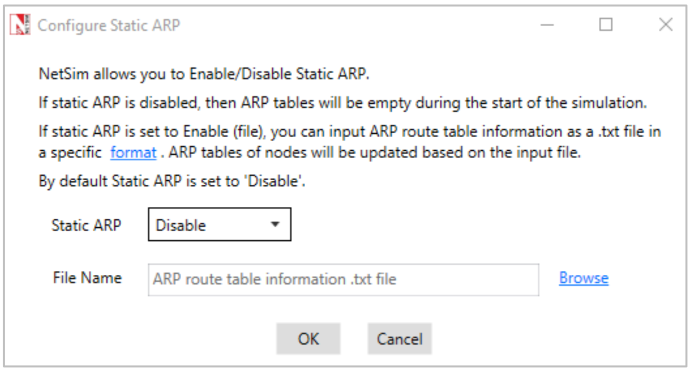


Figure 3-23: Static ARP Configuration Window

Click on OK.

If Static ARP is enabled, then NetSim will automatically create an ARP table for each node. To see the working of the ARP protocol users should disable Static ARP.

By doing so, ARP request would be sent to the destination to find out the destinations MAC Address.

3.2.7 Output – ARP across a LAN

Once the simulation is complete, to view the packet trace file, click on **"Open Packet Trace"** option present in the left-hand-side of the Results Dashboard.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	NODE-1	SWITCH-4
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	SWITCH-4	ROUTER-6
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	SWITCH-4	NODE-2
0	N/A	Control_Packet	ARP_Reply	NODE-2	NODE-1	NODE-2	SWITCH-4
0	N/A	Control_Packet	ARP_Reply	NODE-2	NODE-1	SWITCH-4	NODE-1

Figure 3-24: Open Packet Trace

NODE 1 will send ARP REQUEST to SWITCH-4, SWITCH-4 sends this to ROUTER-6, and SWITCH-4 also sends this to NODE-2. ARP-REPLY is sent by the NODE-2 to SWITCH -4, and in-turn SWITCH-4 sends it to NODE-1.

3.2.8 Discussion – ARP across a LAN

Intra-LAN-IP-forwarding:

ARP PROTOCOL- WORKING:

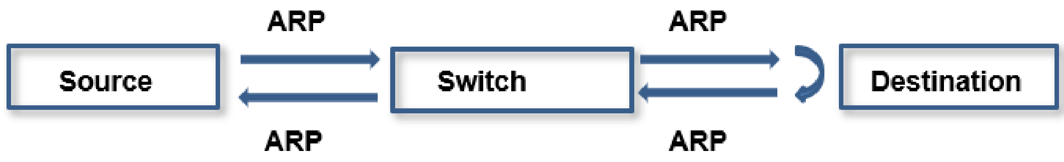


Figure 3-25: Intra LAN IP Forwarding

NODE-1 broadcasts ARP Request, which is then broadcasted by SWITCH-4. NODE-2 sends the ARP Reply to NODE-1 via SWITCH-4. After this step, datagrams are transmitted from NODE-1 to NODE-2. Notice the DESTINATION ID column for ARP Request type packets, which indicates Broadcast-0.

ARP across a WAN

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 3-26.

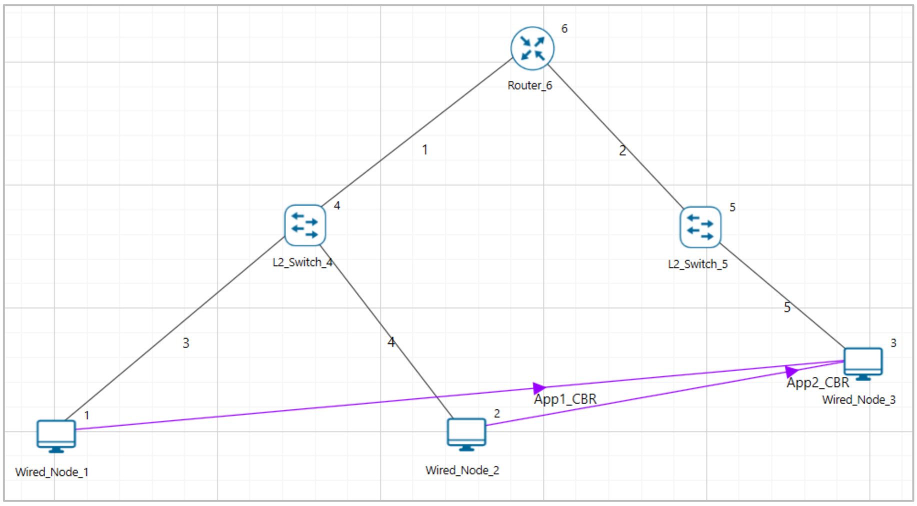


Figure 3-26: Network set up for studying the ARP across a WAN

3.2.9 Procedure

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

Step 1: A network scenario is designed in the NetSim GUI comprising of 3 Wired Nodes, 2 L2 Switches, and 1 Router.

Step 2: Click on the Set Traffic tab and configure the application between the nodes. Click on the application, expand the property panel on the right and set the properties as mentioned below.

APP 1 CBR is created from Wired Node 1 to Wired Node 3, Packet size set as 1460 bytes and Inter arrival time as 20000 Micro sec and Transport layer protocol to UDP.

APP 2 CBR is created from Wired Node 2 to Wired Node 3, Packet size set as 1460 bytes and Inter arrival time as 20000 Micro sec and Transport layer protocol to UDP. Additionally, the start time is set to 1 second and end time to 3 second.

Transport Protocol is set to UDP instead of TCP. If set to TCP, the ARP table will get updated during transmission of TCP control packets thereby eliminating the need for ARP to resolve addresses.

Step 3: Packet Trace is enabled from Configured reports tab, and hence we can view the ARP Request and ARP Reply packets exchanged initially, before transmission of the data packets.

Step 4: Click on Run simulation. The simulation time is set to 10 seconds.

Step 5: Under Options, the "Static ARP" tab, Static ARP is set to disable.

3.2.10 Output I – ARP across a WAN

Once the simulation is complete, to view the packet trace file, click on **"Open Packet Trace"** option present in the left-hand-side of the Results Dashboard.

In packet trace, filter the CONTROL PACKET TYPE/APP NAME filed to view APP 1CBR, ARP_REQUEST, ARP_REPLY.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	NODE-1	SWITCH-4
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	SWITCH-4	ROUTER-6
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	SWITCH-4	NODE-2
0	N/A	Control_Packet	ARP_Reply	ROUTER-6	NODE-1	ROUTER-6	SWITCH-4
0	N/A	Control_Packet	ARP_Reply	ROUTER-6	NODE-1	SWITCH-4	NODE-1
1	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	SWITCH-4
1	0	CBR	App1_CBR	NODE-1	NODE-3	SWITCH-4	ROUTER-6
0	N/A	Control_Packet	ARP_Request	ROUTER-6	Broadcast-0	ROUTER-6	SWITCH-5
0	N/A	Control_Packet	ARP_Request	ROUTER-6	Broadcast-0	SWITCH-5	NODE-3
0	N/A	Control_Packet	ARP_Reply	NODE-3	ROUTER-6	NODE-3	SWITCH-5
0	N/A	Control_Packet	ARP_Reply	NODE-3	ROUTER-6	SWITCH-5	ROUTER-6
1	0	CBR	App1_CBR	NODE-1	NODE-3	ROUTER-6	SWITCH-5
1	0	CBR	App1_CBR	NODE-1	NODE-3	SWITCH-5	NODE-3
2	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	SWITCH-4
2	0	CBR	App1_CBR	NODE-1	NODE-3	SWITCH-4	ROUTER-6
2	0	CBR	App1_CBR	NODE-1	NODE-3	ROUTER-6	SWITCH-5
2	0	CBR	App1_CBR	NODE-1	NODE-3	SWITCH-5	NODE-3
3	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	SWITCH-4
3	0	CBR	App1_CBR	NODE-1	NODE-3	SWITCH-4	ROUTER-6
3	0	CBR	App1_CBR	NODE-1	NODE-3	ROUTER-6	SWITCH-5
3	0	CBR	App1_CBR	NODE-1	NODE-3	SWITCH-5	NODE-3
4	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	SWITCH-4
4	0	CBR	App1_CBR	NODE-1	NODE-3	SWITCH-4	ROUTER-6

Figure 3-27: Open Packet Trace

NODE 1 will send ARP REQUEST to SWITCH-4, SWITCH-4 sends this to ROUTER-6, and SWITCH-4 also sends this to NODE-2. ARP-REPLY is sent by the ROUTER-6 to SWITCH -4, and in-turn SWITCH-4 sends it to NODE-1. Again ROUTER-6 will send ARP REQUEST to SWITCH-5, SWITCH-5 sends this to NODE-3. ARP REPLY is sent by NODE-3 to SWITCH-5 and in-turn SWITCH-5 sends it to ROUTER-6.

The IP forwarding table formed in the router can be accessed from the IP Forwarding Table list present in the Simulation Results window as shown below Figure 3-28.

ROUTER_6

Network Destination	Netmask/Prefix len	Gateway	Interface	Metrics	Type
192.169.0.0	255.255.0.0	on-link	192.169.0.1	300	LOCAL
192.168.0.0	255.255.0.0	on-link	192.168.0.1	300	LOCAL
224.0.0.1	255.255.255.255	on-link	192.168.0.1 192.169.0.1	306	MULTICAST
224.0.0.0	240.0.0.0	on-link	192.168.0.1 192.169.0.1	306	MULTICAST
255.255.255.255	255.255.255.255	on-link	192.168.0.1 192.169.0.1	999	BROADCAST

Figure 3-28: IP Forwarding Table

Click on Detailed View checkbox to view the additional fields as indicated above.

Router forwards packets intended to the subnet 192.169.0.0 to the interface with the IP 192.168.0.1 based on the first entry in its routing table..

3.2.11 Discussion I – ARP across a WAN

From the above case we can understand that, since Router 6 did not know the destination address, the Application packets reach only till Router 6, and ARP mechanism continues with Router 6 re-broadcasting the ARP REQUEST, finding the destination address and the datagram is getting transferred to Wired node 3 (destination).

3.2.12 Output II – ARP across a WAN

In same packet trace, filter the CONTROL PACKET TYPE/APP NAME column to view APP 2 CBR, ARP REQUEST, ARP REPLY only.

In the below figure user can observe that ARP REQUEST is broadcasted from Wired Node 2, the ARP Reply is sent from the Router 6, upon receiving the ARP REPLY. Router 6 directly starts sending the data packet to the Wired Node 3 unlike the previous sample.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	NODE-1	SWITCH-4
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	SWITCH-4	ROUTER-6
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	SWITCH-4	NODE-2
0	N/A	Control_Packet	ARP_Reply	ROUTER-6	NODE-1	ROUTER-6	SWITCH-4
0	N/A	Control_Packet	ARP_Reply	ROUTER-6	NODE-1	SWITCH-4	NODE-1
0	N/A	Control_Packet	ARP_Request	ROUTER-6	Broadcast-0	ROUTER-6	SWITCH-5
0	N/A	Control_Packet	ARP_Request	ROUTER-6	Broadcast-0	SWITCH-5	NODE-3
0	N/A	Control_Packet	ARP_Reply	NODE-3	ROUTER-6	NODE-3	SWITCH-5
0	N/A	Control_Packet	ARP_Reply	NODE-3	ROUTER-6	SWITCH-5	ROUTER-6
0	N/A	Control_Packet	ARP_Request	NODE-2	Broadcast-0	NODE-2	SWITCH-4
0	N/A	Control_Packet	ARP_Request	NODE-2	Broadcast-0	SWITCH-4	ROUTER-6
0	N/A	Control_Packet	ARP_Request	NODE-2	Broadcast-0	SWITCH-4	NODE-1
0	N/A	Control_Packet	ARP_Reply	ROUTER-6	NODE-2	ROUTER-6	SWITCH-4
0	N/A	Control_Packet	ARP_Reply	ROUTER-6	NODE-2	SWITCH-4	NODE-2
1	0	CBR	App2_CBR	NODE-2	NODE-3	NODE-2	SWITCH-4
1	0	CBR	App2_CBR	NODE-2	NODE-3	SWITCH-4	ROUTER-6
1	0	CBR	App2_CBR	NODE-2	NODE-3	ROUTER-6	SWITCH-5
1	0	CBR	App2_CBR	NODE-2	NODE-3	SWITCH-5	NODE-3
2	0	CBR	App2_CBR	NODE-2	NODE-3	NODE-2	SWITCH-4
2	0	CBR	App2_CBR	NODE-2	NODE-3	SWITCH-4	ROUTER-6
2	0	CBR	App2_CBR	NODE-2	NODE-3	ROUTER-6	SWITCH-5
2	0	CBR	App2_CBR	NODE-2	NODE-3	SWITCH-5	NODE-3
3	0	CBR	App2_CBR	NODE-2	NODE-3	NODE-2	SWITCH-4
3	0	CBR	App2_CBR	NODE-2	NODE-3	SWITCH-4	ROUTER-6
3	0	CBR	App2_CBR	NODE-2	NODE-3	ROUTER-6	SWITCH-5

Figure 3-29: Open Packet Trace

3.2.13 Discussion II – ARP across a WAN

Across-Router-IP-forwarding

ARP PROTOCOL- WORKING

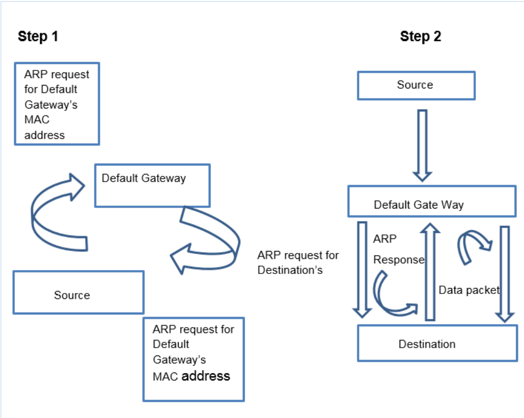


Figure 3-30: Across Router IP Forwarding

NODE-2 transmits ARP Request which is further broadcasted by SWITCH-4. ROUTER-6 sends ARP Reply to NODE-2 which goes through SWITCH-4. Then NODE-2 starts sending datagrams to NODE-3. If router has the MAC address of NODE-3 in its ARP table, then ARP ends here, and router starts forwarding the datagrams to NODE-3 by consulting its forwarding table. Router 6, has this information updated during transmission of APP1 packets and hence ARP request for identifying the MAC address of NODE-3, need not be sent again. In the other case (Output -I), Router sends ARP Request to appropriate subnet and after getting the MAC address of NODE-3, it then forwards the datagrams to NODE-3 using its forwarding table.

3.2.14 Exercises

1. Construct a below ARP experiment by manually adding the ARP entries using the Static ARP option with the file option. Refer to user manual section 3.15, "Static ARP configuration in NetSim".

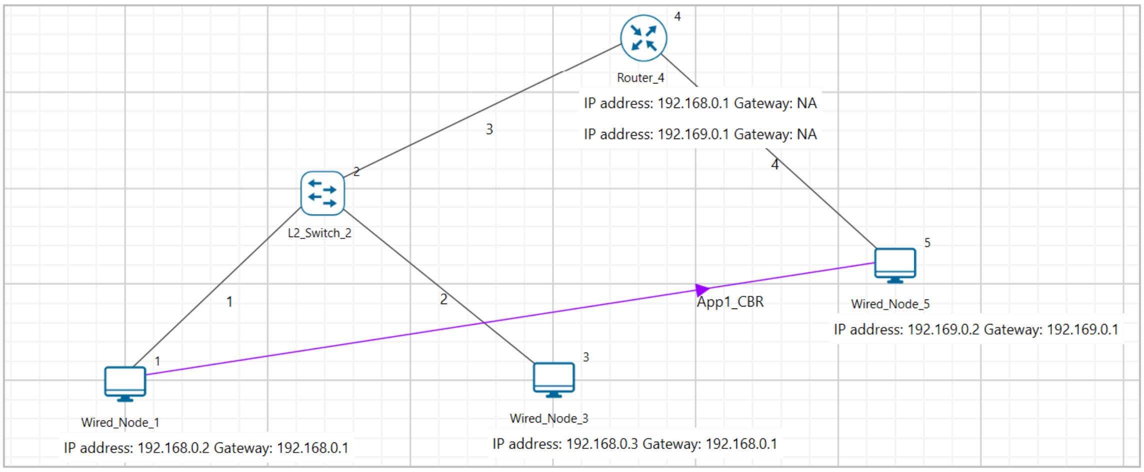


Figure 3-31: Network scenario for ARP exercise.

3.3 Simulate and study the spanning tree protocol (Level 1)

3.3.1 Introduction

Spanning Tree Protocol (STP) is a link management protocol. Using the spanning tree algorithm, STP provides path redundancy while preventing undesirable loops in a network that are created by multiple active paths between stations. Loops occur when there are alternate routes between hosts. To establish path redundancy, STP creates a tree that spans all of the switches in an extended network, forcing redundant paths into a standby, or blocked state. STP allows only one active path at a time between any two network devices (this prevents the loops) but establishes the redundant links as a backup if the initial link should fail. Without spanning tree in place, it is possible that both connections may simultaneously live, which could result in an endless loop of traffic on the LAN.

*(**Reference:** A good reference for this topic is Section 3.1.4: Bridges and LAN switches, of the book, Computer Networks, 5th Edition by Peterson and Davie)*

3.3.2 Network Setup

Open NetSim and click on **Experiments> Internetworks> Routing and Switching> Simulate and study the spanning tree protocol** then click on the tile in the middle panel to load the example as shown in below Figure 3-32.

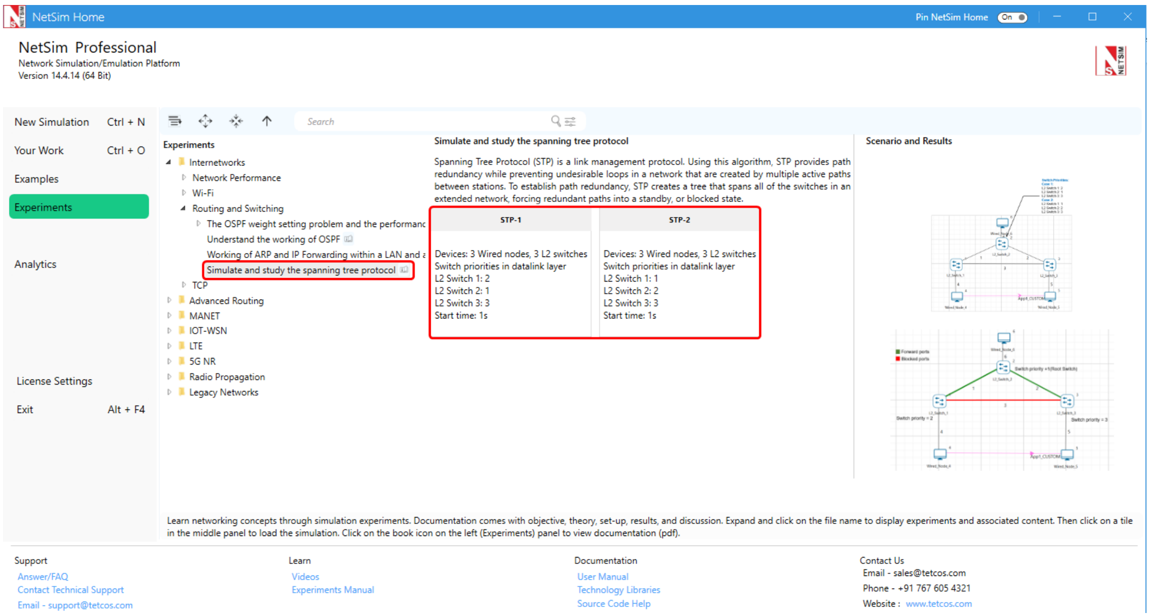


Figure 3-32: List of scenarios for the example of Simulate and study the spanning tree protocol

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 3-33.

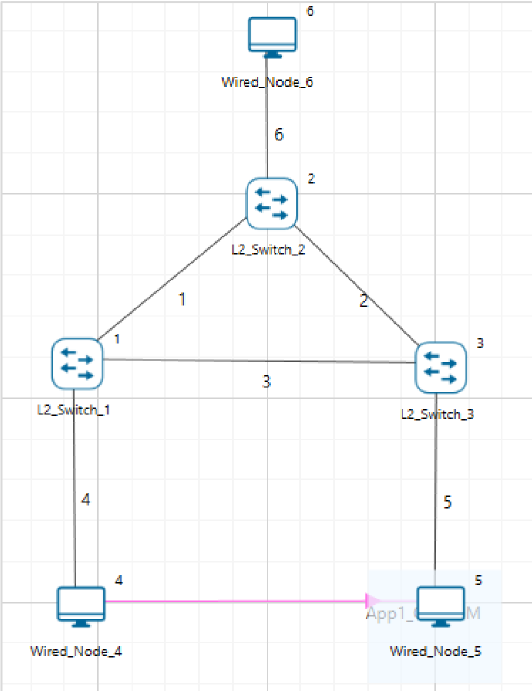


Figure 3-33: Network set up for studying the STP 1

NOTE: At least three L2 Switches are required in the network to analyze the spanning tree formation.

3.3.3 Procedure

STP-1

Step 1: A network scenario is designed in the NetSim GUI comprising of 3 Wired Nodes and 3 L2 Switches in the **"Internetworks"** Network Library.

Step 2: Go to L2 Switch 1 Properties. In the Interface 1 (ETHERNET) > Datalink Layer, **"Switch Priority"** is set to 2. Similarly, for the other interfaces of L2 Switch 1, Switch Priority is set to 2.

To configure any properties in the device, click on the device, expand the property panel on the right side, and change the properties as mentioned in the steps.

Step 3: Go to L2 Switch 2 Properties. In the Interface 1 (ETHERNET) > Datalink Layer, **"Switch Priority"** is set to 1. Similarly, for the other interfaces of L2 Switch 2, Switch Priority is set to 1.

Step 4: Go to L2 Switch 3 Properties. In the Interface 1 (ETHERNET) > Datalink Layer, **"Switch Priority"** is set to 3. Similarly, for the other interfaces of L2 Switch 3, Switch Priority is set to 3.

L2_Switch Properties	L2_Switch 1	L2_Switch 2	L2_Switch 3
Switch Priority	**2**	**1**	**3**

Table 3-5: Switch Priorities for STP-1

NOTE: Switch Priority is set to all the 3 L2 Switches and Switch Priority has to be changed for all the interfaces of L2 Switch.

Switch Priority is interpreted as the weights associated with each interface of a L2 Switch. A higher value indicates a higher priority.

Step 5: Configure Custom application between two wired nodes by clicking on set traffic tab from ribbon on the top. Click on the created application and expand the application property panel on right, set the start time to 1 second and by keeping other properties as default.

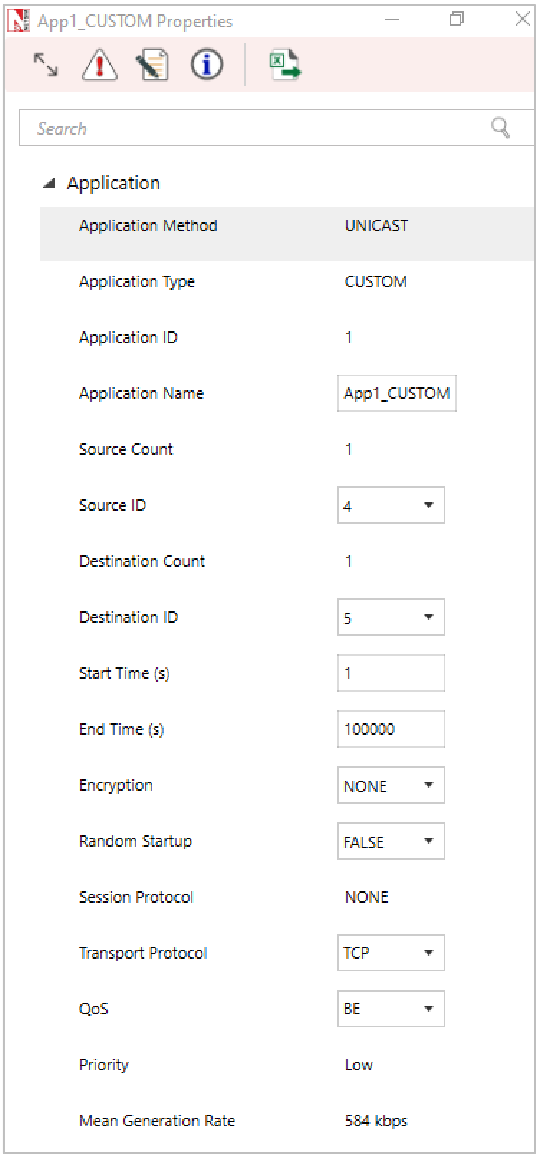


Figure 3-34: Application Configuring Window

Step 6: Enable the packet trace from configure reports tab and run simulation for 10 seconds.

STP-2

The following changes in settings are done from the previous Sample:

In STP 2, the **"Switch Priority"** of all the 3 L2 Switches are changed as follows Table 3-6:

L2 Switch Properties	L2 Switch 1	L2 Switch 2	L2 Switch 3
Switch Priority	1	2	3

Table 3-6: Switch Priorities for STP 2

3.3.4 Output

The active and blocked ports for the two samples STP-1 and STP-2 are illustrated in the below screenshots based on the data flow observed in the packet trace.

STP-1

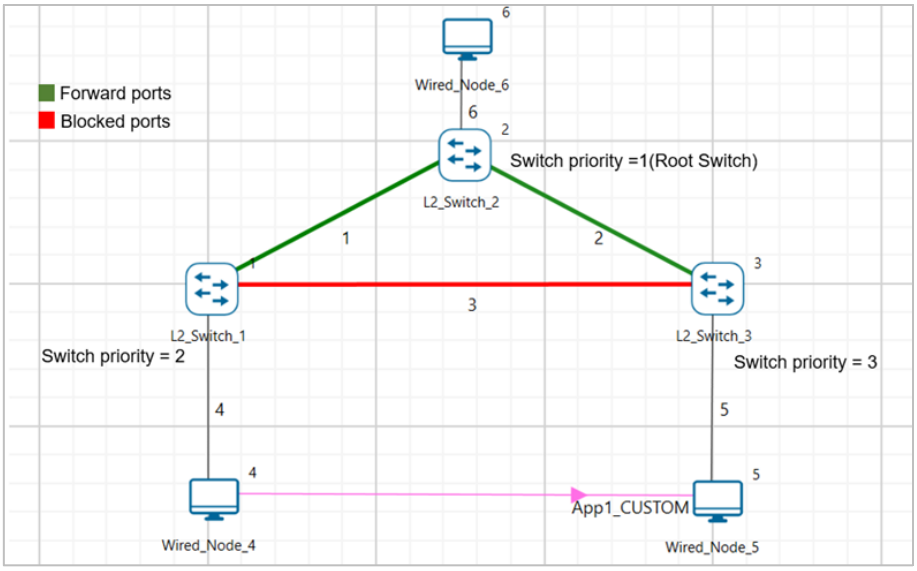


Figure 3-35: A representative image showing active and blocked ports for sample STP-1 based on the path of data flow we observe in the packet trace

Go to Packet Trace and observe that, after the exchange of control packets, the data packets take the following path **Wired Node 4 > L2 Switch 1 > L2 Switch 3 > Wired Node 5.**

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	Custom	App1_CUSTOM	NODE-4	NODE-5	NODE-4	SWITCH-1
1	0	Custom	App1_CUSTOM	NODE-4	NODE-5	SWITCH-1	SWITCH-3
1	0	Custom	App1_CUSTOM	NODE-4	NODE-5	SWITCH-3	NODE-5

Figure 3-36: Observe the transmitter ID and the receiver ID columns in the Packet Trace to determine the path of data flow.

STP-2

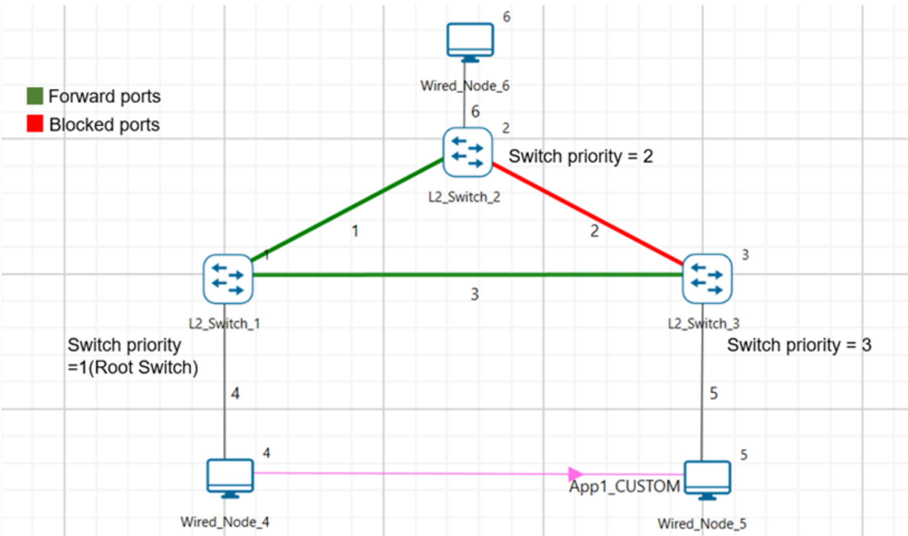


Figure 3-37: A representative image showing active and blocked ports for sample STP-2 based on the path of data flow we observe in the packet trace.

Go to Packet Trace and observe that, after the exchange of control, the data packets take the following path. **Wired Node 4 > L2 Switch 1 > L2 Switch 3 > Wired Node 5.**

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	Custom	App1_CUSTOM	NODE-4	NODE-5	NODE-4	SWITCH-1
1	0	Custom	App1_CUSTOM	NODE-4	NODE-5	SWITCH-1	SWITCH-3
1	0	Custom	App1_CUSTOM	NODE-4	NODE-5	SWITCH-3	NODE-5

Figure 3-38: Observe the transmitter ID and the receiver ID columns in the Packet Trace to determine the path of data flow.

Go to Simulation Results window, In the left panel of the Results Dashboard, click on the Additional metrics and slide down to obtain the Switch MAC address table list of all the L2 Switches.

For each L2 Switch, a Switch MAC Address Table containing the MAC address entries see **Figure 3-39**, the port that is used for reaching it, along with the type of entry can be obtained at the end of Simulation.

Switch Mac address table

L2_SWITCH_1

Mac Address	Type	OutPort
155D00002001	Dynamic	1
155D00003002	Dynamic	2
155D00004001	Dynamic	3
155D00005001	Dynamic	2

L2_SWITCH_2

Mac Address	Type	OutPort
155D00001001	Dynamic	1
155D00003001	Dynamic	2
155D00006001	Dynamic	3
155D00004001	Dynamic	1

L2_SWITCH_3

Mac Address	Type	OutPort
155D00002002	Dynamic	1
155D00001002	Dynamic	2
155D00005001	Dynamic	3
155D00004001	Dynamic	2

Figure 3-39: STP 2 MAC Address table

3.3.5 Discussion

Each L2 Switch has an ID which is a combination of its Lowest MAC address and priority. The Spanning tree algorithm selects the L2 Switch with the smallest ID as the root node of the

Spanning Tree. The root node forward frames out over all its ports. In the other L2 Switches, the ports that have the least cost of reaching the root switch are set as **Forward Ports** and the remaining are set as **Blocked Ports**. In the STP-1, L2 Switch 2 was assigned least priority and was selected as a Root Switch. The green line indicates the forward path, and the red line indicates the blocked path. The frame from Wired Node 4 should take the path through the L2 Switch 1, 2 and 3 to reach the Wired Node 5. In the STP-2, L2 Switch 1 was assigned least priority and selected as a Root switch. In this case, the frame from Wired Node 4 takes the path through the L2 Switch 1 and 3 to reach the destination Wired Node 5.

3.4 Understanding VLAN operation in L2 and L3 Switches (Level 2)

3.4.1 Introduction to VLAN

VLAN is called as virtual local area network, used in Switches and it operates at Layer 2 and Layer 3. A VLAN is a group of hosts which communicate as if they were attached to the same broadcast domain, regardless of their physical Location.

For example, all workstations and servers used by a particular workgroup team can be connected to the same VLAN, regardless of their physical connections to the network or the fact that they might be intermingled with other teams. VLANs have the same attributes as physical LANs, but you can group end stations even if they are not physically located on the same LAN Segment.

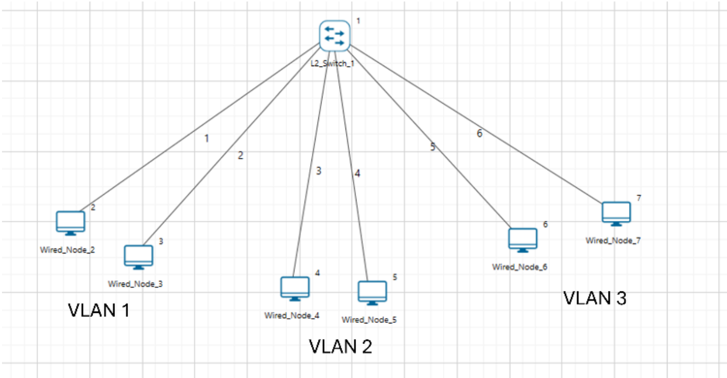


Figure 3-40: Virtual local area network (VLAN)

A VLAN behaves just like a LAN in all respects but with additional flexibility. By using VLAN technology, it is possible to subdivide a single physical switch into several logical switches. VLANs are implemented by using the appropriate switch configuration commands to create the VLANs and assign specific switch interfaces to the desired VLAN.

Switches implement VLANs by adding a VLAN tag to the Ethernet frames as they enter the switch. The VLAN tag contains the VLAN ID and other information, which is determined by the interface from which the frame enters the switch. The switch uses VLAN tags to ensure that each Ethernet frame is confined to the VLAN to which it belongs based on the VLAN ID contained in the VLAN tag. The VLAN tags are removed as the frames exit the switch on the way to their destination.

Any port can belong to a VLAN, and unicast, broadcast, and multicast packets are forwarded and flooded only to end stations in that VLAN. Each VLAN is considered a logical network. Packets destined for stations that do not belong to the VLAN must be forwarded through a router.

In the below screenshot, the stations in the development department are assigned to one VLAN, the stations in the marketing department are assigned to another VLAN, and the stations in the testing department are assigned to another VLAN.

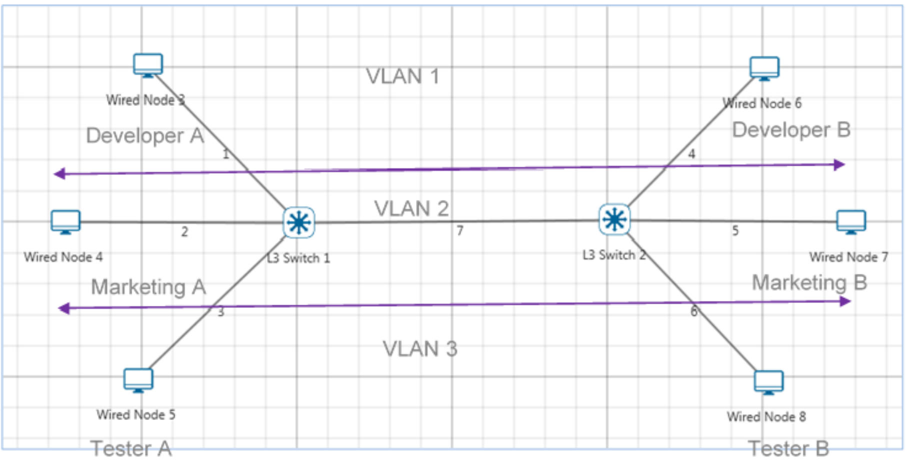


Figure 3-41: Hosts in one VLAN need to communicate with hosts in another VLAN This is known as Inter-VLAN routing.

VLANs divide broadcast domains in a LAN environment. Whenever hosts in one VLAN need to communicate with hosts in another VLAN, the traffic must be routed between them. This is known as Inter-VLAN routing. This can be possible by using L3 Switch.

3.4.1.1 What is a layer 3 switch?

Layer 3 switch (also known as a multi-layer switch) is a multi-functional device that have the same functionality like a layer 2 switch, but behaves like a router when necessary. It’s generally faster than a router due to its hardware-based routing functions, but it’s also more expensive than a normal switch.

3.4.2 Network setup

Open NetSim and click on **Experiments > Advanced Routing> Understanding VLAN operation in L2 and L3 Switches** then click on the tile in the middle panel to load the example as shown in below in Figure 3-42.

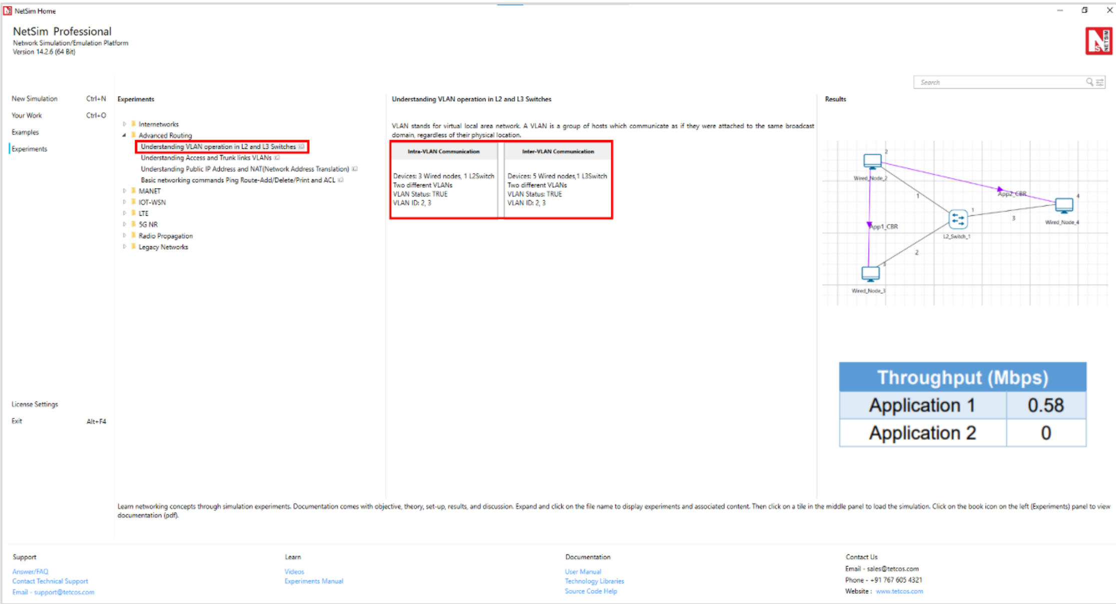


Figure 3-42: List of scenarios for the example of Understanding VLAN operation in L2 and L3 Switches

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 3-43.

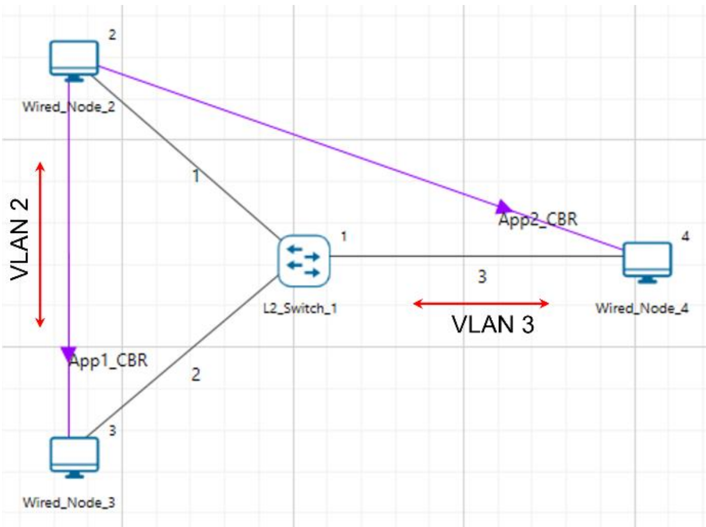


Figure 3-43: Network set up for studying the Intra-VLAN

3.4.3 Procedure

Intra-VLAN

Intra-VLAN is a mechanism in which hosts in same VLAN can communicate to each other.

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

Step 1: A network scenario is designed in NetSim GUI comprising of 3 Wired Nodes and 1 L2 Switch in the **“Internetworks”** Network Library.

Step 2: Click on L2 Switch 1 and expand property panel on the right and set the properties as shown in Table 3-7.

L2 Switch 1			
Interface ID	**VLAN Status**	**VLAN ID**	**VLAN Port Type**
Interface 1	TRUE	2	Access Port
Interface 2	TRUE	2	Access Port
Interface 3	TRUE	3	Access Port

Table 3-7: L2 Switch 1 Properties

In all the INTERFACE (ETHERNET) > DATALINK LAYER Properties of L2 Switch 1, "**VLAN Status**" is set to TRUE.

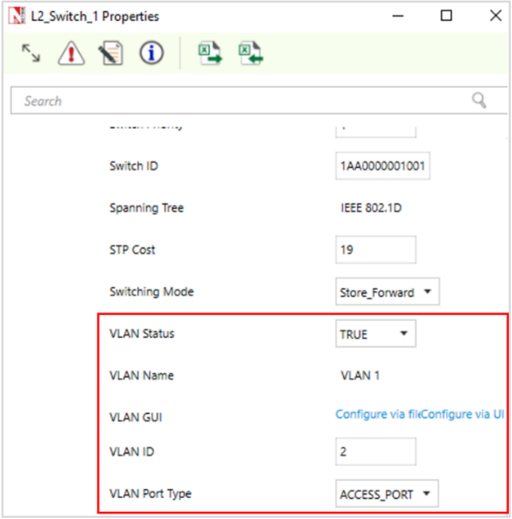


Figure 3-44: DATALINK LAYER Properties of L2 Switch 1

Now click on **"Configure VLAN"** option and the VLAN 2 fields are entered as shown below Figure 3-39.

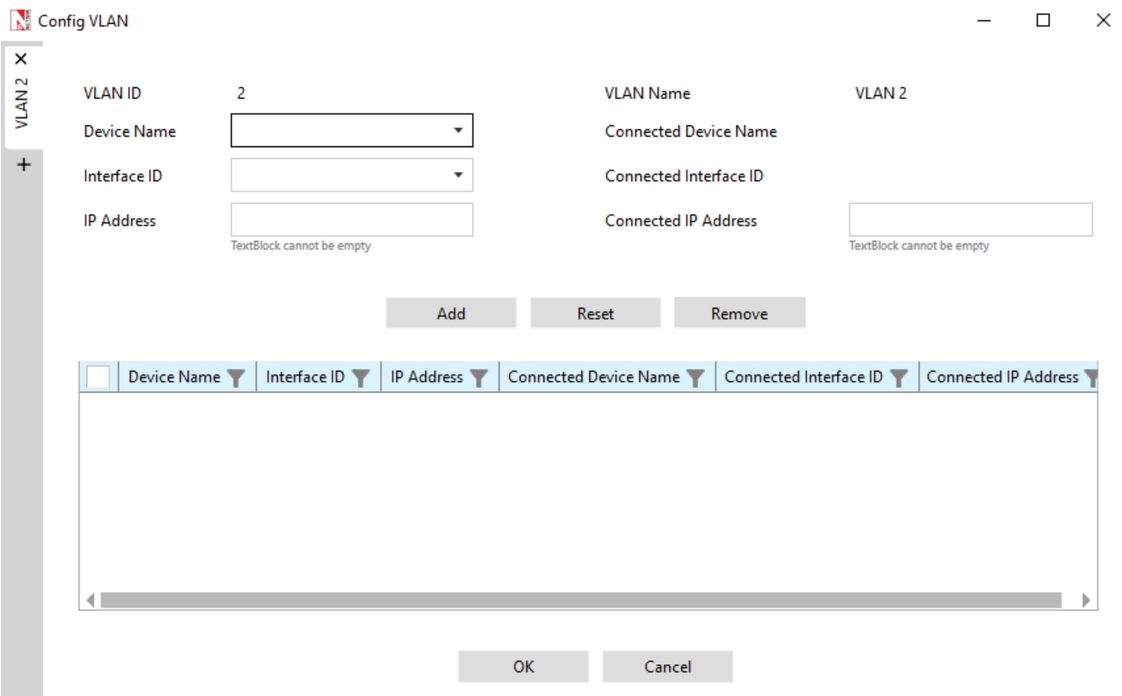


Figure 3-45: VLAN Configure window

To add a new entry after entering the required fields, click on the ADD button.

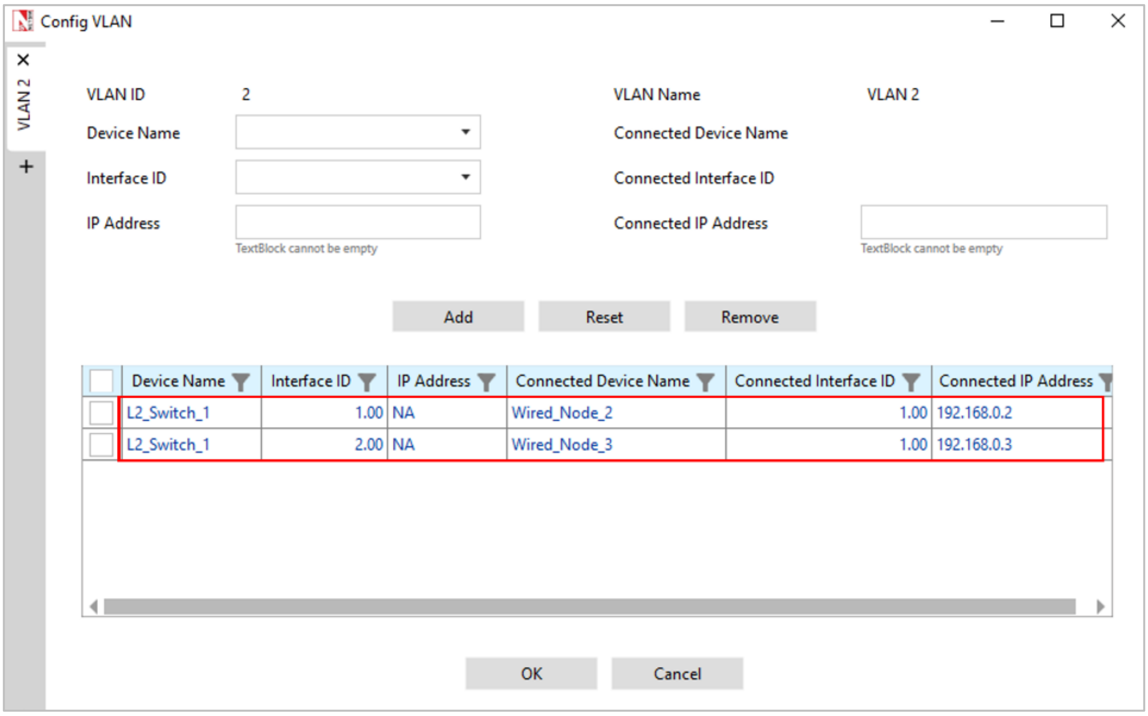


Figure 3-46: Configuring VLAN Properties in VLAN 2

To configure another VLAN, click on the **"+"** symbol located in the top.

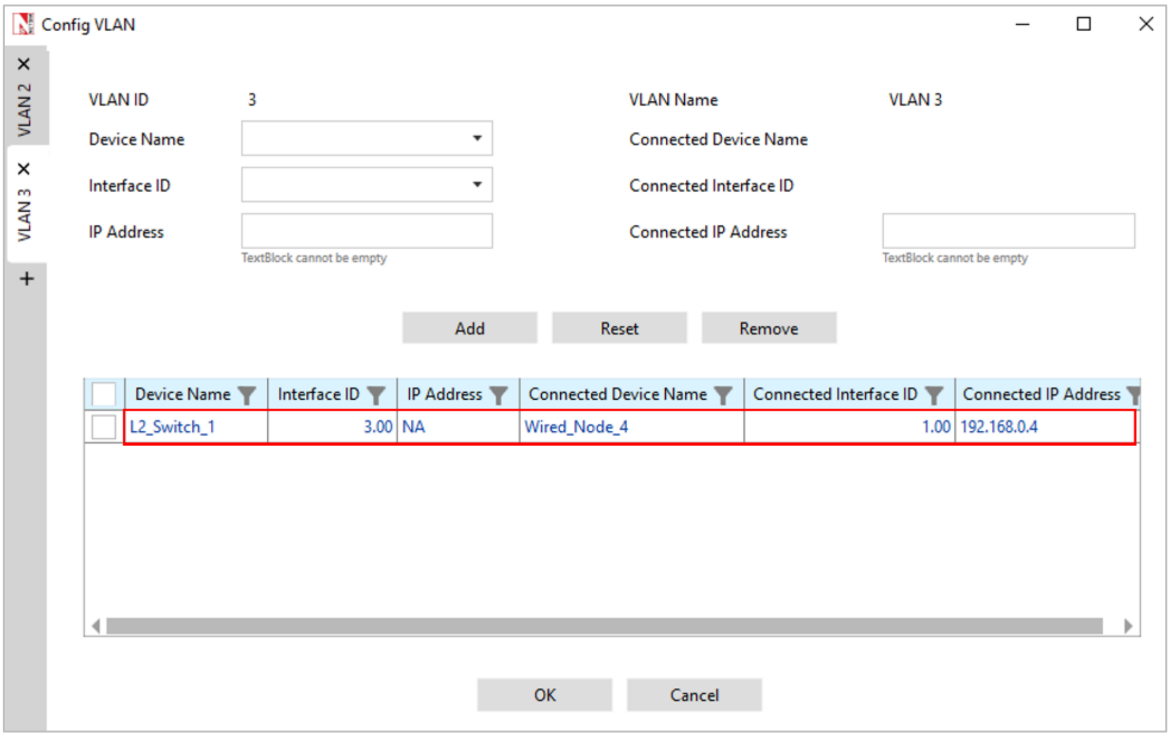


Figure 3-47: Configuring VLAN Properties in VLAN 3

And then we can add the entry to it.

Step 3: Click on the "Configure Reports" tab in the top ribbon, enable the plots, run the simulation for 10 seconds and observe the throughputs.

Inter-VLAN

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 3-48.

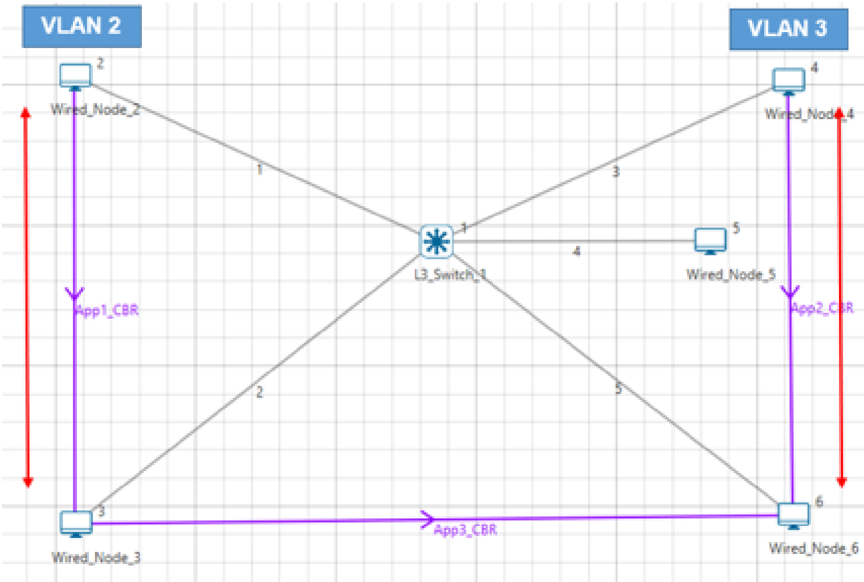


Figure 3-48: Network set up for studying the Inter-VLAN

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

Step 1: A network scenario is designed in NetSim GUI comprising of 5 Wired Nodes and 1 L3 Switch in the **"Internetworks"** Network Library.

Step 2: Click on Wired Node and expand property panel on the right and set the properties are as per the below Table 3-8.

Node	Wired Node2	Wired Node3	Wired Node4	Wired Node5	Wired Node6
	I/f1_Ethernet	I/f1_Ethernet	I/f1_Ethernet	I/f1_Ethernet	I/f1_Ethernet
IP Address	10.0.0.4	10.1.0.4	11.2.0.4	11.3.0.4	11.4.0.4
Default Gateway	10.0.0.3	10.1.0.3	11.2.0.3	11.3.0.3	11.4.0.3

Table 3-8: Wired Node properties

Step 3: The L3 Switch 1 Properties are set as per the below table:

L3 Switch	I/f1_Ethernet	I/f2_Ethernet	I/f3_Ethernet	I/f4_Ethernet	I/f5_Ethernet
	IP Address	IP Address	IP Address	IP Address	IP Address
L3 Switch 1	10.0.0.3	10.1.0.3	11.2.0.3	11.3.0.3	11.4.0.3

Table 3-9: L3 Switch 1 Properties

L3 Switch 1			
Interface ID	VLAN Status	VLAN ID	VLAN Port Type
Interface 1	TRUE	2	Access Port
Interface 2	TRUE	2	Access Port
Interface 3	TRUE	3	Access Port
Interface 4	TRUE	3	Access Port
Interface 5	TRUE	3	Access Port

Table 3-10: VLAN configurations Properties

The VLAN configurations done are shown as follows:

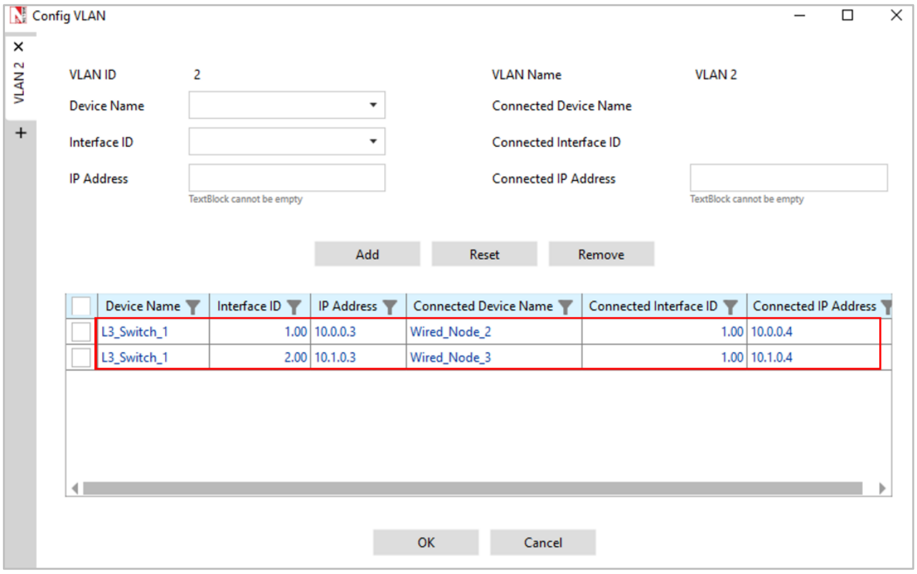


Figure 3-49: Configuring VLAN Properties in VLAN 2

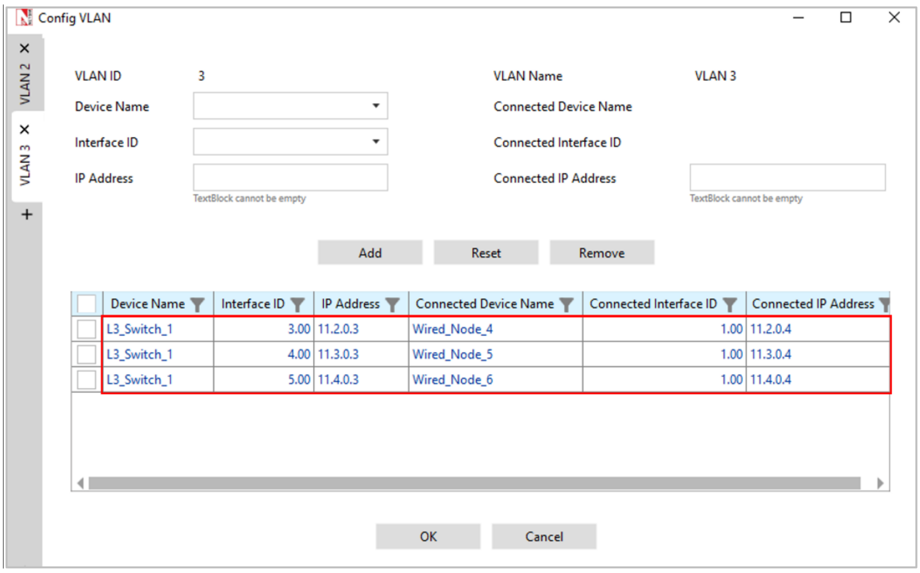


Figure 3-50: Configuring VLAN Properties in VLAN 3

Step 4: Click on the "Configure Reports" tab in the top ribbon, enable the plots, run the simulation for 10 seconds and observe the throughputs.

3.4.4 Output and Inference for Intra-VLAN

Throughput (Mbps)	
Application 1	0.58
Application 2	0

Table 3-11: Results Comparison

The throughput for 2nd application is zero because the source and destination is in different VLANs, thereby traffic flow or communication between 2 VLANs using Layer 2 switch is not possible. To overcome this problem, an L3 switch is used.

3.4.5 Output and Inference for Inter-VLAN

Throughput (Mbps)	
Application 1	0.58
Application 2	0.58
Application 3	0.58

Table 3-12: Results Comparison

In this case, application1 is in VLAN2, application2 is in VLAN3 and application 3 is in between VLAN2 and VLAN3. From the above results, the throughput for application 3 (different VLANs) is nonzero, because of using L3 switch. So, communication between 2 VLANs is possible using L3 Switch.

3.4.6 Exercises

1. Construct the network below by configuring VLANs in L2Switch 4 as follows.
   a. Configure Node1 and Node2 as part of one VLAN and Node2 and Node3 in a different VLAN. Analyse traffic flow within and between different VLANs.
   b. Configure Node1 and Node3 as part of one VLAN and Node2 and Node3 in a different VLAN. Analyse traffic flow within and between different VLANs.

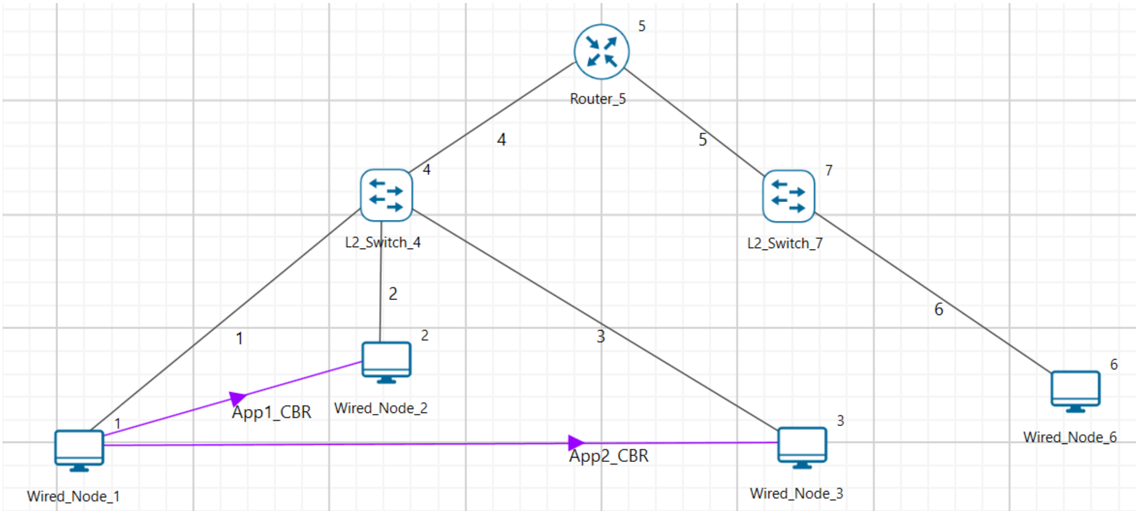


Figure 3-51: Network scenario for VLAN.

3.5 Understanding Access and Trunk Links in VLANs (Level 2)

3.5.1 Theory

An access link is a link that is part of only one VLAN, and normally access links are for end devices. An access-link connection can understand only standard Ethernet frames. Switches remove any VLAN information from the frame before it is sent to an access-link device.

A Trunk link can carry multiple VLAN traffic and normally a trunk link is used to connect switches to other switches or to routers. A trunk link is not assigned to a specific VLAN. Multiple VLAN traffic can be transported between switches using a single physical trunk link.

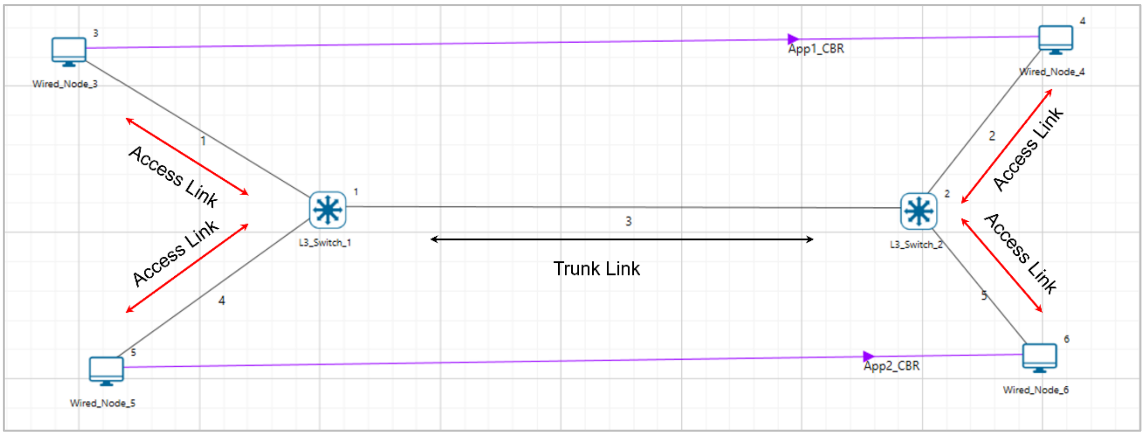


Figure 3-52: Understanding Access and Trunk Links in VLANs

Access link

Access link connection is the connection where switch port is connected with a device that has a standardized Ethernet NIC. Standard NIC only understand IEEE 802.3 or Ethernet II frames. Access link connection can only be assigned with single VLAN. That means all devices connected to this port will be in same broadcast domain.

For example, twenty users are connected to a hub, and we connect that hub with an access link port on switch, then all of these users belong to same VLAN. If we want to keep ten users in another VLAN, then we need to plug in those ten users to another hub and then connect it with another access link port on switch.

Trunk link

Trunk link connection is the connection where switch port relates to a device that is capable to understand multiple VLANs. Usually, trunk link connection is used to connect two switches. A VLAN can span anywhere in network, and that can happen due to trunk link connection. Trunking allows us to send or receive VLAN information across the network. To support trunking, original Ethernet frame is modified to carry VLAN information.

3.5.2 Network Setup

Open NetSim and click on **Experiments> Advanced Routing> Understanding Access and Trunk Links in** VLANs then click on the tile in the middle panel to load the example as shown in below in Figure 3-53.

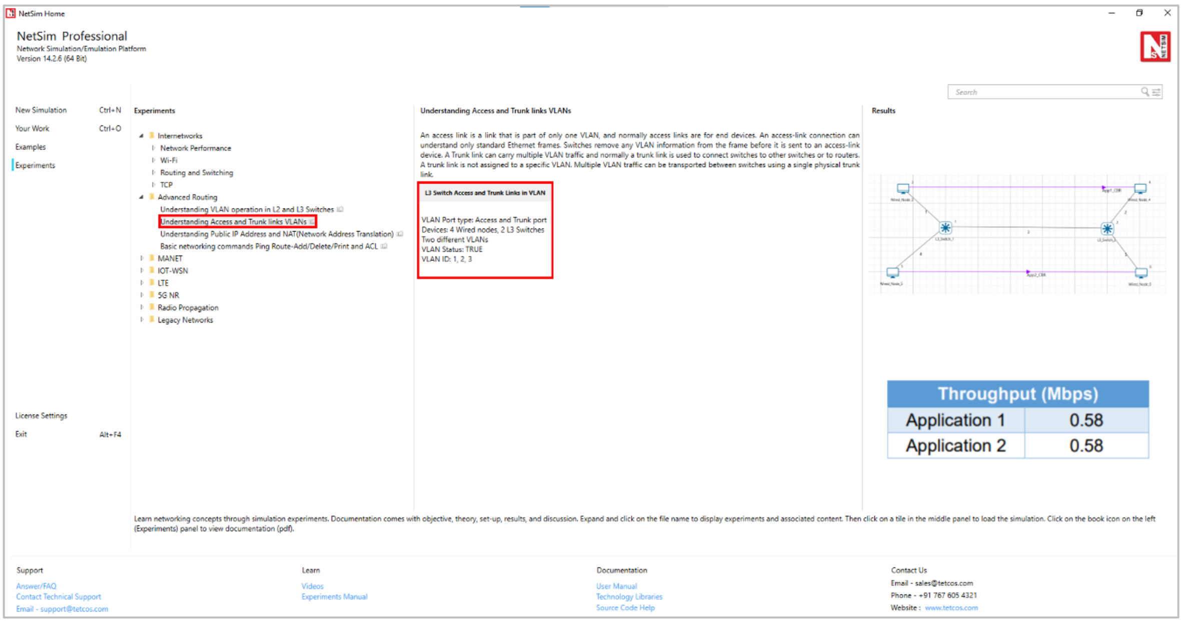


Figure 3-53: List of scenarios for the example of Understanding Access and Trunk Links in VLANs

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 3-54.

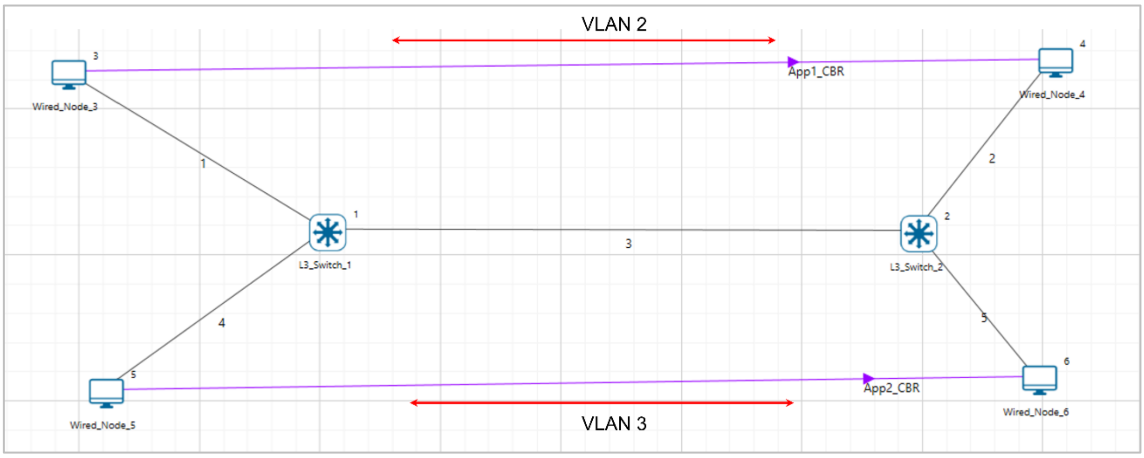


Figure 3-54: Network set up for studying the L3 Switch Access and Trunk Links in VLANs

3.5.3 Procedure

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

Step 1: A network scenario is designed in NetSim GUI comprising of 4 Wired Nodes and 2 L3 Switches in the **"Internetworks"** Network Library.

Step 2: In the INTERFACE (ETHERNET) > NETWORK LAYER Properties, set the following **Table** 3-13.

Node	Wired Node 3	Wired Node 4	Wired Node 5	Wired Node 6
	I/f1_Ethernet	I/f1_Ethernet	I/f1_Ethernet	I/f1_Ethernet
IP Address	192.168.1.3	192.168.1.4	192.168.2.3	192.168.2.4
Default Gateway	192.168.1.1	192.168.1.2	192.168.2.1	192.168.2.2
Subnet Mask	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0

Table 3-13: Network Layer Properties

NOTE: The subnet mask of all L3 Switch interfaces is set to 255.255.255.0

Step 3: Click on L3 Switch 1 and L3 Switch 2 and expand property panel on the right and set the properties as follows:

Switch	I/f1_Ethernet	I/f2_Ethernet	I/f3_Ethernet
	IP Address	IP Address	IP Address
L3 Switch 1	192.168.1.1	192.168.3.1	192.168.2.1
L3 Switch 2	192.168.1.2	192.168.3.2	192.168.2.2

Table 3-14: L3 Switch 1 and L3 Switch 2 properties

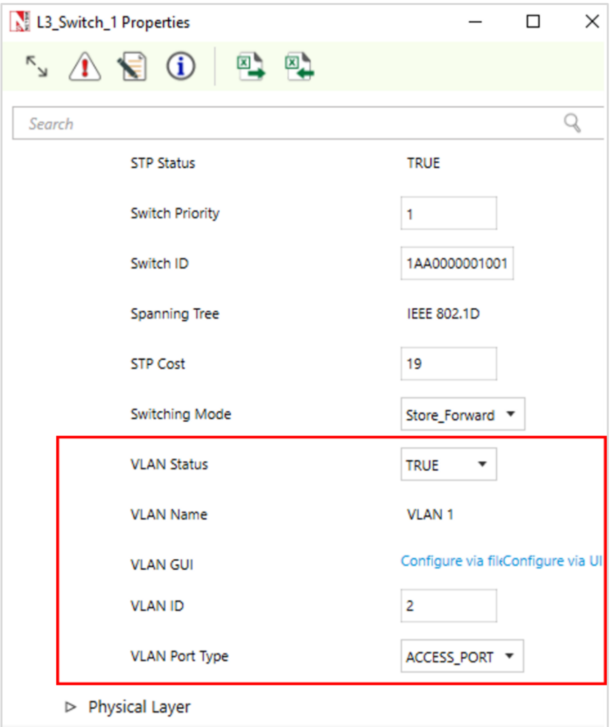


Figure 3-55: Datalink layer properties window

L3 Switch 1			
Interface ID	**VLAN Status**	**VLAN ID**	**VLAN Port Type**
Interface 1	TRUE	2	Access Port
Interface 2	TRUE	1	Trunk Port
Interface 3	TRUE	3	Access Port

Table 3-15: VLAN Properties for L3 Switch 1

L3 Switch 2			
Interface ID	**VLAN Status**	**VLAN ID**	**VLAN Port Type**
Interface 1	TRUE	2	Access Port
Interface 2	TRUE	1	Trunk Port
Interface 3	TRUE	3	Access Port

Table 3-16: VLAN Properties for L3 Switch 2

Step 4: In the INTERFACE (ETHERNET) > DATALINK LAYER Properties of L3 Switch 1, Click on **"Configure VLAN"** to view the properties for VLAN 2 set as per the screenshot shown below Figure 3-56.

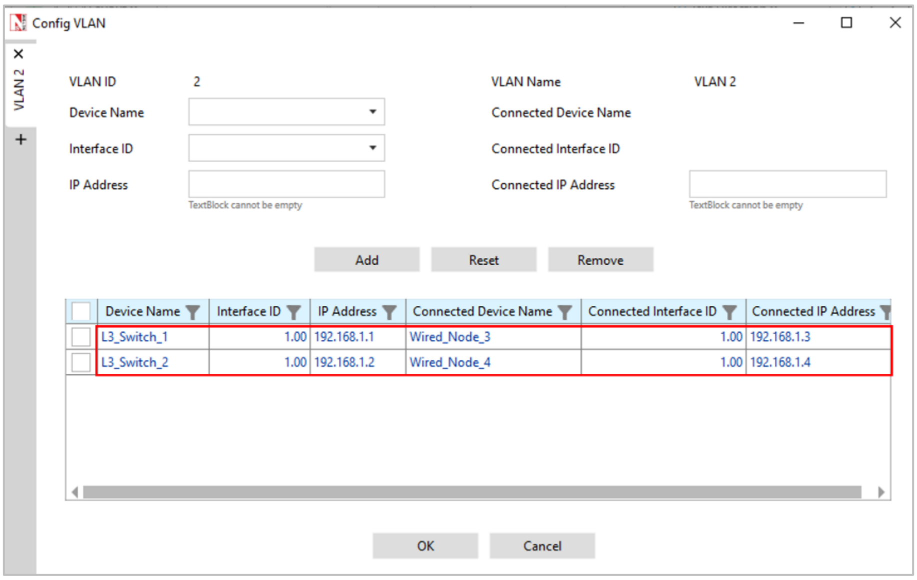


Figure 3-56: Configuring VLAN Properties in VLAN 2

Properties for VLAN 3 is set as per the below screenshot Figure 3-57.

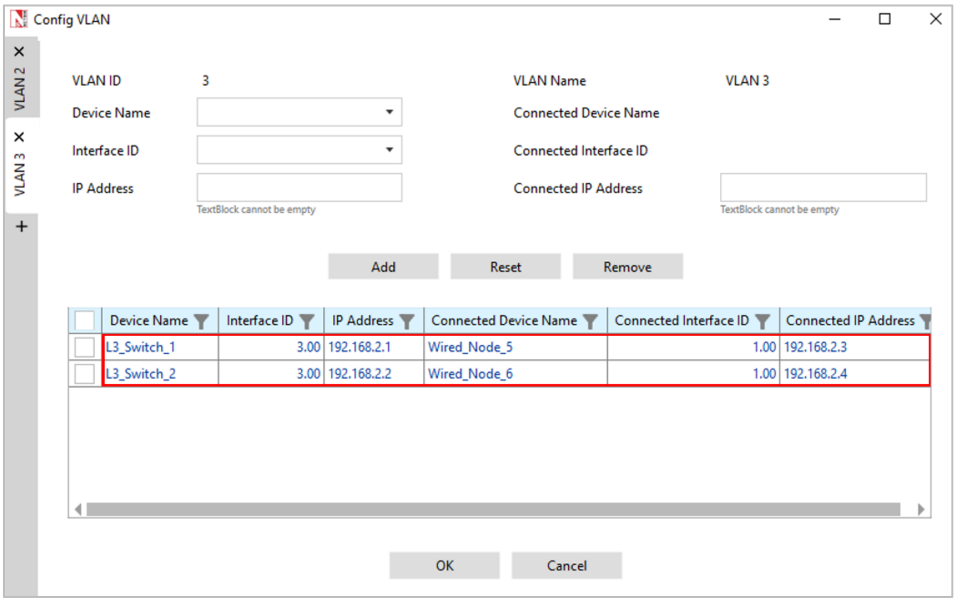


Figure 3-57: Configuring VLAN Properties in VLAN 3

After setting the properties of VLAN2 and VLAN3 click on OK.

Step 5: In the NETWORK LAYER Properties of L3 Switch 1, Enable - Static IP Route ->Click on **"Static IP Route"** to set static route as per the screenshot shown below.

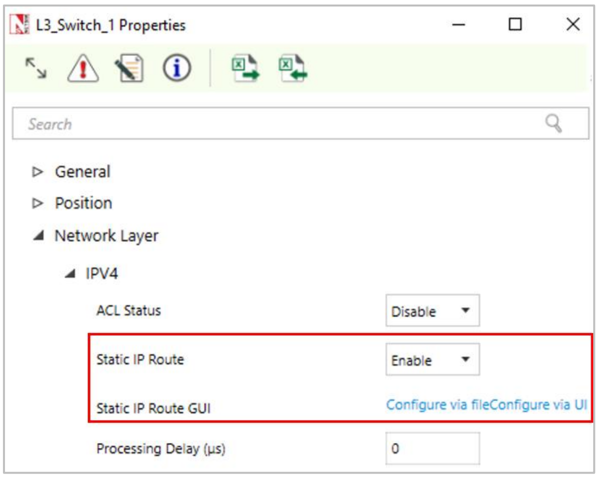


Figure 3-58: Select Configure Static Route IP

Set the properties in Static Route IP window as per the screenshot below and click on **Add**. Click on **OK.**

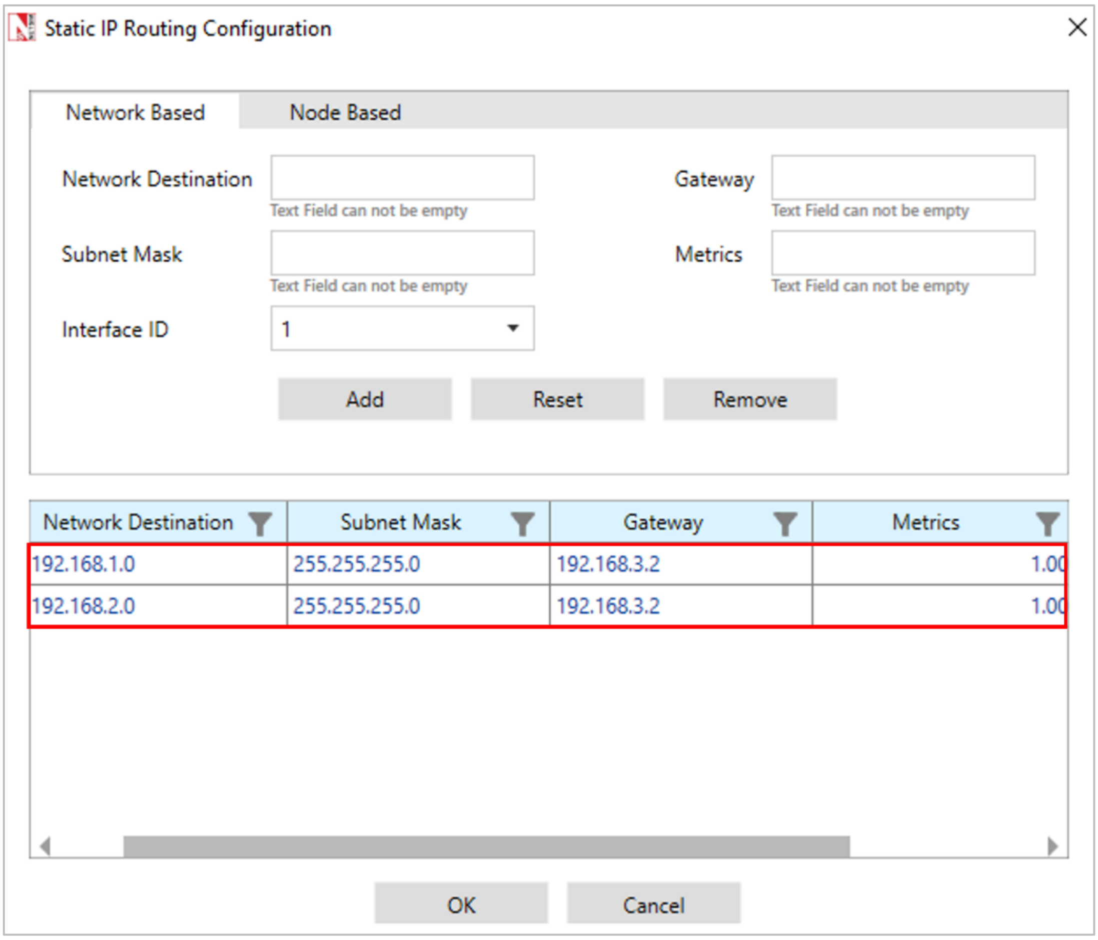


Figure 3-59: Configure Static route in Static Route IP window

NOTE: Transport Protocol is set to UDP in Application properties.

Step 6: Click on the "Configure Reports" tab in the top ribbon, enable the plots, run the simulation for 10 seconds and observe the throughputs.

3.5.4 Output

Throughput (Mbps)	
Application 1	0.58
Application 2	0.58

Table 3-17: Results Comparison

The above results conclude that trunking allows us to send or receive any VLAN information across the network.

3.6 Understanding the working of Public IP Address and Network Address Translation (NAT). (Level 2)

3.6.1 Theory

3.6.1.1 Public Address

A public IP address is assigned to every computer that connects to the Internet where each IP is unique. Hence there cannot exist two computers with the same public IP address all over the Internet. This addressing scheme makes it possible for the computers to “find each other” online and exchange information. User has no control over the IP address (public) that is assigned to the computer. The public IP address is assigned to the computer by the Internet Service Provider as soon as the computer is connected to the Internet gateway.

3.6.1.2 Private Address

An IP address is considered private if the IP number falls within one of the IP address ranges reserved for private networks such as a Local Area Network (LAN). The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private networks (local networks):

Class	Starting IP address	Ending IP address	No. of hosts
A	10.0.0.0	10.255.255.255	16,777,216
B	172.16.0.0	172.31.255.255	1,048,576
C	192.168.0.0	192.168.255.255	65,536

Table 3-18: Private IP address table

Private IP addresses are used for numbering the computers in a private network including home, school and business LANs in airports and hotels which makes it possible for the computers in the network to communicate with each other. For example, if a network A consists of 30 computers each of them can be given an IP starting from **192.168.0.1 to 192.168.0.30.**

Devices with private IP addresses cannot connect directly to the Internet. Likewise, computers outside the local network cannot connect directly to a device with a private IP. It is possible to interconnect two private networks with the help of a router or a similar device that supports Network Address Translation.

If the private network is connected to the Internet (through an Internet connection via ISP) then each computer will have a private IP as well as a public IP. Private IP is used for communication within the network whereas the public IP is used for communication over the Internet.

3.6.1.3 Network address translation (NAT)

A NAT (Network Address Translation or Network Address Translator) is the virtualization of Internet Protocol (IP) addresses. NAT helps to improve security and decrease the number of IP addresses an organization needs.

A device that is configured with NAT will have at least one interface to the inside network and one to the outside network. In a typical environment, NAT is configured at the exit device between a stub domain (inside network) and the backbone. When a packet leaves the domain, NAT translates the locally significant source address into a globally unique address. When a packet enters the domain, NAT translates the globally unique destination address into a local address. If more than one exit point exists, each NAT must have the same translation table. NAT can be configured to advertise to the outside world only one address for the entire network. This ability provides additional security by effectively hiding the entire internal network behind that one address. If NAT cannot allocate an address because it has run out of addresses, it drops the packet and sends an Internet Control Message Protocol (ICMP) host unreachable packet to the destination.

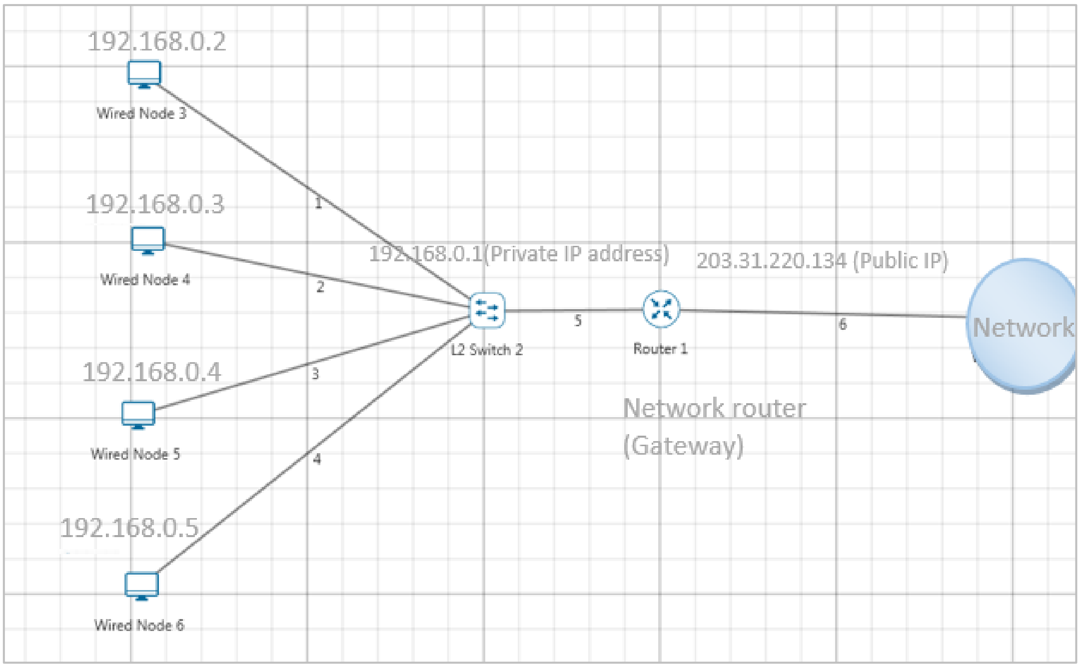


Figure 3-60: NAT implementation

NAT is secure since it hides network from the Internet. All communications from internal private network are handled by the NAT device, which will ensure all the appropriate translations are performed and provide a flawless connection between internal devices and the Internet.

In the above figure, a simple network of 4 hosts and one router that connects this network to the Internet. All hosts in the network have a private Class C IP Address, including the router's private interface (192.168.0.1), while the public interface that's connected to the Internet has a real IP Address (203.31.220.134). This is the IP address the Internet sees as all internal IP addresses are hidden.

3.6.2 Network Setup

Open NetSim and click on **Experiments> Advanced Routing> Understanding Public IP Address and NAT (Network Address Translation)** then click on the tile in the middle panel to load the example as shown in below Figure 3-61.

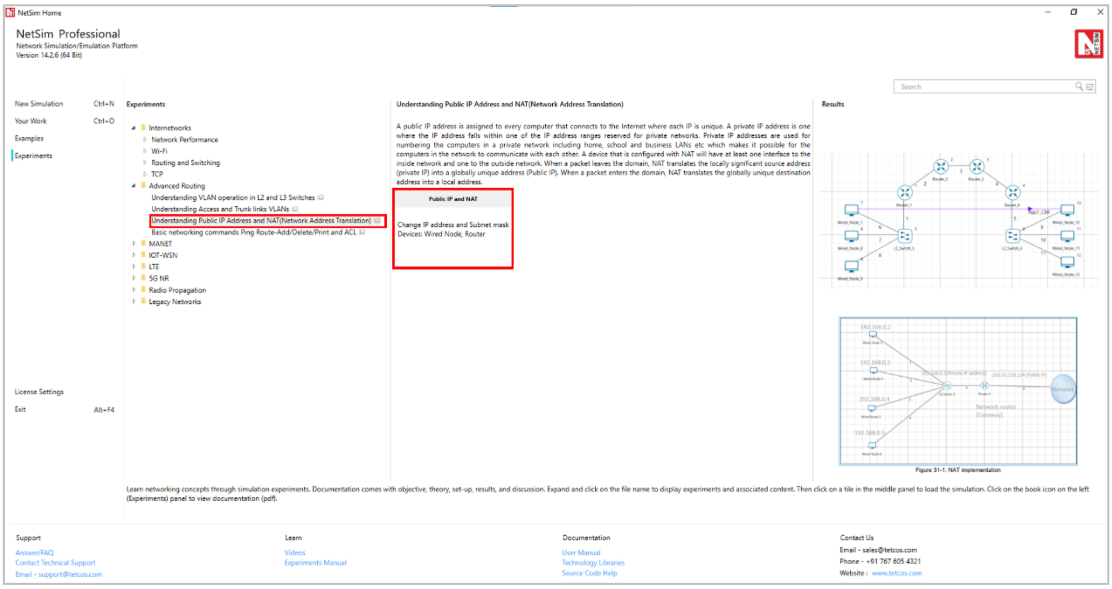


Figure 3-61: List of scenarios for the example of Understanding Public IP Address and NAT (Network Address Translation)

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 3-62.

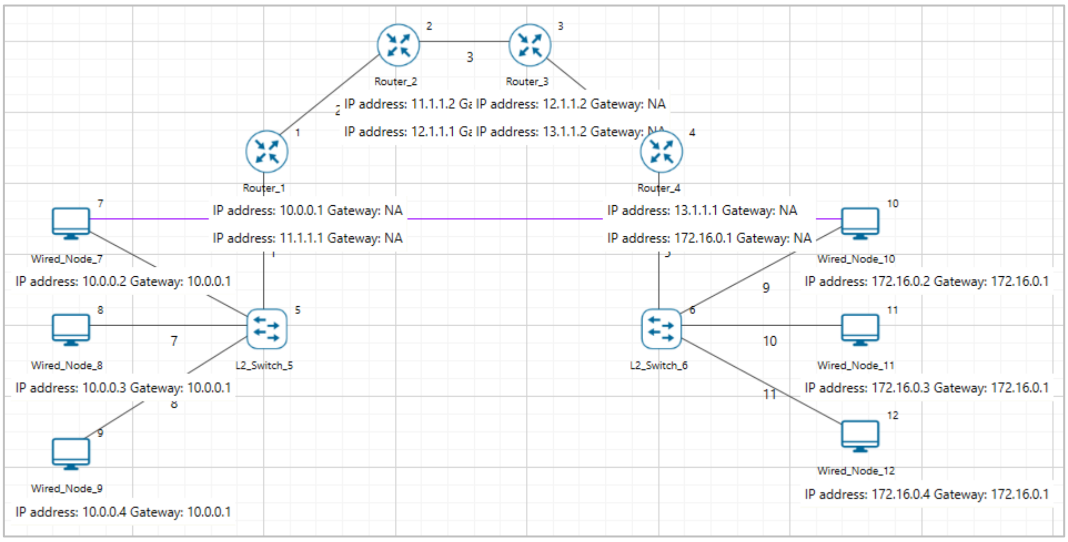


Figure 3-62: Network set up for studying the Understanding Public IP Address and NAT (Network Address Translation)

3.6.3 Procedure

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

Step 1: A network scenario is designed in NetSim GUI comprising of 6 Wired Nodes, 2 L2 Switches, and 4 Routers in the **“Internetworks”** Network Library.

Step 2: Click on Wired Nodes and then open right-side property panel. In the INTERFACE (ETHERNET) > NETWORK LAYER, the IP Address and the Subnet Mask are set as per the table given below Table 3-19.

Wired Node	IP address	Subnet mask
7	10.0.0.2	255.0.0.0
8	10.0.0.3	255.0.0.0
9	10.0.0.4	255.0.0.0
10	172.16.0.2	255.255.0.0
11	172.16.0.3	255.255.0.0
12	172.16.0.4	255.255.0.0

Table 3-19: IP Address and the Subnet mask for Wired nodes

Step 3: The IP Address and the Subnet Mask in Routers are set as per the table given below Table 3-20.

Router	Interface	IP address	Subnet mask
Router 1	Interface 2(WAN)	11.1.1.1	255.0.0.0
	Interface 1(Ethernet)	10.0.0.1	255.0.0.0
Router 2	Interface 1(WAN)	11.1.1.2	255.0.0.0
	Interface 2(WAN)	12.1.1.1	255.0.0.0
Router 3	Interface 1(WAN)	12.1.1.2	255.0.0.0
	Interface 2(WAN)	13.1.1.2	255.0.0.0
Router 4	Interface 1(WAN)	13.1.1.1	255.0.0.0
	Interface 2(Ethernet)	172.16.0.1	255.255.0.0

Table 3-20: IP Address and the Subnet Mask for Routers

Step 4: Configure an application between any two nodes by selecting a CBR application from Wired Node 7 i.e., Source to Wired Node 10 i.e., Destination from Set Traffic tab in the ribbon. Click on the application, expand the right-side property panel and Set Packet Size: 1460 Bytes, Inter Arrival Time remaining 20000µs

Additionally, the “Start Time(s)” parameter is set to 50(Figure 3-57), while configuring the application. This time is usually set to be greater than the time taken for OSPF Convergence (i.e., Exchange of OSPF information between all the routers), and it increases as the size of the network increases.

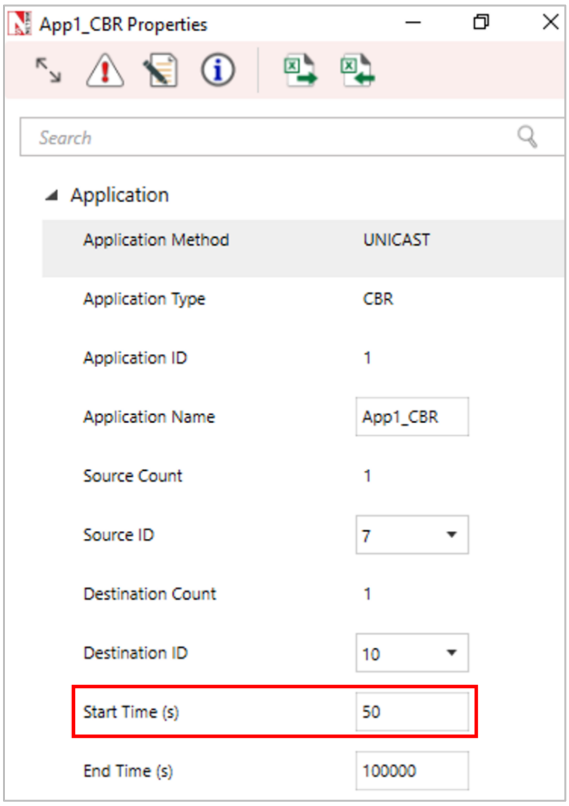


Figure 3-63: Application Properties Window

Step 5: Click on Configure reports tab in ribbon on the top and enable packet trace. Packet Trace can be used for packet level analysis.

Step 6: Click on the "Configure Reports" tab in the top ribbon, enable the plots, run the simulation for 100 seconds.

3.6.4 Output

After simulation Open Packet Trace from the Simulation Results window and filter Packet ID to 1 as shown below.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	SOURCE_IP	DESTINATION_IP	GATEWAY_IP	NEXT_HOP_IP
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	10.0.0.1	10.0.0.2	10.0.0.1
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	10.0.0.1	10.0.0.2	10.0.0.1
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	13.1.1.1	11.1.1.1	11.1.1.2
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	13.1.1.1	12.1.1.1	12.1.1.2
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	13.1.1.1	13.1.1.2	13.1.1.1
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	172.16.0.2	172.16.0.1	172.16.0.2
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	172.16.0.2	172.16.0.1	172.16.0.2

Figure 3-64: Packet Trace

SOURCE IP – source node IP (Node)

DESTINATION IP – gateway IP/ destination IP (Router/ Node)

GATEWAY IP – IP of the device which is transmitting a packet (Router/ Node)

NEXT HOP IP – IP of the next hop (Router/ Node)

Source node 7 (10.0.0.2) wouldn't know how to route to the destination and hence its default gateway is Router 1 with interface IP (10.0.0.1). So, the first line in the above screenshot specifies packet flow from Source Node 7 to L2 Switch 6 with SOURCE_IP (10.0.0.2), DESTINATION IP (10.0.0.1), GATEWAY IP (10.0.0.2) and NEXT HOP IP (10.0.0.1). Since Switch is Layer2 device there is no change in the IPs in second line. Third line specifies the packet flow from Router 1 to Router 2 with SOURCE IP (10.0.0.2), DESTINATION_IP (13.1.1.1- IP of the router connected to destination. Since OSPF is running, the router is looks up the route to its destination from routing table), GATEWAY IP (11.1.1.1) and NEXT HOP IP (11.1.1.2) and so on.

3.6.5 Exercises

1. Create a scenario different from the one in the experiment. It should consist of 3 Local Area Networks (LANs). Each LAN can have 1 switch and 2 nodes. Connect each switch to a Router and interconnect the routers. This completes the network set-up. Next, configure LAN1 to have class-A IP addresses, the LAN2 have class-B IP addresses and the LAN3 have class-C IP addresses. Finally, configure the following data traffic flows (i) from a node in LAN1 to a node in LAN2 (ii) From a node in LAN2 to a node in LAN3 (iii) and from a node in LAN3 to a node in LAN1.

   Post simulation, using the packet trace explain the Source IP, Destination IP, Gateway IP and Next hop IP for each of the traffic flows.

3.7 M/D/1 and M/G/1 Queues (Level 3)

3.7.1 Motivation

In this simulation experiment, we will study a model that is important to understand the queuing and delay phenomena in packet communication links. Let us consider the network shown in Figure 3-66. Wired Node 1 is transmitting UDP packets to Wired Node 2 through a router. Link 1 and Link 2 are of speed 10 Mbps. The packet lengths are 1250 bytes plus a 54-byte header, so that the time taken to transmit a packet on each 10 Mbps link is $\frac{1304 \times 8}{10}$ μsec =1043.2 μsec. In this setting, we would like answers to the following questions:

1. We notice that the maximum rate at which these packets can be carried on a 10 Mbps link is $\frac{10^6}{1043.2} = 958.59$ packets per second. Can the UDP application send packets at this rate?
2. The time taken for a UDP packet to traverse the two links is $2 \times 1043.2 = 2086.4$ μsec. Is this the time it actually takes for a UDP packet generated at Wired Node 1 to reach Wired Node 2.

The answer to these questions depends on the manner in which the UDP packets are being generated at Wired Node 1. If the UDP packets are generated at intervals of 1043.2 μsec then successive packets will enter the Link 1, just when the previous packet departs. In practice, however, the UDP packets will be generated by a live voice or video source. Depending on the voice activity, the activity in the video scene, and the coding being used for the voice and the video, the rate of generation of UDP packets will vary with time. Suppose two packets were generated during the time that one packet is sent out on Link 1, then one will have to wait, giving rise to queue formation. This also underlines the need for a buffer to be placed before each link; a buffer is just some dynamic random-access memory in the link interface card into which packets can be stored while waiting for the link to free up.

Queuing models permit us to understand the phenomenon of mismatch between the service rate (e.g., the rate at which the link can send out packets) and the rate at which packets arrive. In the network in Figure 3-66, looking at the UDP flow from Wired Node 1 to Wired Node 2, via Router 3, there are two places at which queueing can occur. At the interface between Wired Node 1 and Link 1, and at the interface between Router 3 and Link 2. Since the only flow of packets is from Wired Node 1 to Wired Node 2, all the packets entering Link 2 are from Link 1, and these are both of the same bit rate. Link 2, therefore, cannot receive packets faster than it can serve them and, at any time, only the packet currently in transmission will be at Link 2. On the other hand at the Wired Node 1 to Link 1 interface, the packets are generated directly by the application, which can be at arbitrary rates, or inter-packet times.

Suppose that, at Wired Node 1, the application generates the successive packets such that the time intervals between the successive packets being generated are statistically independent, and the probability distribution of the time intervals has a negative exponential density, i.e., of the form $\lambda\, e^{-\lambda x}$, where λ (packets per second) is a parameter, called the *rate* parameter, and x (seconds) is the argument of the density. The application generates the entire packet *instantaneously,* i.e., all the bits of the packet arrive from the application together, and enter the buffer at Link 1, to wait behind the other packets, in a first-in-first-out manner. The resulting random process of the points at which packets enter the buffer of Link 1 is called a Poisson Process of rate λ packets per second. The buffer queues the packets while Link 1 serves them with *service time* $b = 1043.2\ \mu$sec. Such a queue is called an M/D/1 queue, where the notation is to be read as follows.

- The M before the first slash (denoting "Markov") denotes the Poisson Process of instants at which packets enter the buffer.
- The D between the two slashes (denoting "Deterministic") denotes the fixed time taken to serve each queued packet.
- The 1 after the second slash denotes that there is just a single server (Link 1 in our example)

This way of describing a single server queueing system is called Kendall's Notation.

In this experiment, we will understand the M/D/1 model by simulating the above-described network on NetSim. The M/D/1 queueing model, however, is simple enough that it can be mathematically analyzed in substantial detail. We will summarize the results of this analysis in the next section. The simulation results from NetSim will be compared with the analytical results.

3.7.2 Mathematical Analysis of the M/D/1 Queue

The M/D/1 queueing system has a random number of arrivals during any time interval. Therefore, the number of packets waiting at the buffer is also random. It is possible to mathematically analyze the *random process* of the number of waiting packets. The procedure for carrying out such analysis is, however, beyond the scope of this document. We provide the final formulas so that the simulation results from NetSim can be compared with those provided by these formulas.

As described earlier, in this chapter, the M/D/1 queue is characterized by two parameters: λ (packets per second), which is the arrival rate of packets into the buffer, and μ (packets per second), which is the rate at which packets are removed from a nonempty queue. Note that $1/\mu$ is the service time of each packet.

Define $\rho = \lambda \times \frac{1}{\mu} = \lambda/\mu$. We note that ρ is the average number of packets that arrive during the service time of a packet. Intuitively, it can be expected that if $\rho > 1$ then packets arrive faster than the rate at which they can be served, and the queue of packets can be expected grow without bound. When $\rho < 1$ we can expect the queue to be "stable." When $\rho = 1$, the service rate is exactly matched with the arrival rate; due to the randomness, however, the queue can still grown without bound. The details of this case are beyond the scope of this document.

For the k^{th} arriving packet, denote the instant of arrival by a_k, the instant at which service for this packet starts as s_k, and the instant at which the packet leaves the system as d_k. Clearly, for all $k, d_k - s_k = \frac{1}{\mu}$, the deterministic service time. Further define, for each k,

$$W_k = s_k - a_k$$

$$T_k = d_k - a_k$$

i.e., W_k is called the *queuing delay,* i.e., time from the arrival of the k^{th} packet until it starts getting transmitted, whereas T_k is called the *total delay,* i.e., the time from the arrival of the k^{th} packet until its transmission is completed. Considering a large number of packets, we are interested in the average of the values $W_1, W_2, W_3, \cdots,$ i.e., the *average queueing time* of the packets. Denote this average by W. By mathematical analysis of the packet queue process, it can be shown that for an M/D/1 queueing system,

$$W = \frac{1}{2\mu} \times \frac{\rho}{1-\rho}$$

Denoting by T, the average total time in the system (i.e., the average of $T_1, T_2, T_3, \cdots$), clearly

$$T = W + \frac{1}{\mu}.$$

Observe the following from the above formula:

1. As ρ approaches 0, W becomes 0. This is clear, since, when the arrival rate becomes very small, and arriving packet sees a very small queue. For arrival rate approaching 0, packets get served immediately on arrival.
2. As ρ increases, W inreases.
3. As ρ approaches 1 (from values smaller than 1), the mean delay goes to ∞.

We will verify these observations in the NetSim simulation.

3.7.3 The Experimental Setup

Open NetSim and click on **Experiments> Internetworks> Network Performance> MD1 and MG1 Queues** then click on the tile in the middle panel to load the example as shown in below Figure 3-65.

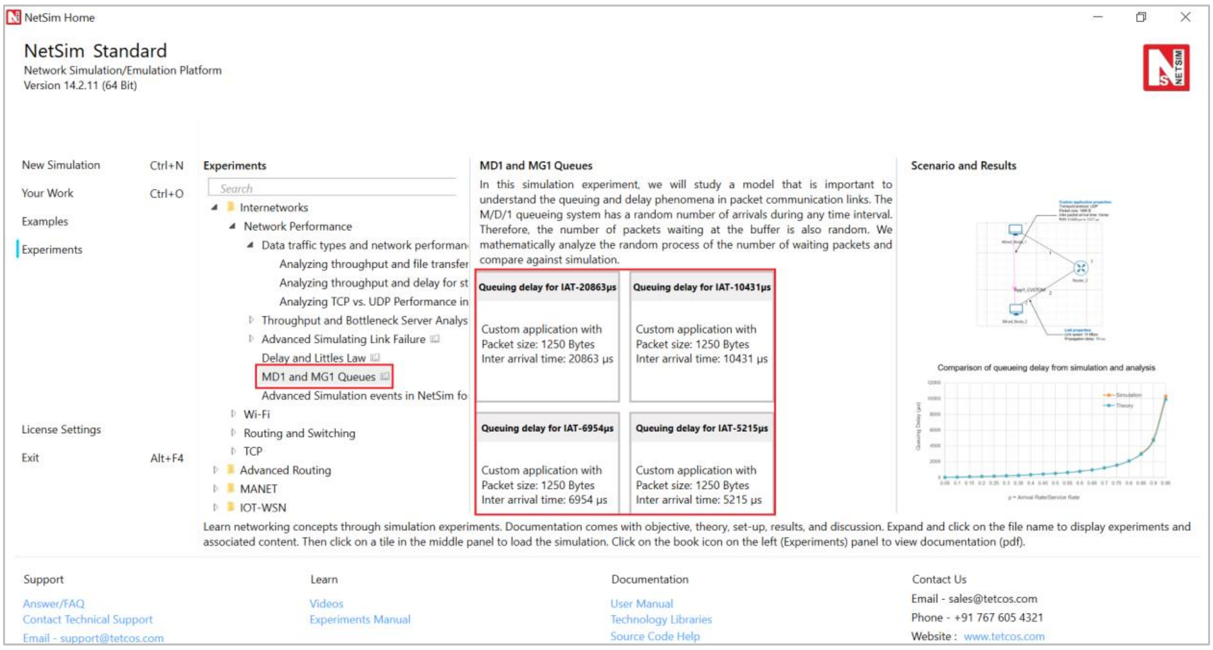


Figure 3-65: List of scenarios for the example of MD1 and MG1 Queues

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

The model described at the beginning of this chapter is shown in Figure 3-66.

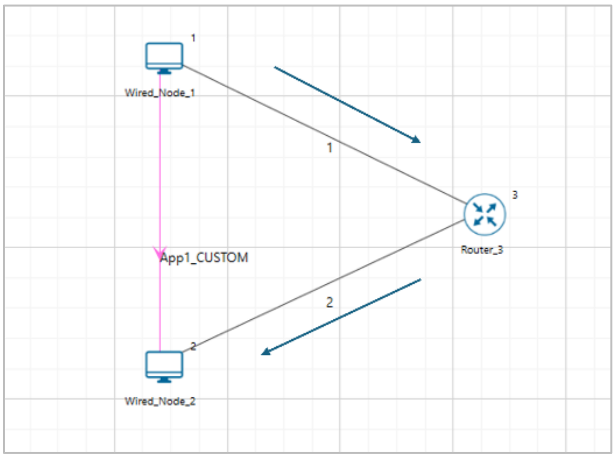


Figure 3-66: A single wired node (Wired Node 1) sending UDP packets to another wired node (Wired Node 2) through a router (Router 3). The packet interarrival times at Wired Node 1 are exponentially distributed, and packets are all of the same length, i.e., 1250 bytes plus UDP/IP header.

3.7.4 Procedure

Queuing delay for IAT-20863 (μs) **Sample:**

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 1 Router in the **"Internetworks"** Network Library.

Step 2: Link Properties are set as per the table given below Table 3-21.To set the link properties, click on the link and expand property panel on right and configure as mentioned.

Link Properties	Link	Link
Uplink Speed (Mbps)	10	10
Downlink Speed (Mbps)	10	10
Uplink BER	0	0
Downlink BER	0	0
Uplink Propagation Delay (µs)	0	0
Downlink Propagation Delay (µs)	0	0

Table 3-21: Wired link properties

Step 3: Configure Custom application between Wired node 1 to Wired node 2 by clicking on set traffic tab from the ribbon on the top. To set the application properties, click on the application and set the **Transport Protocol** to UDP, **Packet Size** to 1250 bytes, **Distribution** to exponential, and **Mean** to 20863 µs.

The Packet Size and Inter Arrival Time parameters are set such that the Generation Rate equals 0.479 Mbps. Generation Rate can be calculated using the formula:

$$Generation\ Rate\ (Mbps)\ =\ Packet\ Size\ (Bytes)\ *\ 8/Interarrival\ time\ (\mu s)$$

Step 4: Packet Trace is enabled by clicking on the configure reports tab on the top. At the end of the simulation, a very large .csv file containing all the packet information is available for the users to perform packet level analysis.

Step 5: Run the Simulation for 100 Seconds.

Similarly, the other samples are created by changing the Inter Arrival Time per the formula

$$IAT = \frac{10^6}{958.59 * \rho}$$

as per the table given below Table 3-22.

P	IAT (μs)
0.05	20863
0.1	10431
0.15	6954
0.2	5215
0.25	4172
0.3	3477
0.35	2980
0.4	2607
0.45	2318
0.5	2086
0.55	1896
0.6	1738
0.65	1604
0.7	1490
0.75	1390
0.8	1303
0.85	1227
0.9	1159
0.95	1098

Table 3-22: Inter Arrival Time Settings

Even though the packet size at the application layer is 1250 bytes, as the packet moves down the layers, overhead is added. The overheads added in different layers are shown in the below table and can be obtained from the packet trace:

Layer	Overhead (Bytes)
Transport Layer	8
Network Layer	20
MAC layer	26
Physical Layer	0
Total	54

Table 3-23: Overheads added to a packet as it flows down the network stack

3.7.5 Obtaining the Mean Queuing delay from the Simulation Output

After running the simulation, note down the “Mean Delay” from the Application Metrics in NetSim results window. This is the average time between the arrival of packets into the buffer at Wired Node 1, and their reception at Wired Node 2.

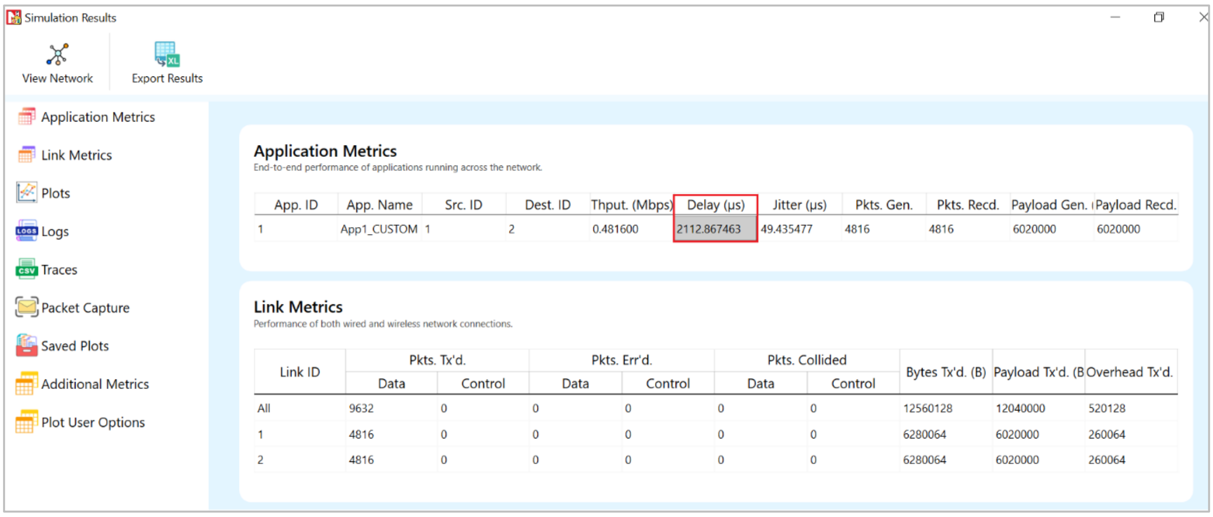


Figure 3-67: Observing Delay value from the NetSim simulation results window

As explained in the beginning of this chapter, for the network shown in Figure 3-66, the end-to-end delay of a packet is the sum of the queueing delay at the buffer between the wired-node and Link 1, the transmission time on Link 1, and the transmission time on Link 2 (there being no queueing delay between the Router and Link_2). It follows that.

$$Mean\ Delay = \left(\frac{1}{2\mu} \times \frac{\rho}{1-\rho}\right) + \frac{1}{\mu} + \frac{1}{\mu}$$

Note: The Simulation results are calculated from the Packet Trace whereas the theoretical results are calculated using the formula.

3.7.6 Output Table

Sample	ρ	λ	Mean Delay (μs)	Queuing Delay (μs) (Simulation)	Queuing Delay (μs) (Theory)
1	0.05	47.93	2113.76	27.36	27.45
2	0.10	95.86	2144.45	58.05	57.96
3	0.15	143.79	2179.57	93.17	92.05
4	0.20	191.72	2219.15	132.75	130.40
5	0.25	239.65	2260.54	174.14	173.87
6	0.30	287.58	2311.43	225.03	223.54
7	0.35	335.51	2368.29	281.88	280.86
8	0.40	383.44	2439.03	352.63	347.73
9	0.45	431.37	2518.15	431.74	426.76
10	0.50	479.30	2613.99	527.59	521.60
11	0.55	527.22	2728.85	642.43	637.51
12	0.60	575.15	2874.69	788.29	782.40
13	0.65	623.08	3066.09	979.68	968.68
14	0.70	671.01	3323.56	1237.48	1217.07
15	0.75	718.94	3661.65	1575.23	1564.80
16	0.80	766.87	4206.31	2119.91	2086.40
17	0.85	814.80	5205.57	3119.14	2955.73
18	0.90	862.73	7161.82	5075.40	4694.39

19	0.95	910.66	13274.34	11167.07	9910.39

Table 3-24: Mean Delay, Queueing delay from Simulation and Queueing delay from analysis

Comparison Chart

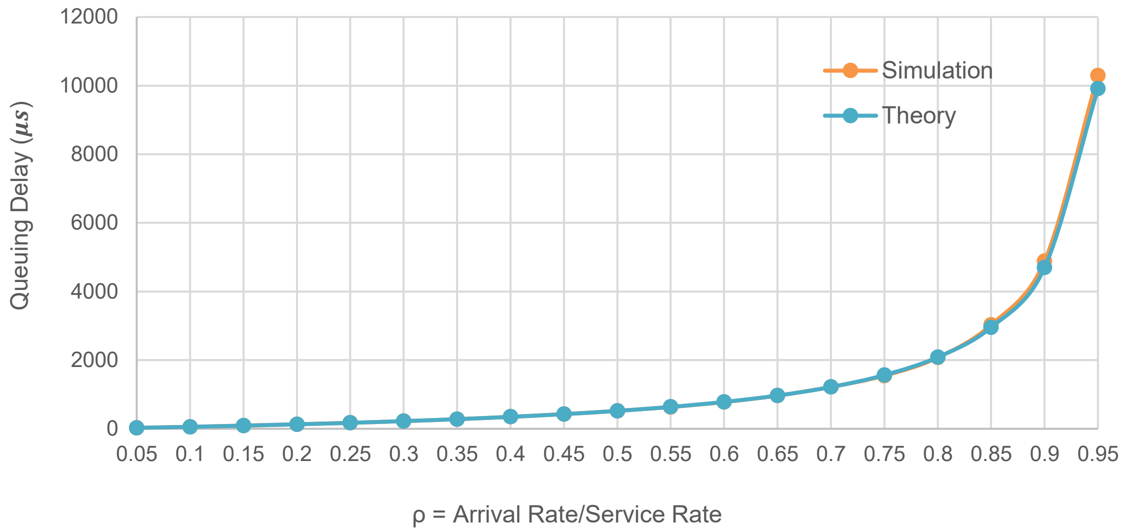


Figure 3-68: Comparison of queueing delay from simulation and analysis

3.7.7 Advanced Topic: The M/G/1 Queue

In Section 3.7.1, we introduced the M/D/1 queue. Successive packets were generated instantly at exponentially distributed time intervals (i.e., at the points of a Poisson process); this gave the "M" in the notation. The packets were all of fixed length; this gave the "D" in the notation. Such a model was motivated by the transmission of packetized voice over a fixed bit rate wireline link. The voice samples are packetized into constant length UDP packets. For example, typically, 20ms of voice samples would make up a packet, which would be emitted at the instant that the 20ms worth of voice samples are collected. A voice source that is a part of a conversation would have listening periods, and "silence" periods between words and sentences. Thus, the intervals between emission instants of successive UDP packets would be random. A simple model for these random intervals is that they are exponentially distributed, and independent from packet to packet. This, formally, is called the Poisson point process. With exponentially distributed (and independent) inter-arrival times, and fixed length packets we obtain the M/D/1 model. On the other hand, some applications, such as video, generate unequal length packets. Video frames could be encoded into packets. To reduce the number of bits being transmitted, if there is not much change in a frame, as compared to the previous one, then the frame is encoded into a small number of bits; on the other hand if there is a large change then a large number of bits would need to be used to encode the new information in the frame. This motivates variable packet sizes. Let us suppose that, from such an application, the packets arrive at the points of a Poisson process of rate λ, and that the randomly varying packet transmission times can be modelled as independent and identically distributed random variables, $B_1, B_2, B_3, \cdots$, with mean b and second moment $b^{(2)}$, i.e., variance $b^{(2)} - b^2$. Such a model is denoted by M/G/1, where M denotes the Poisson arrival process, and G ("general") the "generally" distributed service times. Recall the notation M/D/1 (from earlier in this section), where the D denoted fixed (or "deterministic") service times. Evidently, the M/D/1 model is a special case of the M/G/1 model.

Again, as defined earlier in this section, let W denote the mean queueing delay in the M/G/1 system. Mathematical analysis of the M/G/1 queue yields the following formula for W

$$W = \frac{\rho}{1 - \rho} \frac{b^{(2)}}{2b}$$

where, as before, $\rho = \lambda b$. This formula is called the Pollacek-Khinchine formula or P-K formula, after the researchers who first obtained it. Denoting the variance of the service time by $Var(B)$, the P-K formula can also be written as

$$W = \frac{\rho b}{2(1-\rho)} \left(\frac{Var(B)}{b^2} + 1\right)$$

Applying this formula to the M/D/1 queue, we have $Var(B) = 0$. Substituting this in the M/G/1 formula, we obtain.

$$W = \frac{\rho}{1-\rho} \frac{b}{2}$$

which, with $b = 1/\mu$, is exactly the M/D/1 mean queuing delay formula displayed earlier in this section.

3.7.8 A NetSim Exercise Utilising the M/G/1 Queue

In this section we demonstrate the use of the M/G/1 queueing model in the context of the network setup shown in Figure 3-66. The application generates exponentially distributed data segment with mean d bits, i.e., successive data segment lengths are sampled independently from an exponential distribution with rate parameter $\frac{1}{d}$. Note that, since packets are integer multiples of bits, the exponential distribution will only serve as an approximation. These data segments are then packetized by adding a constant length header of length h bits. The packet generation instants form a Poisson process of rate λ. Let us denote the link speed by c. Let us denote the random data segment length by X and the packet transmission time by B, so that

$$B = \frac{X + h}{c}$$

Denoting the mean of B by b, we have

$$b = \frac{d + h}{c}$$

Further, since h is a constant,

$$Var(B) = Var(X)/c^2$$

These can now be substituted in the P-K formula to obtain the mean delay in the buffer between Node 1 and Link 1.

We set the mean packet size to 100B or 800 bits, the header length h = 54B or 432 bits and $\lambda =$ 5000

For a 10Mbps link, the service rate $\mu = \frac{10*10^6}{154\times 8} = 8116.8$

Using the Pollaczek–Khinchine (PK) formula, the waiting time for a M/G/1 queuing system is

$$w = \frac{\rho + \lambda \times \mu \times Var(s)}{2(\mu - \lambda)}$$

Where $var(s)$ is the variance of the service time distribution S. Note that

$var(s) = \frac{1}{(\mu')^2}$ where μ' is the mean service time of the exponential random variable (100B packets and not 154B)

$$\mu' = \frac{10 \times 10^6}{100 * 8} = 12500$$

Hence substituting into the PK formula, one gets

$$w = \frac{0.4 + \frac{(3467.7\ \times 8116.8)}{12500^2}}{2\ (8116.8 - 3246.7)} = 59.5\ \mu s$$

By simulation the queuing delay is 60.5 μs.

The queuing delay is not available in the NetSim results dashboard. It can be got from the packet trace. It is the average of (PHY layer Arrival time – APP layer arrival time) for packets being sent from Node 1.

3.8 Understand the working of OSPF and SPF (Level 3)

NOTE: NetSim Academic supports a maximum of 20 routers and hence this experiment cannot be done with NetSim Academic. NetSim Standard/Pro would be required to simulate this configuration.

3.8.1 Objective

To understand the working of OSPF and Shortest Path First (SPF) tree creation.

3.8.2 Theory

OSPF

Open Shortest Path First (OSPF) is an Interior Gateway Protocol (IGP) standardized by the Internet Engineering Task Force (IETF) and commonly used in large Enterprise networks. OSPF is a link-state routing protocol providing fast convergence and excellent scalability. Like all link-state protocols, OSPF is very efficient in its use of network bandwidth.

Shortest path First Algorithm

OSPF uses a shorted path first algorithm to build and calculate the shortest path to all known destinations. The shortest path is calculated with the use of the Dijkstra algorithm. The algorithm by itself is quite complicated. This is a very high level, simplified way of looking at the various steps of the algorithm:

- Upon initialization or due to any change in routing information, a router generates a link-state advertisement. This advertisement represents the collection of all link-states on that router.
- All routers exchange link-states by means of flooding. Each router that receives a link-state update should store a copy in its link-state database and then propagate the update to other routers.
- After the database of each router is completed, the router calculates a Shortest Path Tree to all destinations. The router uses the Dijkstra algorithm in order to calculate the shortest path tree. The destinations, the associated cost and the next hop to reach those destinations form the IP routing table.
- In case no changes in the OSPF network occur, such as cost of a link or a network being added or deleted, OSPF should be very quiet. Any changes that occur are communicated through link-state packets, and the Dijkstra algorithm is recalculated in order to find the shortest path.

The algorithm places each router at the root of a tree and calculates the shortest path to each destination based on the cumulative cost required to reach that destination. Each router will have its own view of the topology even though all the routers will build a shortest path tree using the same link-state database.

Example

Refer from OSPF RFC 2328 (https://tools.ietf.org/html/rfc2328#section-2.3).txt

The below network shows a sample map of an Autonomous System.

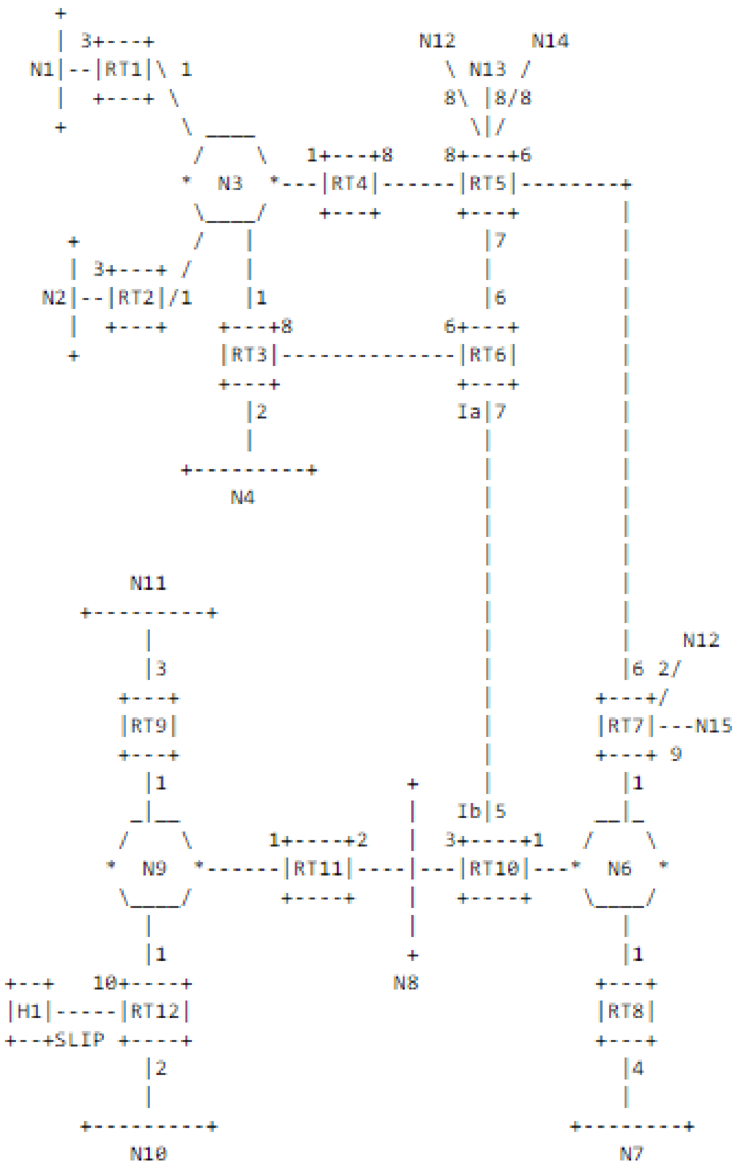


Figure 3-69: Sample maps of an Autonomous system

A cost is associated with the output side of each router interface. This cost is configurable by the system administrator. The lower the cost, the more likely the interface is to be used to forward data traffic. Costs are also associated with the externally derived routing data (e.g., the BGP-learned routes).

The directed graph resulting from the above network is depicted in the following table. Arcs are labelled with the cost of the corresponding router output interface. Arcs having no labelled cost have a cost of 0. Note that arcs leading from networks to routers always have cost 0.

	FROM																
		RT1	RT2	RT3	RT4	RT5	RT6	RT7	RT8	RT9	RT 10	RT 11	RT 12	N3	N6	N8	N9
	RT1													0			
	RT2													0			
	RT3						6							0			
	RT4					8								0			
	RT5				8		6	6									
	RT6			8		7											
	RT7					6									0		
	RT8														0		
	RT9																0
	RT10						7								0	0	
	RT11															0	0
	RT12																0
	N1	3															
	N2		3														
	N3	1	1	1	1												
	N4			2													
	N5																
TO	N6							1	1		1						
	N7								4								
	N8										3	2					
	N9									1		1	1				
	N10												2				
	N11									3							
	N12					8		2									
	N13					8											
	N14					8											
	N15							9									
	H1												10				

Table 3-25: Directed graph

A router generates its routing table from the above directed graph by calculating a tree of shortest paths with the router itself as root. Obviously, the shortest-path tree depends on the router doing the calculation. The shortest-path tree for Router RT6 in our example is depicted in the following figure.

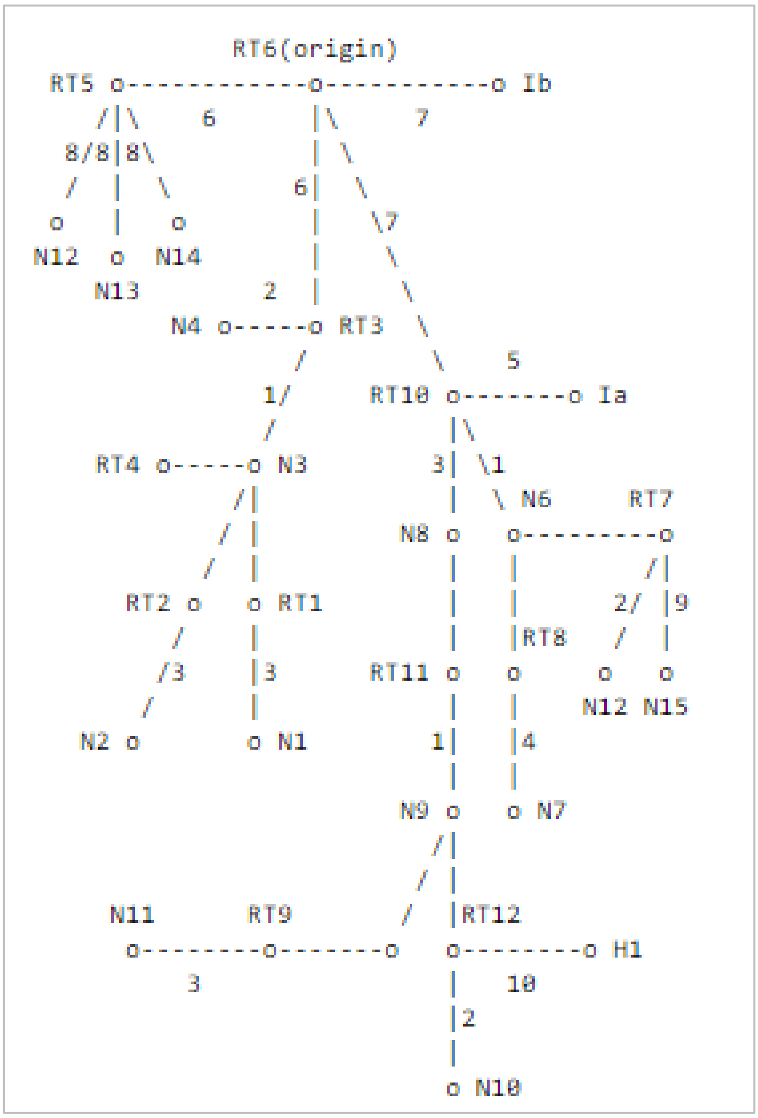


Figure 3-70: SPF tree for Router 6

Routing Table

The IP forwarding table formed in the routers and nodes can be accessed from the IP Forwarding Table list present in the Simulation Results window as shown below:

The tree gives the entire path to any destination network or host. However, only the next hop to the destination is used in the forwarding process. Note also that the best route to any router has also been calculated. For the processing of external data, we note the next hop and distance to any router advertising external routes. The resulting routing table for Router RT6 is shown in the following table.

Destination	IP Address	Next hop	Distance
N1	11.0.0.130	RT3	10
N2	11.0.0.138	RT3	10
N3	11.0.0.2	RT3	7
N4	11.0.0.170	RT3	8
N6	11.0.0.66	RT10	8
N7	11.0.0.194	RT10	12
N8	11.0.0.106	RT10	10
N9	11.0.0.82	RT10	11
N10	11.0.0.202	RT10	13
N11	11.0.0.211	RT10	14
H1	11.0.0.227	RT10	21
RT5	11.0.0.34	RT5	6
RT7	11.0.0.58	RT10	8
N12	11.0.0.147	RT10	10
N13	11.0.0.154	RT5	14
N14	11.0.0.162	RT5	14
N15	11.0.0.186	RT10	17

Table 3-26: Routing Table for RT6

Distance calculation

RT6 has 3 interfaces i.e., RT3, RT5 and RT10. The distance obtained is 10 for destination N1 via RT3 interface. The packets from RT6 would reach N1 via RT3, N3 and RT1. The cost assigned to routers in this path is 6+1+0+3 = 10 (cost can be seen in SPF tree for RT6).

3.8.3 Network Setup

Open NetSim and click on **Experiments> Internetworks> Routing and Switching> Understand the working of OSPF** then click on the tile in the middle panel to load the example as shown in below Figure 3-71.

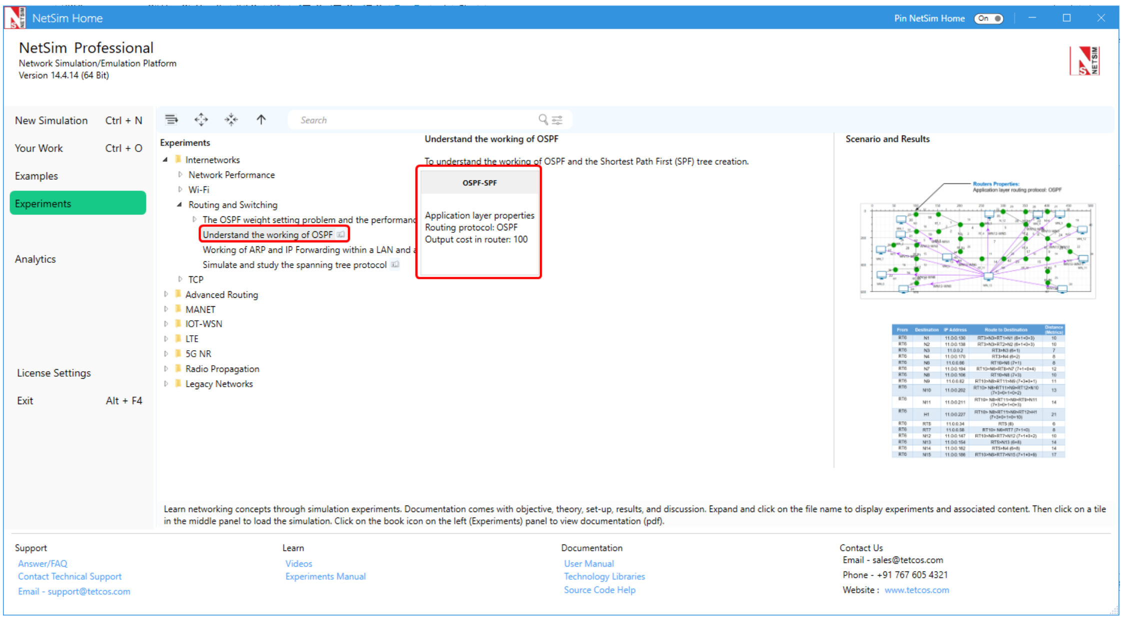


Figure 3-71: List of scenarios for the example of Understand the working of OSPF

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 3-72.

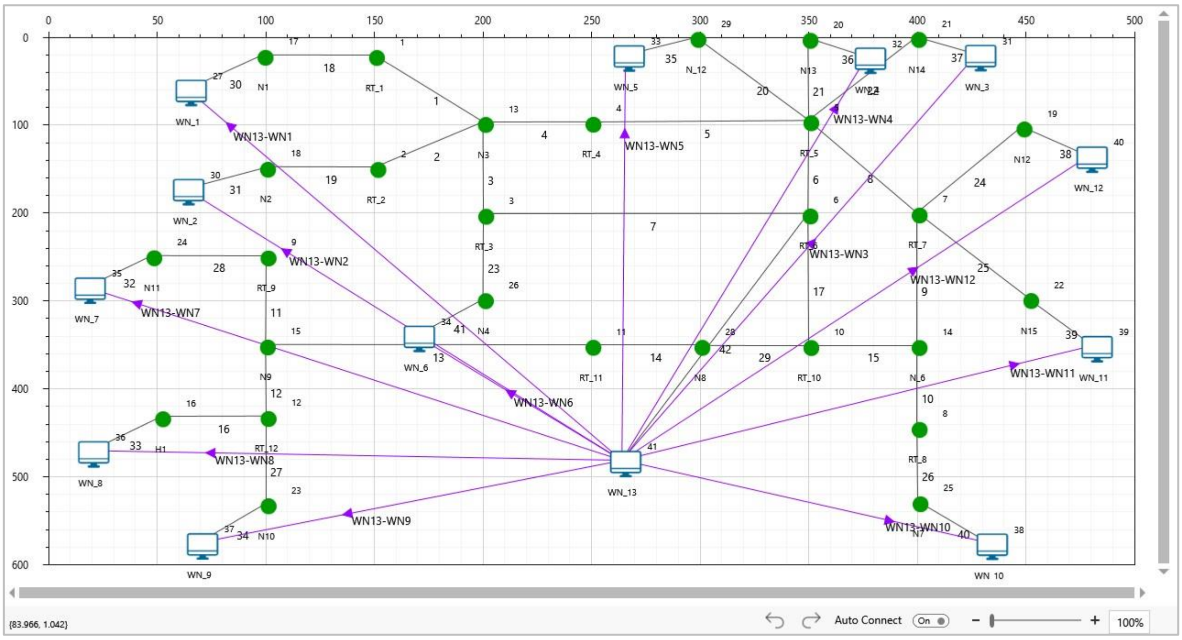


Figure 3-72: Network topology created in NetSim and it is similar to the network as per the OSPF RFC 2328

3.8.4 Procedure

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

Step 1: A network scenario is designed in NetSim GUI comprising of 28 Routers and 13 Wired Nodes in the **"Internetworks"** Network Library.

Step 2: The Output Cost for all the Routers in the network is set in Applications > OSPF > WAN Interface as per Table 3-25.

Step 3: Configure a **CBR** application with all the destination nodes by selecting CBR application icon from Set Traffic tab. Right click on the application, open properties as a new window and set the start time of all the applications to 30 seconds.

Step 4: Packet Trace is enabled in the NetSim GUI, and hence we can track the route in which the packets have chosen to reach the destination based on the Output Cost that is set.

Step 5: Run the Simulation for 40 seconds.

3.8.5 Output

The following image is a depiction of the shortest path first tree created in NetSim. This is for representational purposes and cannot be opened in NetSim. The blue color numbers are the "Output Cost" parameter of the link and is set by the user in Router > Application Layer > Interface. The red numbers are the IP addresses of the interfaces of the Routers.

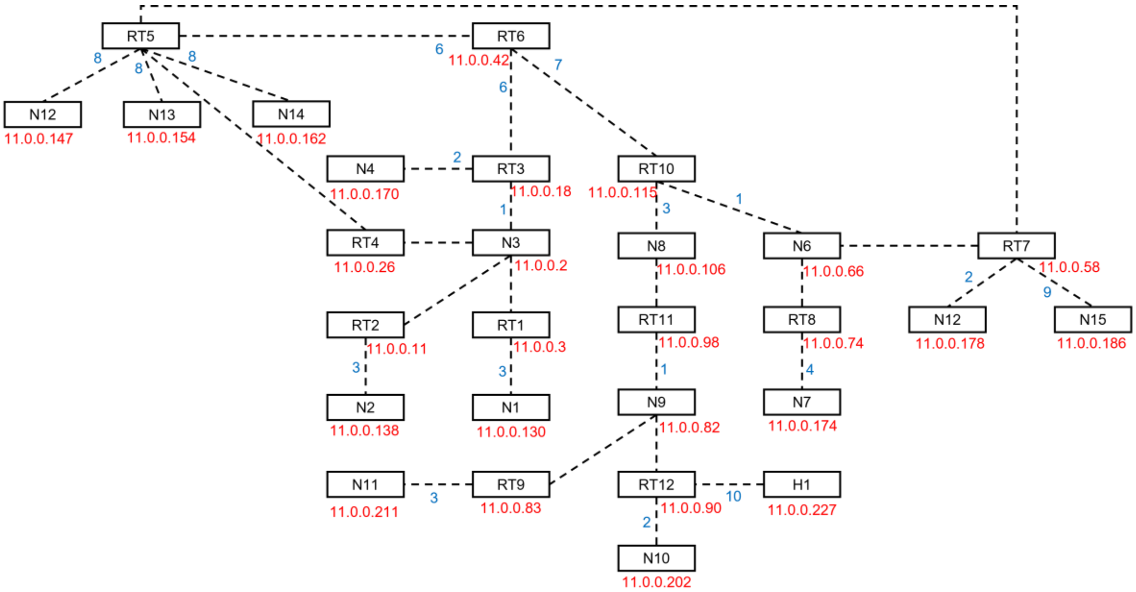


Figure 3-73: SPF tree for Router 6

NOTE: NetSim, does not implement Link type3 (Link to Stub Network). Hence users would notice a slight difference between the SPF trees of RFC and NetSim.

The IP forwarding table formed in the routers can be accessed from the IP Forwarding Table list present in the Simulation Results window as shown below table.

Network Destination	Gateway	Interface	Metrics	Type
11.0.0.18	11.0.0.50	11.0.0.51	6	OSPF
11.0.0.50	11.0.0.50	11.0.0.51	6	OSPF
11.0.0.171	11.0.0.50	11.0.0.51	6	OSPF
11.0.0.34	11.0.0.43	11.0.0.42	6	OSPF
11.0.0.43	11.0.0.43	11.0.0.42	6	OSPF
11.0.0.59	11.0.0.43	11.0.0.42	6	OSPF
11.0.0.146	11.0.0.43	11.0.0.42	6	OSPF
11.0.0.155	11.0.0.43	11.0.0.42	6	OSPF
11.0.0.163	11.0.0.43	11.0.0.42	6	OSPF
11.0.0.3	11.0.0.50	11.0.0.51	7	OSPF
11.0.0.131	11.0.0.50	11.0.0.51	7	OSPF
11.0.0.11	11.0.0.50	11.0.0.51	7	OSPF
11.0.0.139	11.0.0.50	11.0.0.51	7	OSPF
11.0.0.26	11.0.0.50	11.0.0.51	7	OSPF
11.0.0.35	11.0.0.50	11.0.0.51	7	OSPF
11.0.0.115	11.0.0.122	11.0.0.123	7	OSPF
11.0.0.122	11.0.0.122	11.0.0.123	7	OSPF
11.0.0.218	11.0.0.122	11.0.0.123	7	OSPF
11.0.0.2	11.0.0.50	11.0.0.51	7	OSPF
11.0.0.10	11.0.0.50	11.0.0.51	7	OSPF
11.0.0.19	11.0.0.50	11.0.0.51	7	OSPF
11.0.0.27	11.0.0.50	11.0.0.51	7	OSPF
11.0.0.58	11.0.0.122	11.0.0.123	8	OSPF
11.0.0.67	11.0.0.122	11.0.0.123	8	OSPF

11.0.0.179	11.0.0.122	11.0.0.123	8	OSPF
11.0.0.187	11.0.0.122	11.0.0.123	8	OSPF
11.0.0.74	11.0.0.122	11.0.0.123	8	OSPF
11.0.0.195	11.0.0.122	11.0.0.123	8	OSPF
11.0.0.66	11.0.0.122	11.0.0.123	8	OSPF
11.0.0.75	11.0.0.122	11.0.0.123	8	OSPF
11.0.0.114	11.0.0.122	11.0.0.123	8	OSPF
11.0.0.170	11.0.0.50	11.0.0.51	8	OSPF
11.0.0.98	11.0.0.122	11.0.0.123	10	OSPF
11.0.0.107	11.0.0.122	11.0.0.123	10	OSPF
11.0.0.130	11.0.0.50	11.0.0.51	10	OSPF
11.0.0.138	11.0.0.50	11.0.0.51	10	OSPF
11.0.0.178	11.0.0.122	11.0.0.123	10	OSPF
11.0.0.106	11.0.0.122	11.0.0.123	10	OSPF
11.0.0.219	11.0.0.122	11.0.0.123	10	OSPF
11.0.0.83	11.0.0.122	11.0.0.123	11	OSPF
11.0.0.210	11.0.0.122	11.0.0.123	11	OSPF
11.0.0.90	11.0.0.122	11.0.0.123	11	OSPF
11.0.0.203	11.0.0.122	11.0.0.123	11	OSPF
11.0.0.226	11.0.0.122	11.0.0.123	11	OSPF
11.0.0.82	11.0.0.122	11.0.0.123	11	OSPF
11.0.0.91	11.0.0.122	11.0.0.123	11	OSPF
11.0.0.99	11.0.0.122	11.0.0.123	11	OSPF
11.0.0.194	11.0.0.122	11.0.0.123	12	OSPF
11.0.0.202	11.0.0.122	11.0.0.123	13	OSPF
11.0.0.154	11.0.0.43	11.0.0.42	14	OSPF
11.0.0.162	11.0.0.43	11.0.0.42	14	OSPF
11.0.0.211	11.0.0.122	11.0.0.123	14	OSPF
11.0.0.147	11.0.0.43	11.0.0.42	14	OSPF
11.0.0.186	11.0.0.122	11.0.0.123	17	OSPF
11.0.0.227	11.0.0.122	11.0.0.123	21	OSPF

Table 3-27: Exported list of IP forwarding table of RT 6 from NetSim's Result Dashboard

From the above table, the router forwards packets intended to the subnet:

From	Destination	IP Address	Route to Destination	Distance (Metrics)
RT6	N1	11.0.0.130	RT3>N3>RT1>N1 (6+1+0+3)	10
RT6	N2	11.0.0.138	RT3>N3>RT2>N2 (6+1+0+3)	10
RT6	N3	11.0.0.2	RT3>N3 (6+1)	7
RT6	N4	11.0.0.170	RT3>N4 (6+2)	8
RT6	N6	11.0.0.66	RT10>N6 (7+1)	8
RT6	N7	11.0.0.194	RT10>N6>RT8>N7 (7+1+0+4)	12
RT6	N8	11.0.0.106	RT10>N8 (7+3)	10
RT6	N9	11.0.0.82	RT10>N8>RT11>N9 (7+3+0+1)	11
RT6	N10	11.0.0.202	RT10> N8>RT11>N9>RT12>N10 (7+3+0+1+0+2)	13
RT6	N11	11.0.0.211	RT10> N8>RT11>N9>RT9>N11 (7+3+0+1+0+3)	14
RT6	H1	11.0.0.227	RT10> N8>RT11>N9>RT12>H1 (7+3+0+1+0+10)	21
RT6	RT5	11.0.0.34	RT5 (6)	6
RT6	RT7	11.0.0.58	RT10> N6>RT7 (7+1+0)	8
RT6	N12	11.0.0.147	RT10>N6>RT7>N12 (7+1+0+2)	10
RT6	N13	11.0.0.154	RT5>N13 (6+8)	14
RT6	N14	11.0.0.162	RT5>N4 (6+8)	14
RT6	N15	11.0.0.186	RT10>N6>RT7>N15 (7+1+0+9)	17

Table 3-28: Distance calculated from RT6 to destinations.

Open Packet Trace and filter PACKET ID to 1 and CONTROL PACKET TYPE/APP NAME to WN13-WN1 to view the detailed information of the routes taken to reach the destination WN1.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
1	0	CBR	WN13-WN1	NODE-41	NODE-27	NODE-41	ROUTER-6
1	0	CBR	WN13-WN1	NODE-41	NODE-27	ROUTER-6	ROUTER-3
1	0	CBR	WN13-WN1	NODE-41	NODE-27	ROUTER-3	ROUTER-13
1	0	CBR	WN13-WN1	NODE-41	NODE-27	ROUTER-13	ROUTER-1
1	0	CBR	WN13-WN1	NODE-41	NODE-27	ROUTER-1	ROUTER-17
1	0	CBR	WN13-WN1	NODE-41	NODE-27	ROUTER-17	NODE-27

Figure 3-74: Packet Trace showing the route in which the packets have chosen to reach the destination.

Similarly, filter CONTROL PACKET TYPE/APP NAME to the remaining destination nodes to view the routes taken.

Thus, we are able to simulate the exact example as provided in the RFC and report that SPF Tree obtained, and the routing costs match the analysis provided in the RFC.

4 Transmission control protocol (TCP)

4.1 Introduction to TCP connection management (Level 1)

4.1.1 Introduction

When an application process in a client host seeks a reliable data connection with a process in another host (say, server), the client-side TCP then proceeds to establish a TCP connection with the TCP at the server side. A TCP connection is a point-to-point, full-duplex logical connection with resources allocated only in the end hosts. The TCP connection between the client and the server is established in the following manner and is illustrated in Figure 4-1.

1. The TCP at the client side first sends a special TCP segment, called the SYN packet, to the TCP at the server side.
2. Upon receiving the SYN packet, the server allocates TCP buffer and variables to the connection. Also, the server sends a connection-granted segment, called the SYN-ACK packet, to the TCP at the client side.
3. Upon receiving the SYN-ACK segment, the client also allocates buffers and variables to the connection. The client then acknowledges the server's connection granted segment with an ACK of its own.

This connection establishment procedure is often referred to as the three-way handshake. The special TCP segments can be identified by the values in the fields SYN, ACK and FIN in the TCP header (see Figure 4-2). We also note that the TCP connection is uniquely identified by the source and destination port numbers (see Figure 4-2) exchanged during TCP connection establishment and the source and destination IP addresses.

Once a TCP connection is established, the application processes can send data to each other. The TCP connection can be terminated by either of the two processes. Suppose that the client application process seeks to terminate the connection. Then, the following handshake ensures that the TCP connection is torn down.

1. The TCP at the client side sends a special TCP segment, called the FIN packet, to the TCP at the server side.
2. When the server receives the FIN segment, it sends the client an acknowledgement segment in return and its own FIN segment to terminate the full-duplex connection.
3. Finally, the client acknowledges the FIN-ACK segment (from the server) with an ACK of its own. At this point, all the resources (i.e., buffers and variables) in the two hosts are deallocated.

During the life of a TCP connection, the TCP protocol running in each host makes transitions through various TCP states. Figure 4-1 illustrates the typical TCP states visited by the client and the server during connection establishment and data communication.

TCP is defined in RFCs 793, 1122, 7323 and, 2018. A recommended textbook reference for TCP is Chapter 3: Transport layer, of Computer Networking: A top-down approach, by James Kurose and Keith Ross (Pearson).

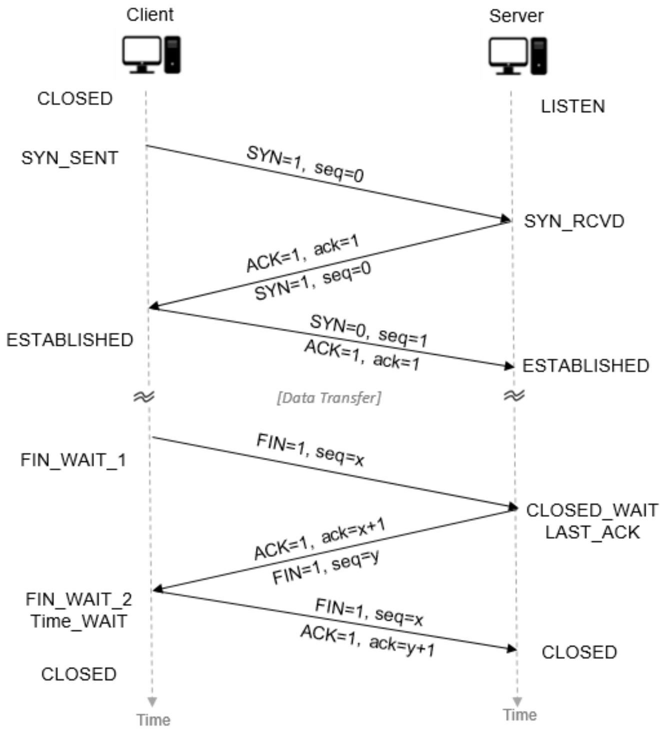


Figure 4-1: TCP connection establishment between a client and a server

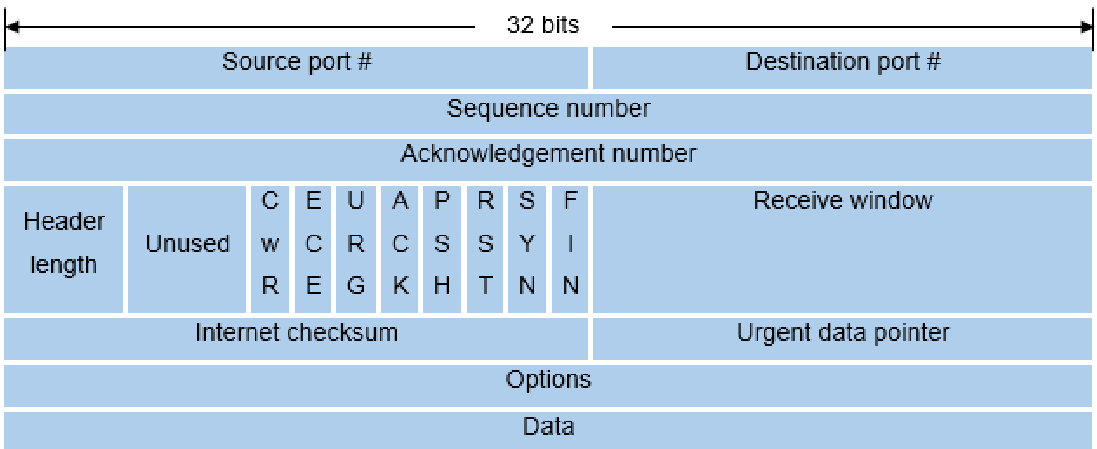


Figure 4-2: TCP Header

4.1.2 Network Setup

Open NetSim and click on **Experiments >Internetworks> TCP> Introduction to TCP connection management** then click on the tile in the middle panel to load the example as shown in below Figure 4-3.

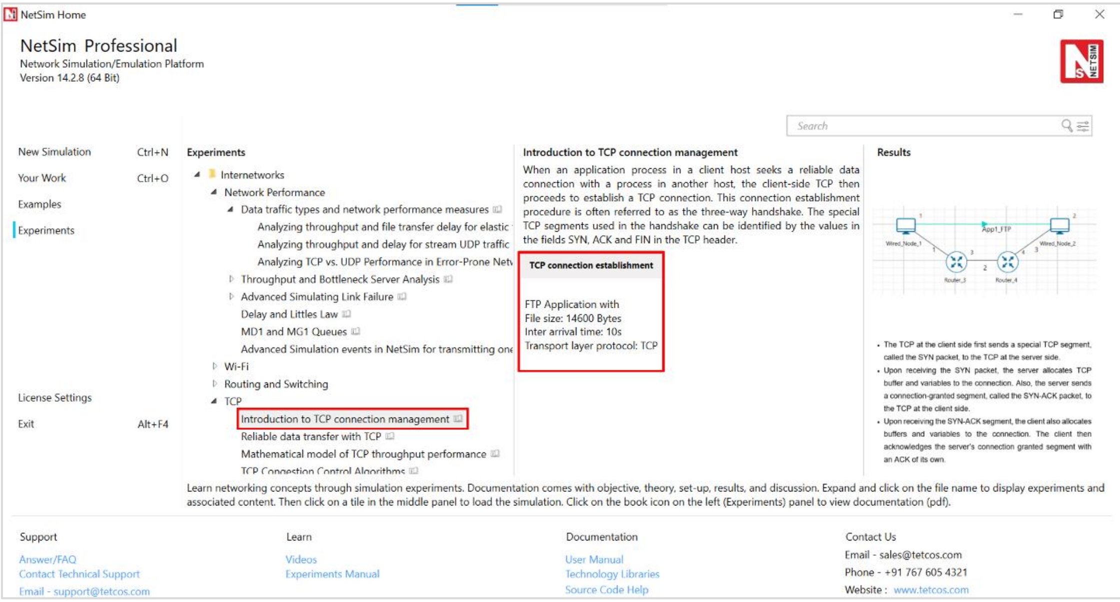


Figure 4-3: List of scenarios for the example of Introduction to TCP connection management

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 4-4

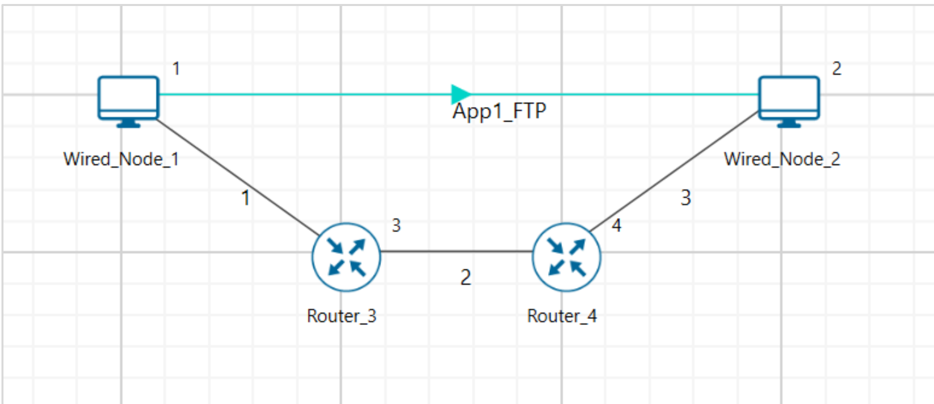


Figure 4-4: Network set up for studying the Introduction to TCP connection management

4.1.3 Procedure

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 Control algorithm set to NEW RENO.

To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned in the steps.

***NOTE:** Routers are configured with default properties.*

Step 3: Click on the link ID (of a wired link) and expand the link property panel on right. 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: Configure FTP application between two nodes by selecting an application from the Set Traffic tab in the ribbon at the top. Click on the application flow App1 FTP, expand the application properties panel on the right, and set the File Size set to 14600 Bytes and File Inter Arrival Time set to 10 Seconds

Step 5: Click on Show/Hide info > Device IP check box in the NetSim GUI to view the network topology along with the IP address.

Step 6: Enable the Throughput vs Time plots under Link and Application performance and Run simulation for 10 seconds.

4.1.4 Output

We have enabled Wireshark capture in Wired Node 1. The PCAP file is generated at the end of the simulation and is shown in Figure 4-5.

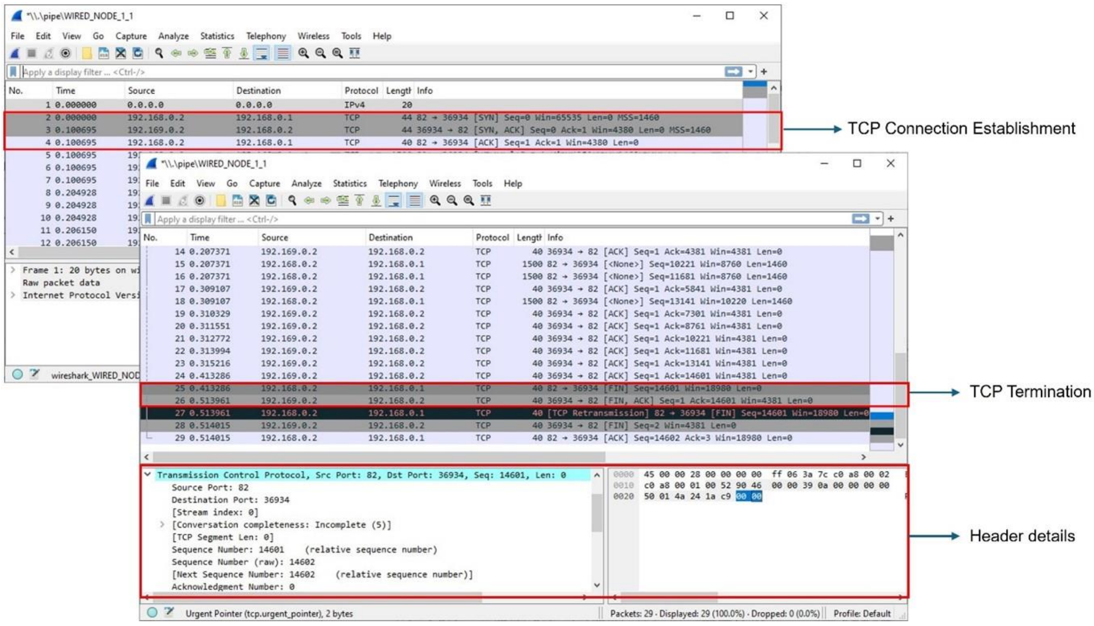


Figure 4-5: Wireshark Packet capture at Wired Node 1

1. The 3-way handshake of TCP connection establishment and TCP connection termination is observed in the packet capture (Figure 4-5).
2. Data is transferred only after the TCP connection is established.
3. We can access the packet header details of the TCP segments (SYN, SYN-ACK, FIN, FINACK) in Wireshark.

Note that:

- When devices in different Wide Area Networks (WANs) need to communicate, their private IP addresses are converted to public IP addresses using Network Address Translation (NAT).
- In the above TCP sample, WireShark captures the TCP three-way handshake as shown in Figure 4-5. Here the source device in one network (192.168.0.2) sends data to another network via Router 3.
- Router 3 converts its private IP (192.168.0.1) to a public one (11.0.0.3) using NAT, allowing it to communicate with devices outside its network.
- The data is then forwarded to Router 4 in another network, where NAT converts the public IP (11.0.0.2) back to a private one (192.169.0.1) to reach the destination device (192.169.0.2).
- This process enables communication between devices in different networks using public IP addresses, while devices within the same network can communicate using their private IP addresses.

4.1.5 Exercises

1. Construct a scenario with 2 nodes and 2 routers. Introduce the link failure and recovery at Router3. Set Uptime as 0, 10, and the downtime as 5. This means the link is up from 0s – 5s, then down from 5s - 10s and then up again after 10s. Explain what you observe in between 5th second and 10th second using the packet trace. Does TCP reestablish connection after 10s?

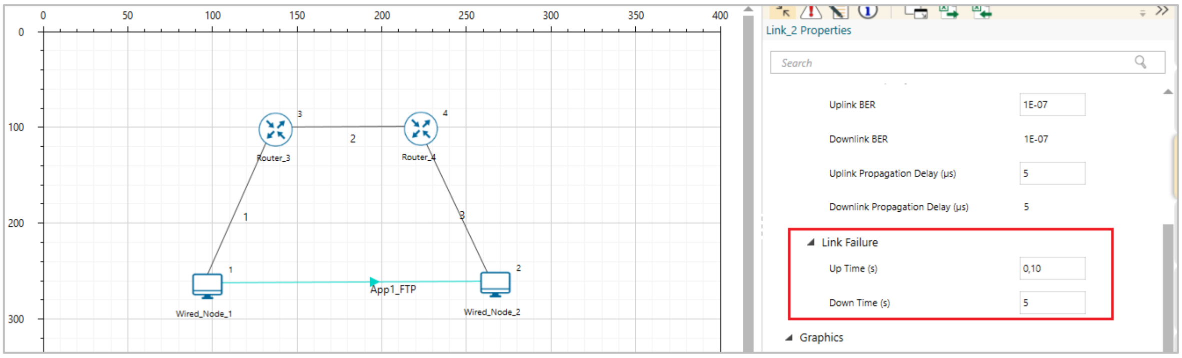


Figure 4-6: Network scenario for TCP.

2. For more variations change the number of routers, link speeds, and the time at which the link fails and recovers.

4.2 Reliable data transfer with TCP (Level 1)

4.2.1 Introduction

TCP provides reliable data transfer service to the application processes even when the underlying network service (IP service) is unreliable (loses, corrupts, garbles or duplicates packets). TCP uses checksum, sequence numbers, acknowledgements, timers and retransmission to ensure correct and in order delivery of data to the application processes.

TCP views the data stream from the client application process as an ordered stream of bytes. TCP will grab chunks of this data (stored temporarily in the TCP send buffer), add its own header and pass it on to the network layer. A key field of the TCP header is the sequence number which indicates the position of the first byte of the TCP data segment in the data stream. The sequence number will allow the TCP receiver to identify segment losses, duplicate packets and to ensure correct delivery of the data stream to the server application process.

When a server receives a TCP segment, it acknowledges the same with an ACK segment (the segment carrying the acknowledgement has the ACK bit set to 1) and also conveys the sequence number of the first missing byte in the application data stream, in the acknowledgement number field of the TCP header. All acknowledgements are cumulative; hence, all missing, and out-of-order TCP segments will result in duplicate acknowledgements for the corresponding TCP segments.

TCP sender relies on sequence numbering and acknowledgements to ensure reliable transfer of the data stream. In the event of a timeout (no acknowledgement is received before the timer expires) or triple duplicate acknowledgements (multiple ACK segments indicate a lost or missing TCP segment) for a TCP segment, the TCP sender will retransmit the segment until the TCP segment is acknowledged (at least cumulatively). In Figure 4-7, we illustrate retransmission by the TCP sender after a timeout for acknowledgement.

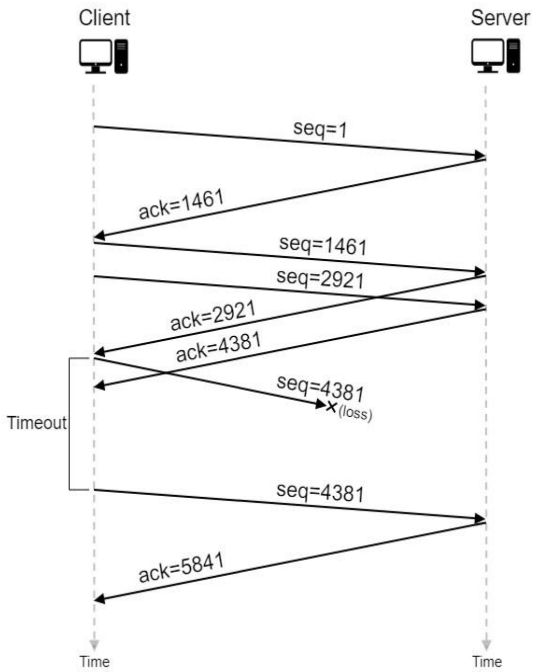


Figure 4-7: An illustration of TCP retransmission with timeout. The segment with sequence number 4381 is lost in the network. The TCP client retransmits the segment after a timeout event.

4.2.2 Network Setup

Open NetSim and click on **Experiments > Internetworks> TCP> Reliable data transfer with TCP** then click on the tile in the middle panel to load the example as shown in below Figure 4-8.

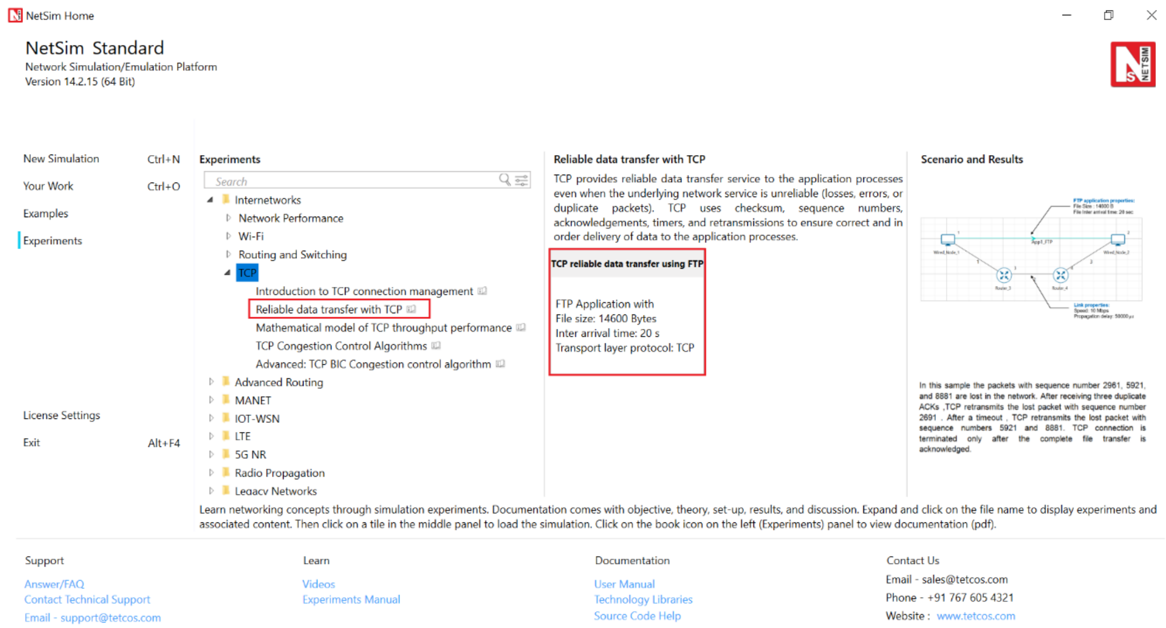


Figure 4-8: List of scenarios for the example of Reliable data transfer with TCP

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 4-9.

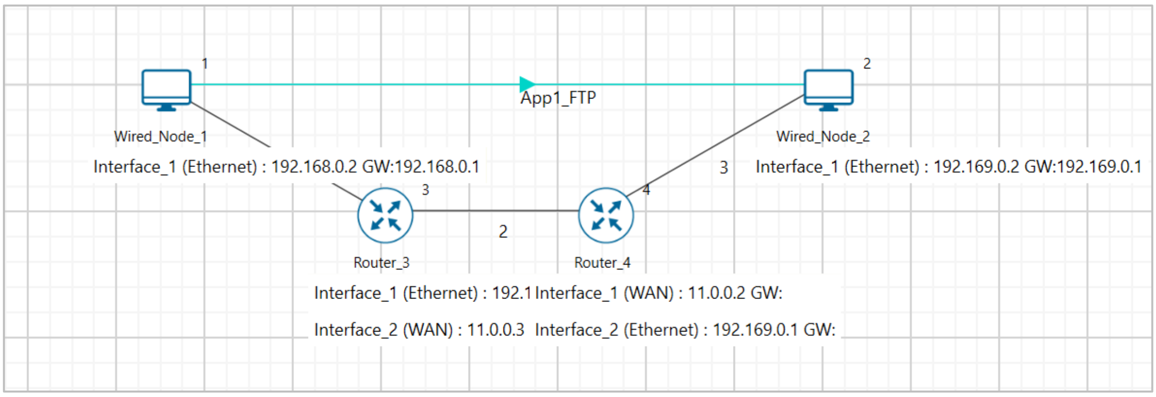


Figure 4-9: Network set up for studying the Reliable data transfer with TCP

We will seek a simple file transfer with TCP over a lossy link to study reliable data transfer with TCP. We will simulate the network setup illustrated in Figure 4-9 with the configuration parameters listed in detail to study reliable data transfer with TCP connection.

4.2.3 Procedure

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 and Wired Node 2 i.e., Destination, Wireshark Capture is set to Online.

To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned in the steps.

Note: Routers are configured with default properties.

Step 3: Click on the link ID (of a wired link) and expand the link properties panel on right. 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 **0.00001** for the backbone link connecting the routers, i.e., 2.

Step 4: Configure FTP application between two nodes by selecting an application from the Set Traffic tab in the ribbon at the top. Click on the application flow App1 FTP, expand the application properties panel on the right, and set the File Size to 14600 bytes and the File Inter Arrival Time to 20 seconds.

Step 5: Click on Show/Hide info > Enable Device IP check box in the NetSim GUI to view the network topology along with the IP address.

Step 6: Enable Throughput vs Time plots under Application and link performance and click on Run simulation. The simulation time is set to 20 seconds.

4.2.4 Output

We aimed to transfer a file of size 14600 bytes (i.e., 10 packets, each of size 1460 bytes) with TCP over a lossy link. In Figure 4-10, we report the application metrics data for FTP which indicates that the complete file was transferred.

Application Metrics

End-to-end performance of applications running across the network.

Application ID	Application Name	Source ID	Destination ID	Throughput (Mbps	Delay (µs)	Jitter (µs)
1	App1_FTP	1	2	0.005840	2503061.760000	380974.542222

Figure 4-10: Application Metrics table for FTP

We have enabled Wireshark Capture in Wired_Node 1 and Wired Node_2. The PCAP files are generated at the end of the simulation and are shown in Figure 4-11 and Figure 4-12.

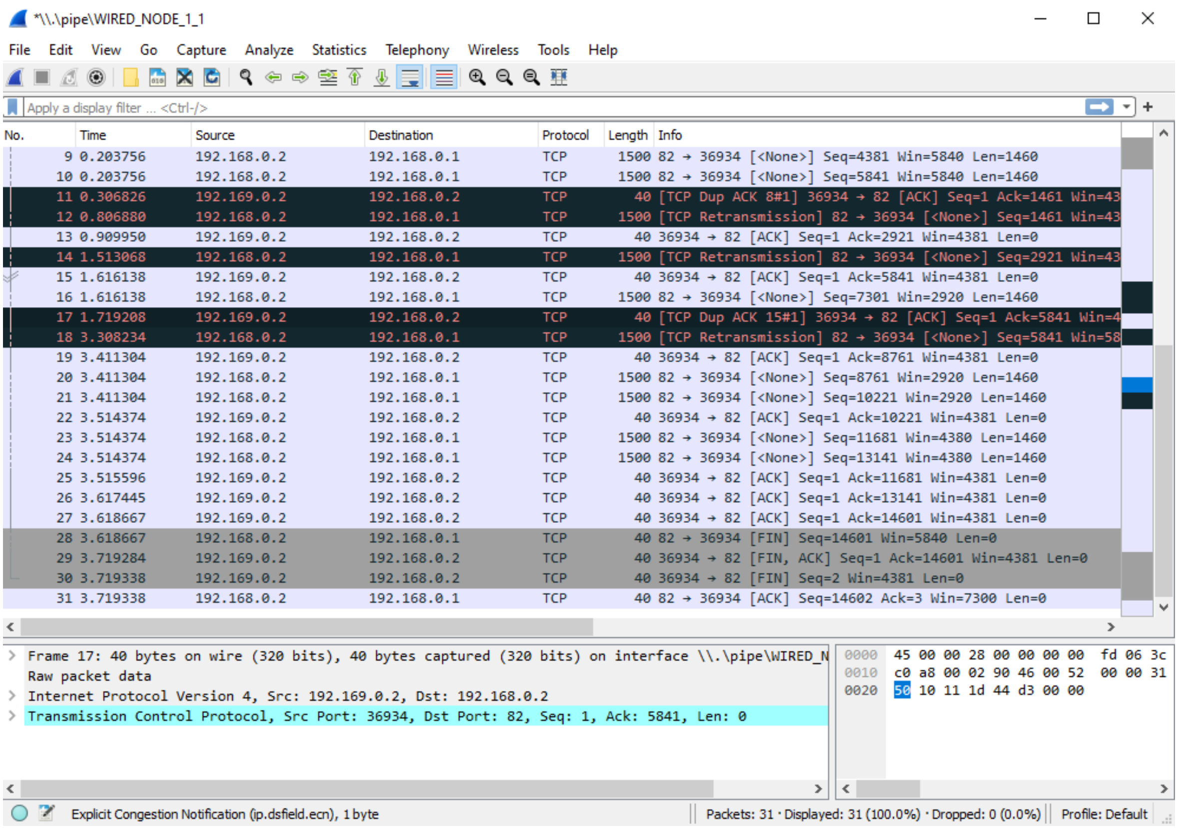


Figure 4-11: PCAP file at Wired Node 1. TCP ensures reliable data transfer using timeout, duplicate ACKs and retransmissions.

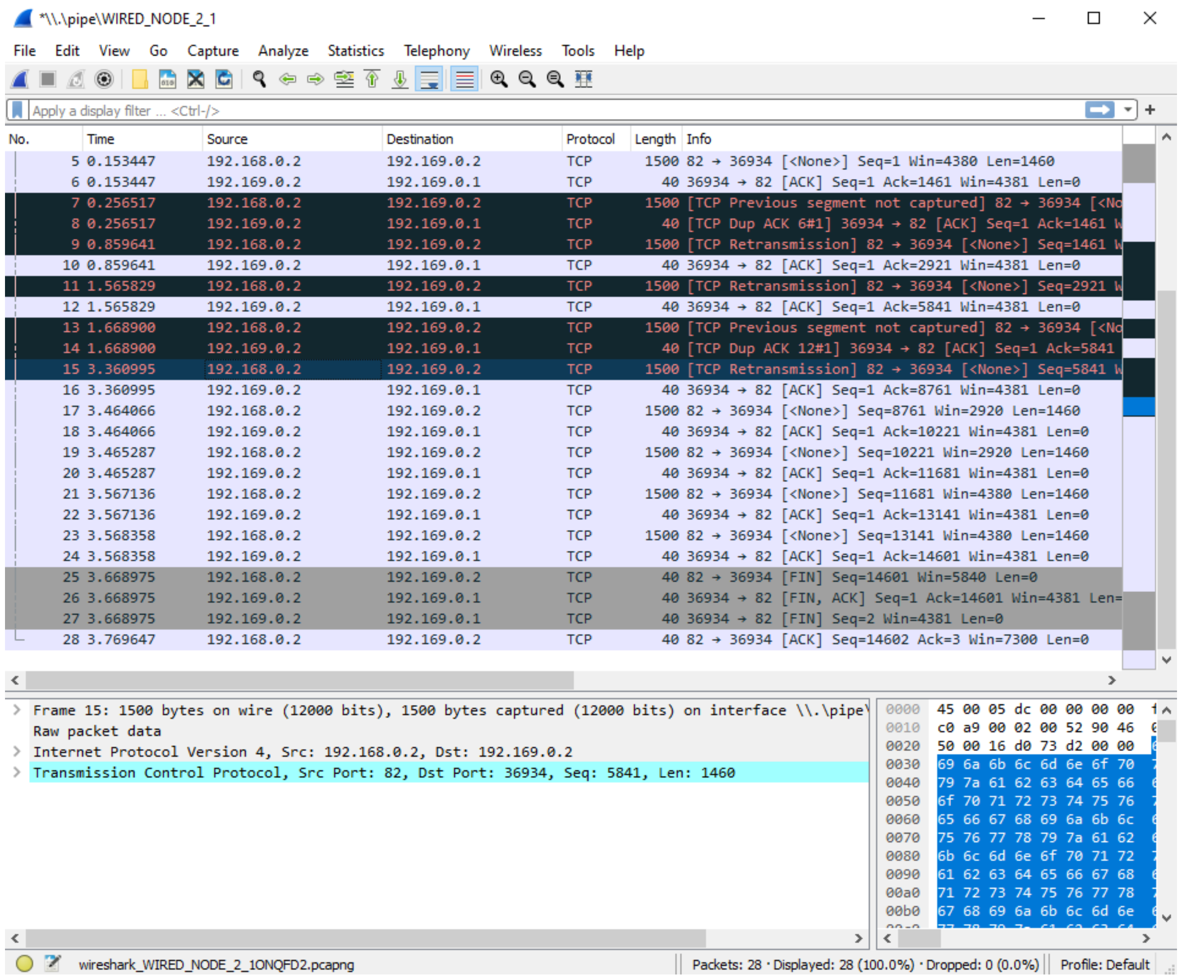


Figure 4-12: PCAP file at Wired Node 2

4.2.5 Inference

1. From Figure 4-11 and Figure 4-12, we note that the packets with sequence number 1461 and 5841 are errored in the network, which can also observe in Packet Trace.
2. After receiving three duplicate ACKs (in lines 13, 14 of Figure 4-11), TCP retransmits the errored packet. (In line 15 of Figure 4-11).
3. TCP connection is terminated only after the complete file transfer is acknowledged which can be observed in Figure 4-5 (Line 25 and 26).

4.3 Mathematical Modelling of TCP Throughput Performance (Level 2)

4.3.1 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.

4.3.1.1 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

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

4.3.1.2 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 4-20** 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,

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

4.3.2 Network Setup

Open NetSim and click on **Experiments> Internetworks> TCP> Mathematical model of TCP throughput performance** then click on the tile in the middle panel to load the example as shown below in Figure 4-13.

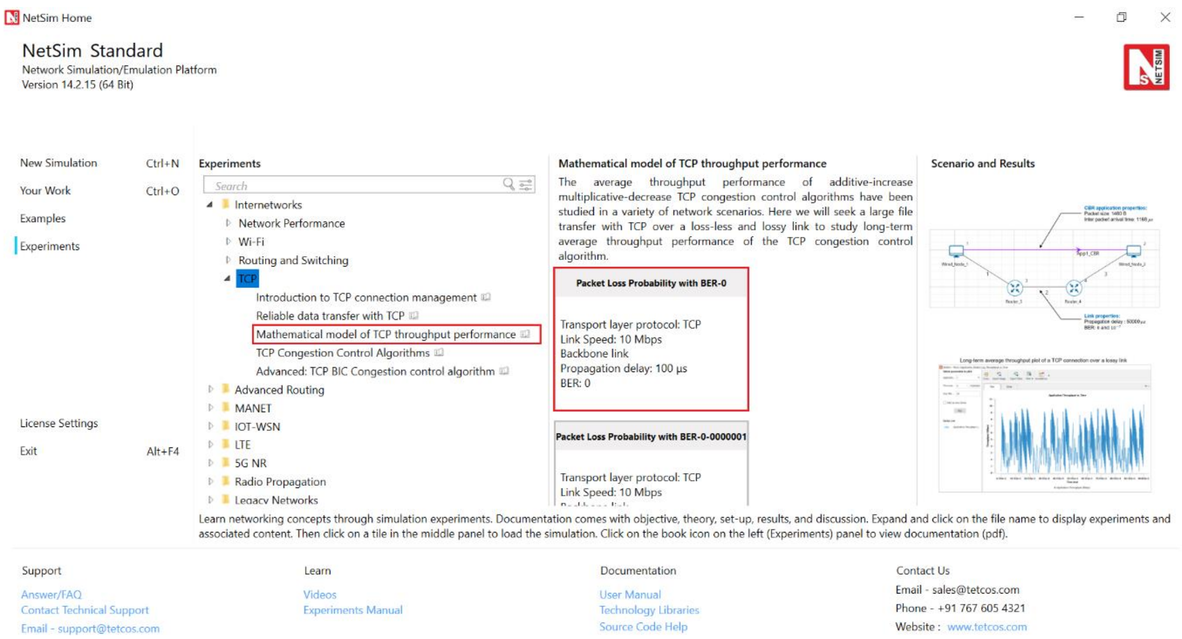


Figure 4-13: List of scenarios for the example of Mathematical model of TCP throughput performance

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 4-14.

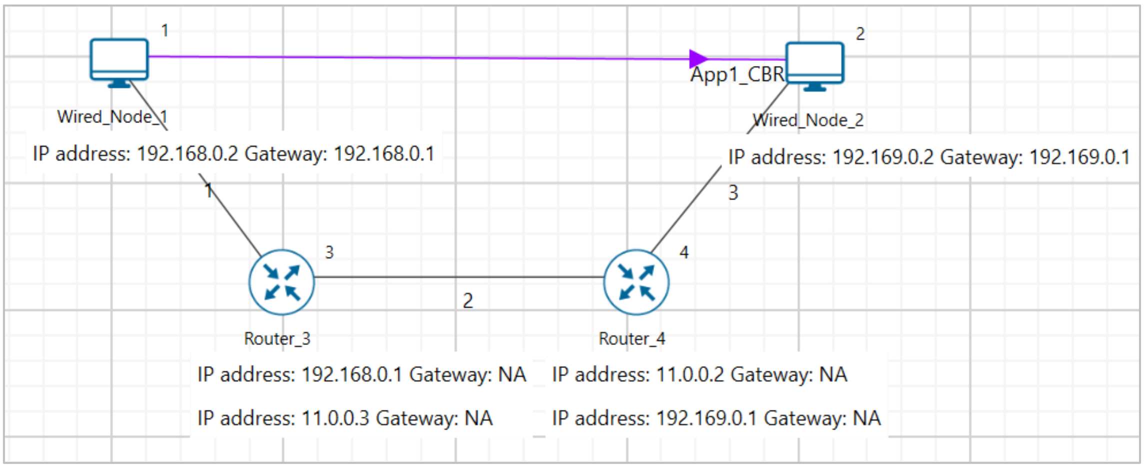


Figure 4-14: Network set up for studying the Mathematical model of TCP throughput performance

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 4-14 with the two (loss-less and lossy) configuration parameters listed in detail to study the throughput performance of TCP New Reno.

4.3.3 Procedure

Packet Loss Probability with BER-0 Sample

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 Control algorithm is set to NEW RENO.

To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned in the steps.

NOTE: Routers are configured with default properties.

Step 3: Click the link ID (of a wired link) and expand the link properties panel on the right. 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: Configure an application between two nodes by selecting an application from the Set Traffic tab in the ribbon at the top. Click on the application flow App1 CBR, expand the application properties panel on the right, and set the Packet Size to 1460 bytes and the Inter Arrival Time to 1168 microseconds.

Step 5: Click on Show/Hide info > Device IP Enable check box in the NetSim GUI to view the network topology along with the IP address.

Step 6: Enable **Application throughput vs time plot** from NetSim UI. 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.

Packet Loss Probability with BER-0.0000001 Sample

Step 1: Click the link ID (of a wired link) and expand the link properties panel on the right. 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.

4.3.4 Output

In Figure 4-15, we report the application metrics data for data transfer over a loss-less link (**Packet Loss Probability with BER-0 sample**).

Application Metrics

End-to-end performance of applications running across the network.

Application ID	Application Name	Source ID	Destination ID	Throughput (Mbps	Delay (µs)	Jitter (µs)
1	App1_CBR	1	2	4.912024	25667056.436303	1206.187458

Figure 4-15: Application Metrics with BER = 0

In Figure 4-16, we report the plot of long-term average throughput of the TCP connection over the loss-less link.

Set average window size to 100 ms and plot the graph.

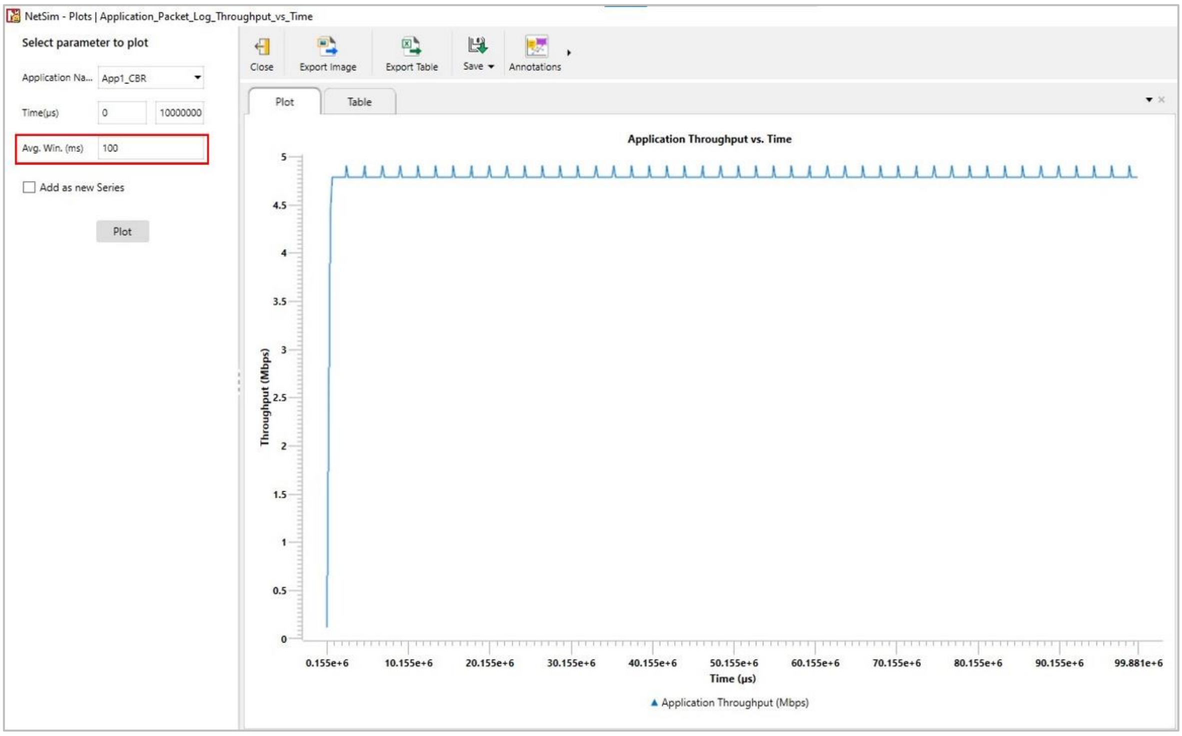


Figure 4-16: Long-term average throughput of TCP New Reno over a loss-less link

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** . In Figure 4-17, we report the congestion window evolution of TCP New Reno over the loss-less link.

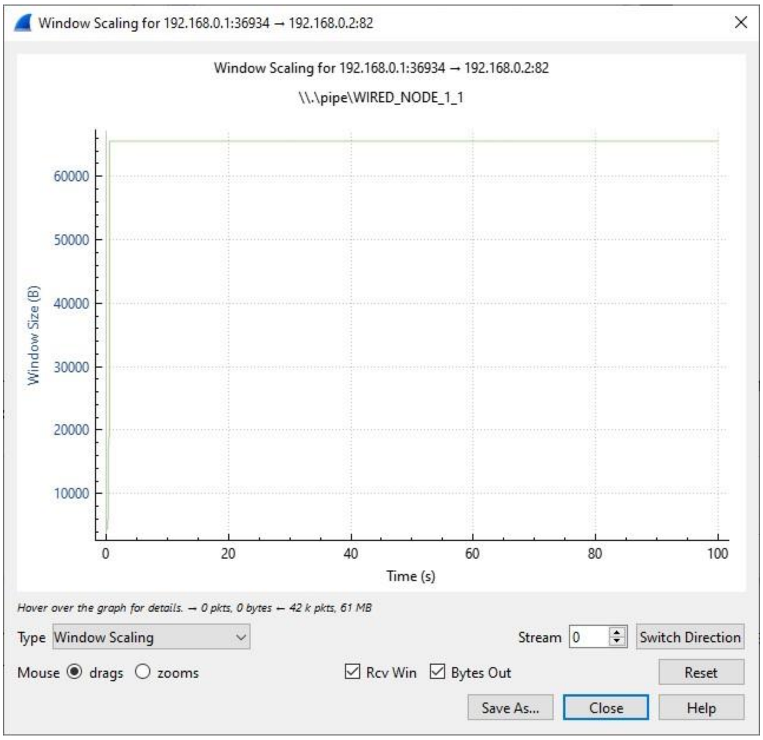


Figure 4-17: Congestion window evolution with TCP New Reno over a loss-less link.

In Figure 4-18, we report the application metrics data for data transfer over a lossy link (**Packet Loss Probability with BER-0.0000001 sample**).

Application Metrics

End-to-end performance of applications running across the network.

Application ID	Application Name	Source ID	Destination ID	Throughput (Mbps	Delay (µs)	Jitter (µs)
1	App1_CBR	1	2	3.806395	31885008.631269	1976.201851

Figure 4-18: Application Metrics when $BER = 1 \times 10^{-7}$

In Figure 4-19, we report the plot of long-term average throughput of the TCP connection over the lossy link.

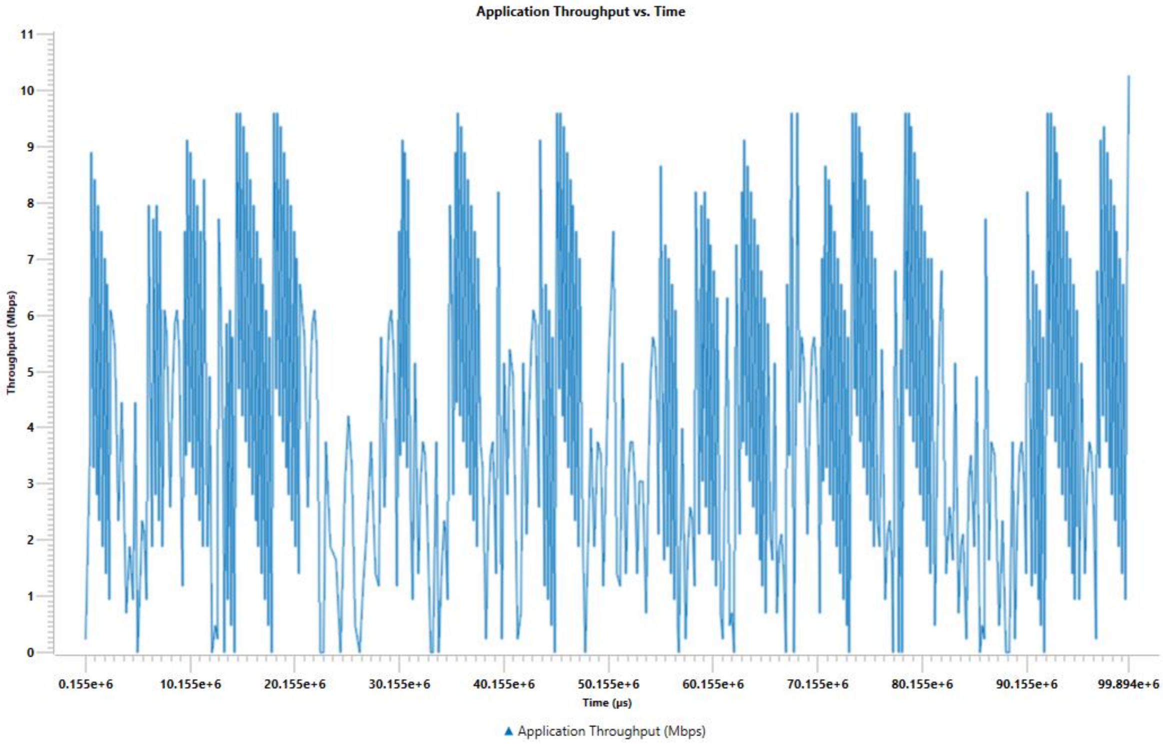


Figure 4-19: Throughput graph

In Figure 4-20, we report the congestion window evolution of TCP New Reno over the lossy link.

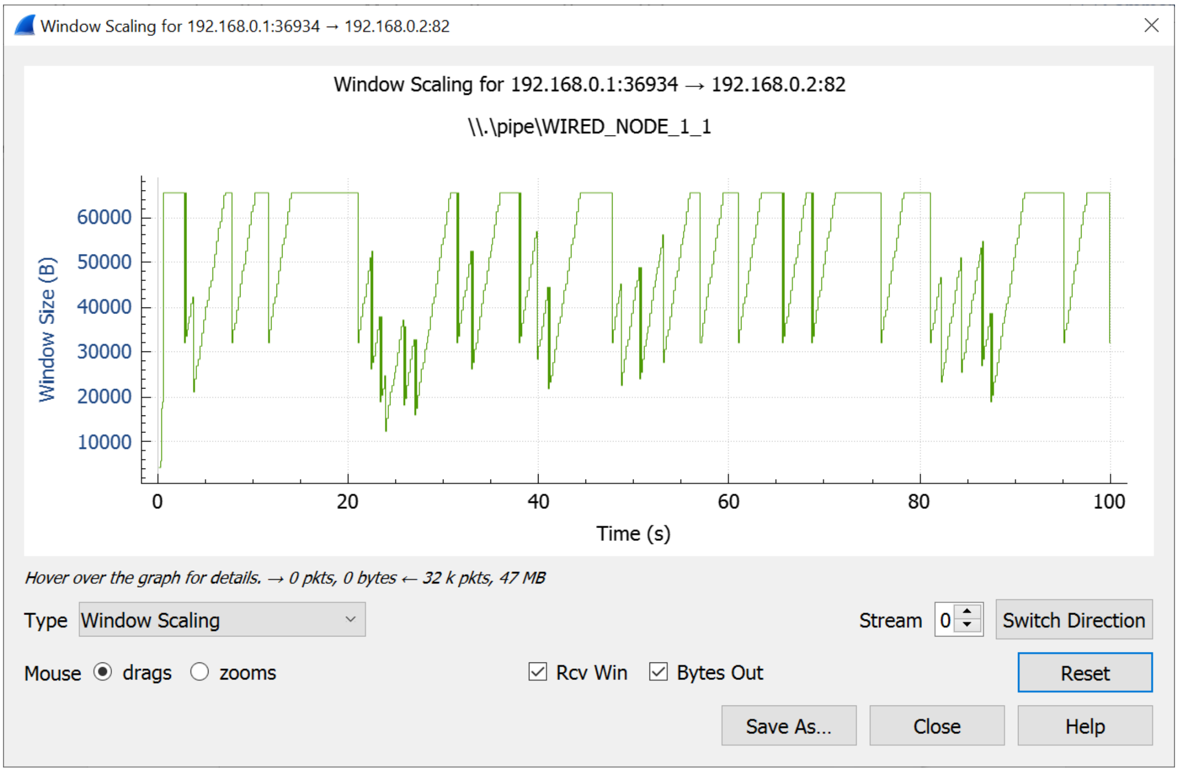


Figure 4-20: Congestion window evolution with TCP New Reno over a lossy link

4.3.5 Observations and Inference

1. In Figure 4-17, 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.

$$Throughput \approx \frac{Window\ Size\ (in\ bits)}{RTT\ (in\ secs)}$$

$$= \frac{65535 \times 8}{100 \times 10^{-3}} = 5.24\ Mbps$$

   We note that the observed long-term average throughput (see **Figure 4-15**) is approximately equal to the above computed value.

2. In Figure 4-20, 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

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

$$\approx \sqrt{\frac{3}{2 \times 1.2 \times 10^{-3}}} \times \frac{1460 \times 8}{100 \times 10^{-3}}$$

$$= 4.12\ Mbps$$

   where the packet error rate p can be computed from the bit error rate ($BER = 1e^{-7}$) and the 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 4-18**) is approximately equal to the above computed value.

4.4 TCP Congestion Control Algorithms (Level 2)

4.4.1 Introduction

A key component of TCP is end-to-end congestion control algorithm. The TCP congestion control algorithm limits the rate at which the sender sends traffic into the network based on the perceived network congestion. The TCP congestion control algorithm at the sender maintains a variable called congestion window, commonly referred as *cwnd*, that limits the amount of unacknowledged data in the network. The congestion window is adapted based on the network conditions, and this affects the sender's transmission rate. The TCP sender reacts to congestion and other network conditions based on new acknowledgements, duplicate acknowledgements and timeouts. The TCP congestion control algorithms describe the precise manner in which TCP adapts *cwnd* with the different events.

The TCP congestion control algorithm has three major phases (a) slow-start, (b) congestion avoidance, and (c) fast recovery. In slow-start, TCP is aggressive and increases *cwnd* by one MSS with every new acknowledgement. In congestion avoidance, TCP is cautious and increases the *cwnd* by one MSS per round-trip time. Slow-start and congestion avoidance are mandatory components of all TCP congestion control algorithms. In the event of a packet loss (inferred by timeout or triple duplicate acknowledgements), the TCP congestion control algorithm reduces the congestion window to 1 (e.g., Old Tahoe, Tahoe) or by half (e.g., New Reno). In fast recovery, TCP seeks to recover from intermittent packet losses while maintaining a high congestion window. The new versions of TCP, including TCP New Reno, incorporate fast recovery as well. Figure 4-21 presents a simplified view of the TCP New Reno congestion control algorithm highlighting slow-start, congestion avoidance and fast recovery phases.

TCP congestion control algorithm is often referred to as additive-increase multiplicative-decrease (AIMD) form of congestion control. The AIMD congestion control algorithm often leads to a “saw tooth” evolution of the congestion window (with linear increase of the congestion window during bandwidth probing and a multiplicative decrease in the event of packet losses), see Figure 4-26.

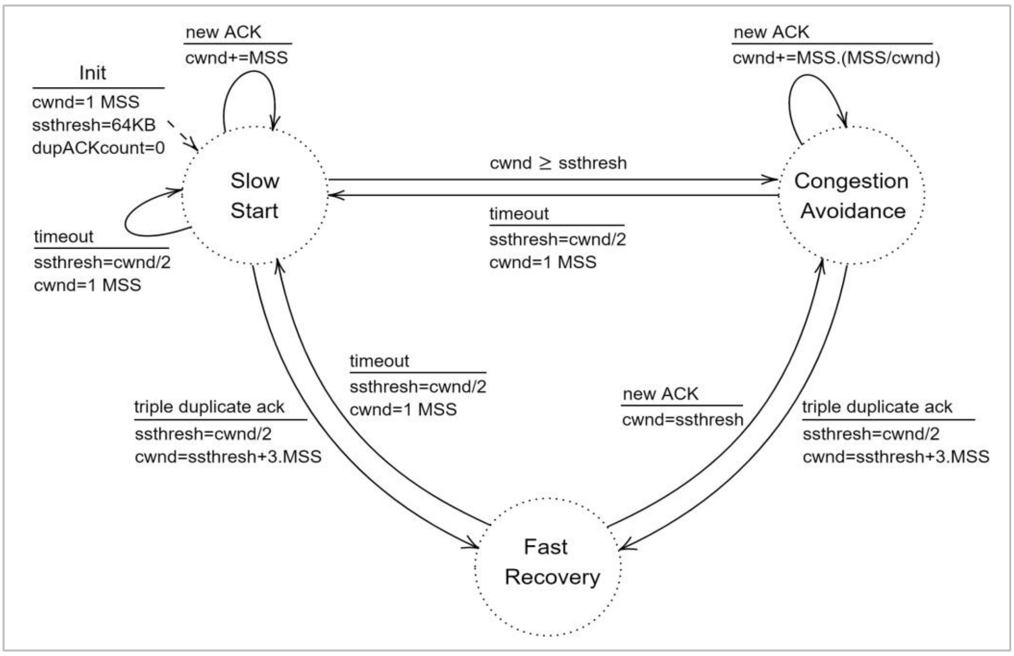


Figure 4-21: A simplified view of FSM of the TCP New Reno congestion control algorithm

4.4.2 Network Setup

We will seek a large file transfer with TCP over a lossy link to study the TCP congestion control algorithms. We will simulate the network setup illustrated in Figure 4-23 with the configuration parameters listed in detail in steps to study the working of TCP congestion control algorithms.

Open NetSim and click on **Experiments> Internetworks>TCP> TCP Congestion Control Algorithms > Old-Tahoe** then click on the tile in the middle panel to load the example as shown in below Figure 4-22.

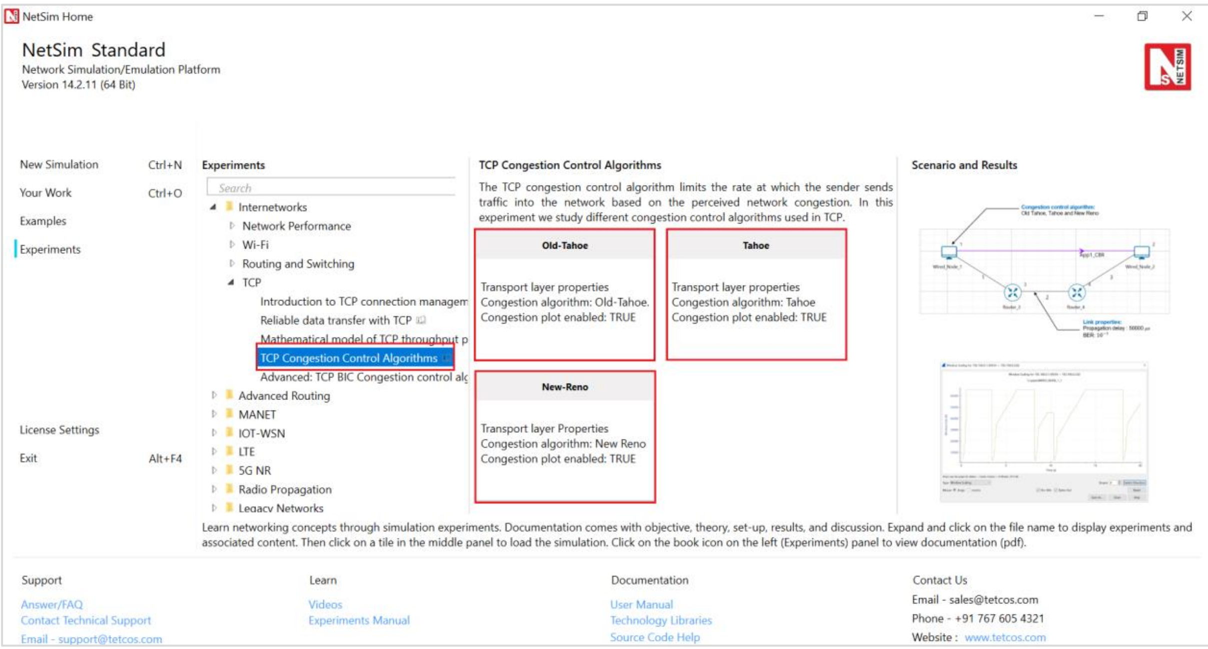


Figure 4-22: List of scenarios for the example of TCP Congestion Control Algorithms

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

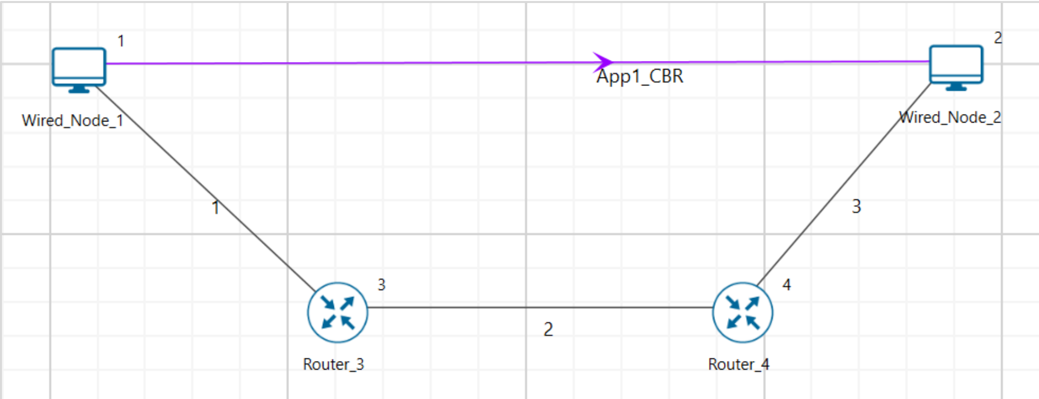


Figure 4-23: Network set up for studying the TCP Congestion Control Algorithms

4.4.3 Procedure

Old Tahoe

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 Wired Node 1(Source node), the Congestion Control Algorithm in the Transport layer properties is set to **OLD TAHOE** .

To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as described.

Step 3: In the General Properties of In the Wired Node 1(Source node), Wireshark Capture is set to Online.

NOTE: Routers properties are set to default.

Step 4: The link properties are configured as shown in the table below. To set the wired link properties, click on the link, expand the link property panel on the right, and configure the settings as mentioned in the table.

Wired link Properties	
Link 2 Properties (Backbone link)	
Parameter	Parameter value
Uplink / Downlink Speed (Mbps)	10
Uplink / Downlink BER	0.0000001
Uplink / Downlink Propagation Delay (µs)	50000
Link 1 and 3 Properties	
Uplink / Downlink Speed (Mbps)	10
Uplink / Downlink BER	0
Uplink / Downlink Propagation Delay (µs)	100

Table 4-1: Wired link properties

Step 5: Configure CBR application between Wired node 1 and Wired node 2 by clicking on Set traffic tab from the ribbon on top. To configure application properties, click on created application and set the Packet Size to 1460 Bytes and Inter Arrival Time to 1168 microseconds by keeping the transport layer to TCP.

Step 6: Click on Show/Hide info > Device IP check box in the NetSim GUI to view the network topology along with the IP address.

Step 7: Click on Run simulation. The **simulation time** is set to 20 seconds.

Tahoe

Step 1: In Wired Node 1 (the source node), the congestion control algorithm is set to TAHOE under the transport layer properties.

Step 2: Run simulation for 20 seconds.

New Reno

Step 1: In Wired Node 1 (the source node), the congestion control algorithm is set to NEW RENO under the transport layer properties.

Step 2: Run simulation for 20 seconds.

4.4.4 Output

We have enabled WireShark Capture in Wired Node 1. The PCAP file is generated during 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.**

The congestion window evolution for Old Tahoe, Tahoe and New Reno congestion control algorithms are presented in Figure 4-24, Figure 4-25, and Figure 4-26, respectively.

Table 4-2 shows the throughput values of different congestion control algorithms (obtained from the Application Metrics).

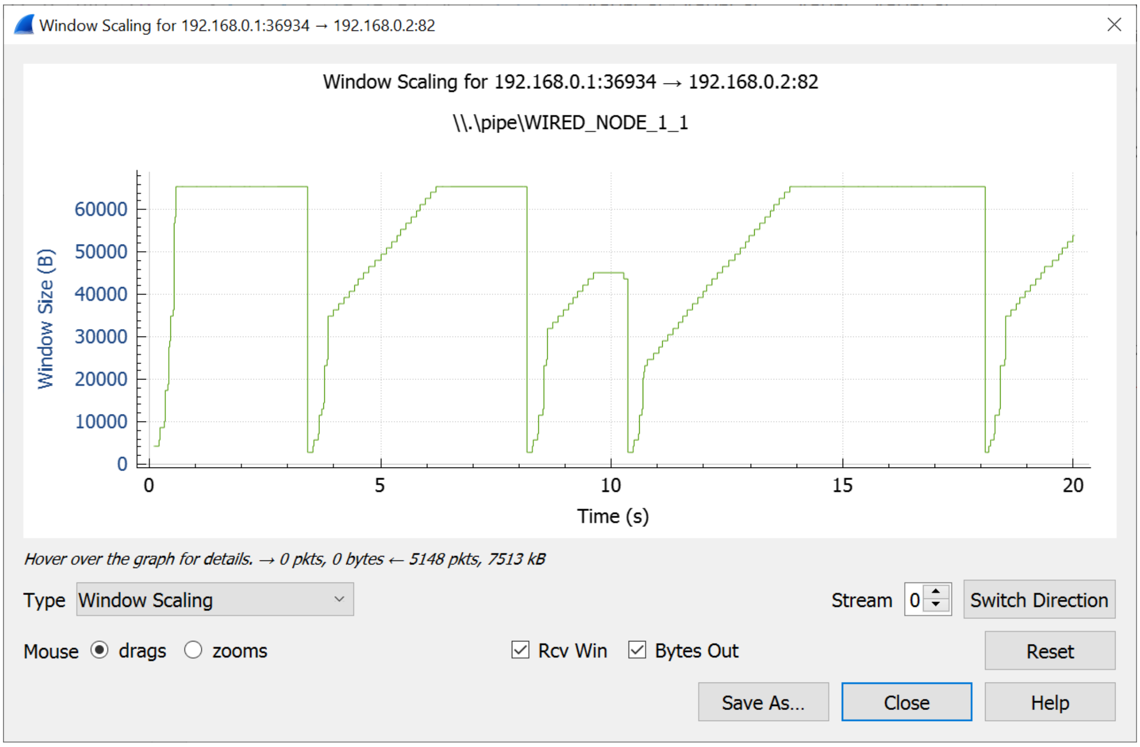


Figure 4-24: Congestion window evolution with TCP Old Tahoe. We note that Old Tahoe infers packet loss only with timeouts, and updates the slow-start threshold ssthresh and congestion window cwnd as ssthresh = cwnd/2 and cwnd = 1

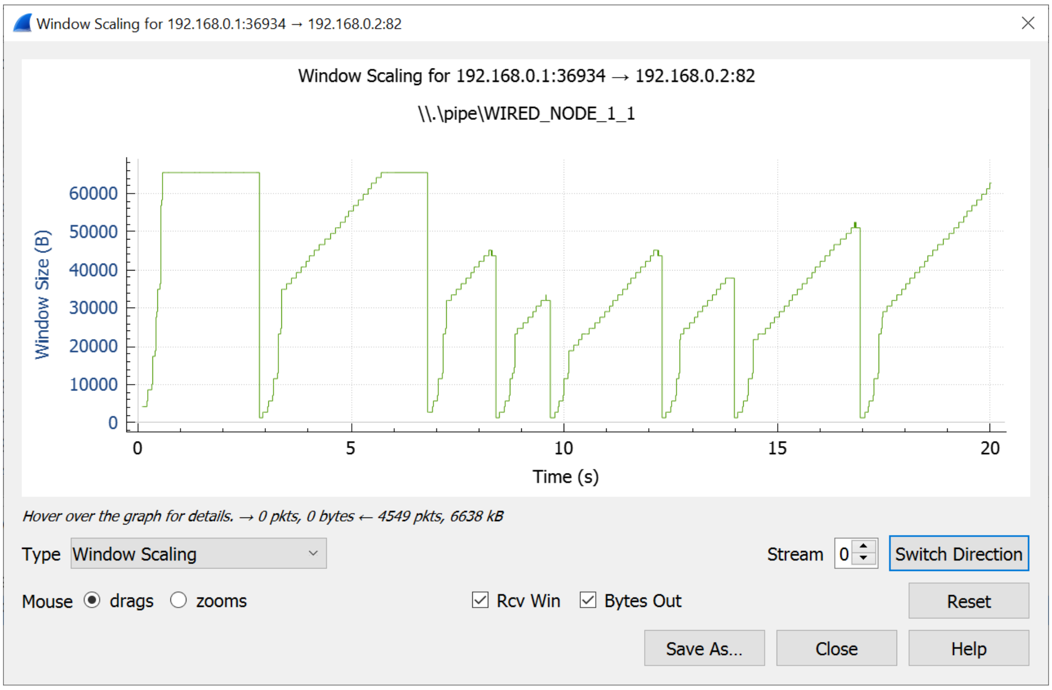


Figure 4-25: Congestion window evolution with TCP Tahoe. We note that Tahoe infers packet loss with timeout and triple duplicate acknowledgements, and updates the slow-start threshold ssthresh and congestion window cwnd as ssthresh = cwnd/2 and cwnd = 1

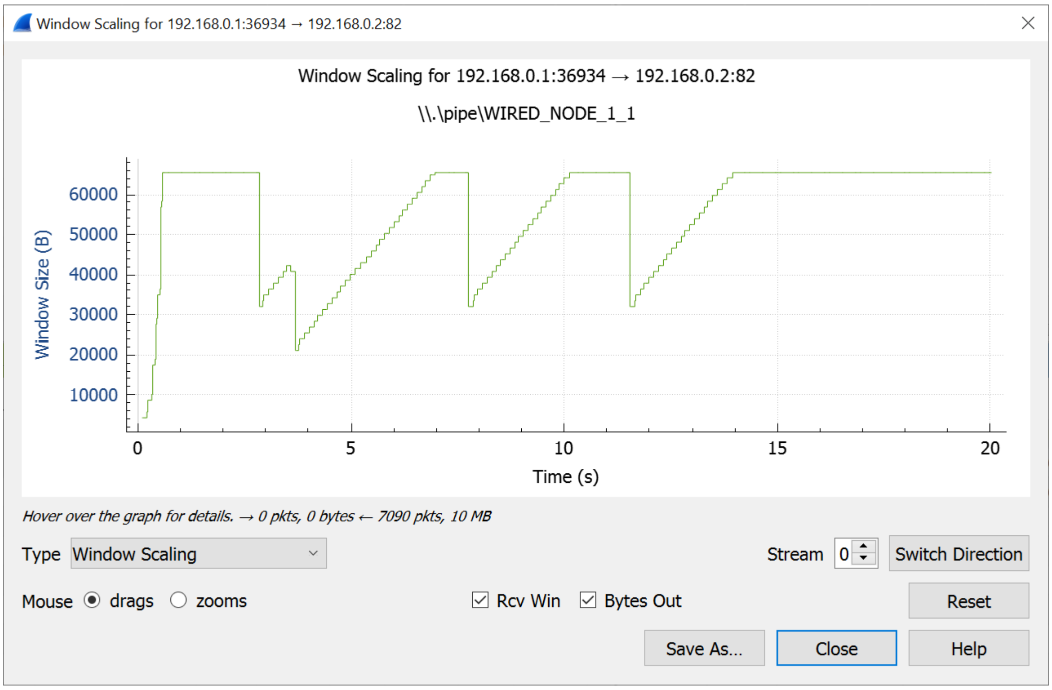


Figure 4-26: Congestion window evolution with TCP New Reno. We note that New Reno infers packet loss with timeout and triple duplicate acknowledgements and updates the slow-start threshold ssthresh and congestion window cwnd as ssthresh = cwnd/2 and cwnd = ssthresh + 3MSS (in the event of triple duplicate acknowledgements).

Congestion Control Algorithm	Throughput (Mbps)
Old Tahoe	2.98
Tahoe	2.62
New Reno	4.12

Table 4-2: Long-term average throughput of the different TCP congestion control algorithms

4.4.5 Observations and Inference

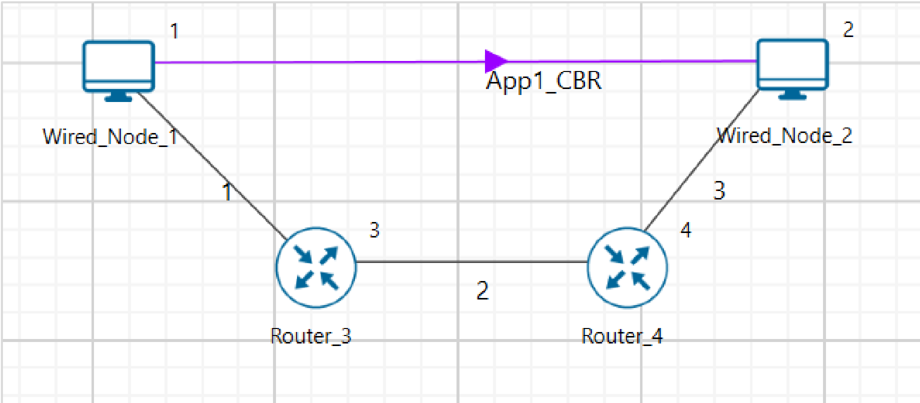
1. We can observe slow start, congestion avoidance, timeout, fast retransmit and recovery phases in the Figure 4-24, Figure 4-25, and Figure 4-26. In Figure 4-24, we note that Old Tahoe employs timeout, slow-start and congestion avoidance for congestion control. In Figure 4-25, we note that Tahoe employs fast retransmit, slow-start and congestion avoidance for congestion control. In Figure 4-26, we note that New Reno employs fast retransmit and recovery, congestion avoidance and slow-start for congestion control.
2. We note that TCP New Reno reports a higher long term average throughput (in comparison with Old Tahoe and Tahoe, see Table 4-2) as it employs fast retransmit and recovery to recover from packet losses.

4.4.6 Exercises

1. Impact of Bit Error Rate on TCP Congestion Control Algorithms

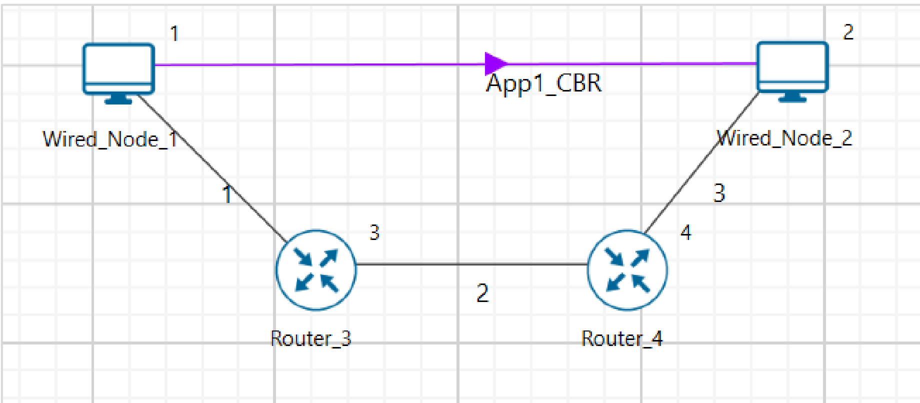
Consider the similar experiment, change the BER value 1×10^{-7}, 2×10^{-7},5×10^{-7}, 1×10^{-6} on link 2, simulate it for 500 seconds and compare the performance of TCP Old Tahoe, Tahoe, and New Reno by analyzing the throughput under different error rates.

Also, attach the window scaling graph obtained from Wireshark for 1×10^{-7} case.



2. Performance analysis of TCP congestion control algorithms with varying Maximum Segment Sizes (MSS)

Consider the similar experiment by varying the Maximum Segment Size (MSS) to 500, 700, 900, and 1100 bytes. Simulate the network for 500 seconds and compare the performance of TCP Old Tahoe, Tahoe, and New Reno by analysing the throughput under different segment sizes.



3. Comparative Analysis of TCP Congestion Control Algorithms for Video Application

Construct the scenario using 2 Routers and 2 Wired node, set the link properties similar to settings done in experiment, configure a video application with a generation rate of 5 Mbps, set transport protocol to TCP and vary the TCP congestion control algorithm. Tabulate the throughput and jitter values obtained for each case and discuss how each TCP congestion control algorithm affected throughput and jitter. Include window scaling screenshots to support your analysis and highlight any notable differences in the graphs.

Configure the Independent Gaussian model with the following values to generate 5 Mbps of video

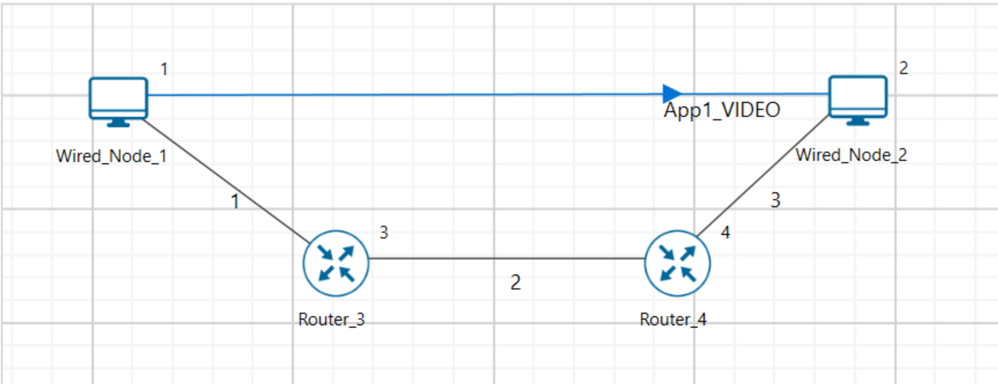
$frames\ per\ second\ (fps)$ = 10

$pixel\ per\ frame\ (ppf)$ = 961538

$Mean, bits\ per\ pixel\ (bpp\ (\mu))$ = 0.52

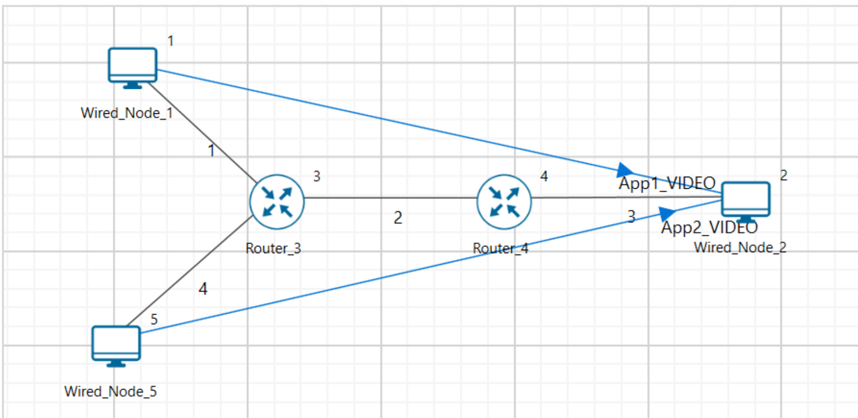
The generation rate for video application can be calculated by using the formula shown below:

$$Generation\ Rate\ (bits\ per\ second)\ =\ fps\ \times\ ppf\ \times bpp$$

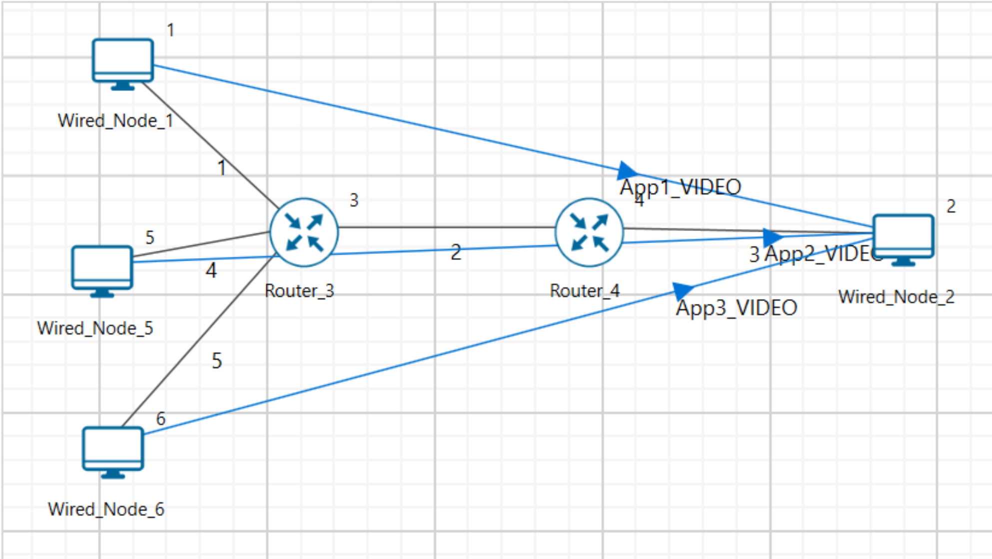


4. Impact of TCP Congestion Control on Video application in Multi-Node Networks

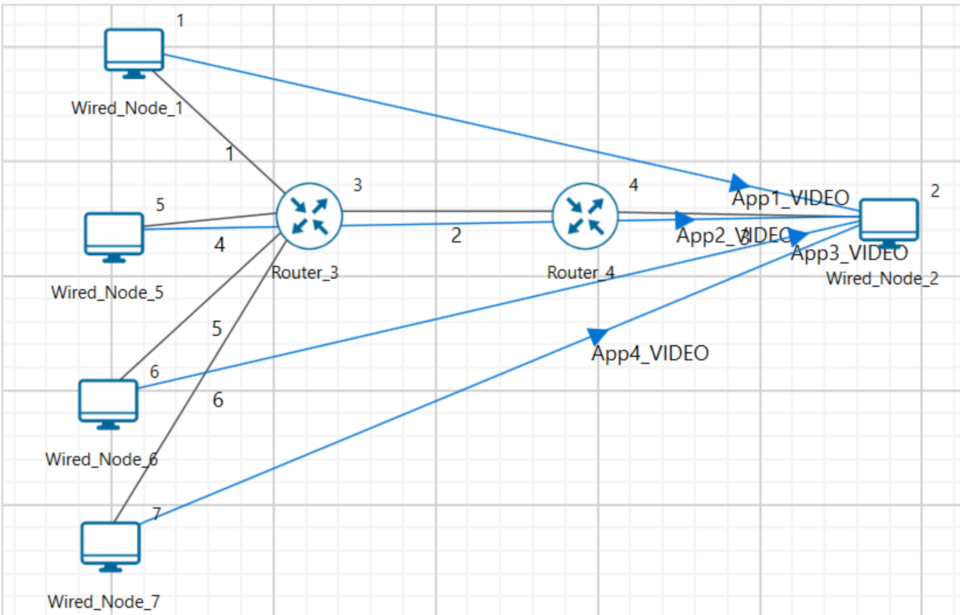
Construct the scenario using 3 wired node and 2 Routers, set the link speed to 20 Mbps, propagation delay and BER to 0, configure a video application with a generation rate of 5 Mbps, set transport protocol to TCP and vary the TCP congestion control algorithm. Tabulate the throughput and jitter values obtained for each case and discuss how each TCP congestion control algorithm affected throughput and jitter.



Case a: Increase the transmitter count to 3 (add one more wired node to the existing setup) and create one video application from newly dropped device and increase the bottleneck link capacity to 30 Mbps. Vary the TCP congestion control algorithms and tabulate the throughput and jitter values obtained for each algorithm.



Case b: Increase the transmitter count to 4 (add one more wired node for case a) and create one video application from newly dropped device and increase the bottleneck link capacity to 40 Mbps. Vary the TCP congestion control algorithms and tabulate the throughput and jitter values obtained for each algorithm.



4.5 Understand the working of TCP BIC Congestion control algorithm, simulate, and plot the TCP congestion window (Level 2)

4.5.1 Theory

In BIC congestion control is viewed as a searching problem in which the system can give yes/no feedback through packet loss as to whether the current sending rate (or window) is larger than the network capacity. The current minimum window can be estimated as the window size at which the flow does not see any packet loss. If the maximum window size is known, we can apply a binary search technique to set the target window size to the midpoint of the maximum and minimum. As increasing to the target, if it gives any packet loss, the current window can be treated as a new maximum and the reduced window size after the packet loss can be the new minimum. The midpoint between these new values becomes a new target. Since the network incurs loss around the new maximum but did not do so around the new minimum, the target window size must be in the middle of the two values. After reaching the target and if it gives no packet loss, then the current window size becomes a new minimum, and a new target is calculated. This process is repeated with the updated minimum and maximum until the difference between the maximum and the minimum falls below a preset threshold, called the minimum increment (S_{min}). This technique is called binary search increase.

Additive Increase

To ensure faster convergence and RTT-fairness, binary search increase is combined with an additive increase strategy. When the distance to the midpoint from the current minimum is too large, increasing the window size directly to that midpoint might add too much stress to the network. When the distance from the current window size to the target in binary search increase is larger than a prescribed maximum step, called the maximum increment (Smax) instead of increasing window directly to that midpoint in the next RTT, we increase it by Smax until the distance becomes less than Smax, at which time window increases directly to the target. Thus, after a large window reduction, the strategy initially increases the window linearly, and then increases logarithmically. This combination of binary search increase and additive increase is called as binary increase. Combined with a multiplicative decrease strategy, binary increase becomes close to pure additive increase under large windows. This is because a larger window results in a larger reduction by multiplicative decrease and therefore, a longer additive increase period. When the window size is small, it becomes close to pure binary search increase – a shorter additive increase period.

Slow Start

After the window grows past the current maximum, the maximum is unknown. At this time, binary search sets its maximum to be a default maximum (a large constant) and the current window size to be the minimum. So, the target midpoint can be very far. According to binary increase, if the target midpoint is very large, it increases linearly by the maximum increment. Instead, run a “slow start” strategy to probe for a new maximum up to Smax. So if cwnd is the current window and the maximum increment is Smax, then it increases in each RTT round in steps cwnd+1, cwnd+2, cwnd+4,,, cwnd+Smax. The rationale is that since it is likely to be at the saturation point and also the maximum is unknown, it probes for available bandwidth in a “slow start” until it is safe to increase the window by Smax. After slow start, it switches to binary increase.

Fast Convergence

It can be shown that under a completely synchronized loss model, binary search increase combined with multiplicative decrease converges to a fair share. Suppose there are two flows with different window sizes, but with the same RTT. Since the larger window reduces more in multiplicative decrease (with a fixed factor β), the time to reach the target is longer for a larger window. However, its convergence time can be very long. In binary search increase, it takes $log(d) - log(Smin)$ RTT rounds to reach the maximum window after a window reduction of d. Since the window increases in a log step, the larger window and smaller window can reach back to their respective maxima very fast almost at the same time (although the smaller window flow gets to its maximum slightly faster). Thus, the smaller window flow ends up taking away only a small amount of bandwidth from the larger flow before the next window reduction. To remedy this behaviour, binary search increase is modified as follows. After a window reduction, new maximum and minimum are set. Suppose these values are max_wini and min_wini for flow i (i =1, 2). If the new maximum is less than the previous, this window is in a downward trend. Then, readjust the new maximum to be the same as the new target window (i.e. max_wini = (max_wini-min_wini)/2), and then readjust the target. After that apply the normal binary increase. This strategy is called fast convergence.

4.5.2 Network setup

Open NetSim and click on **Experiments> Internetworks> TCP> Advanced TCP BIC Congestion control algorithm** then click on the tile in the middle panel to load the example as shown in below Figure 4-27.

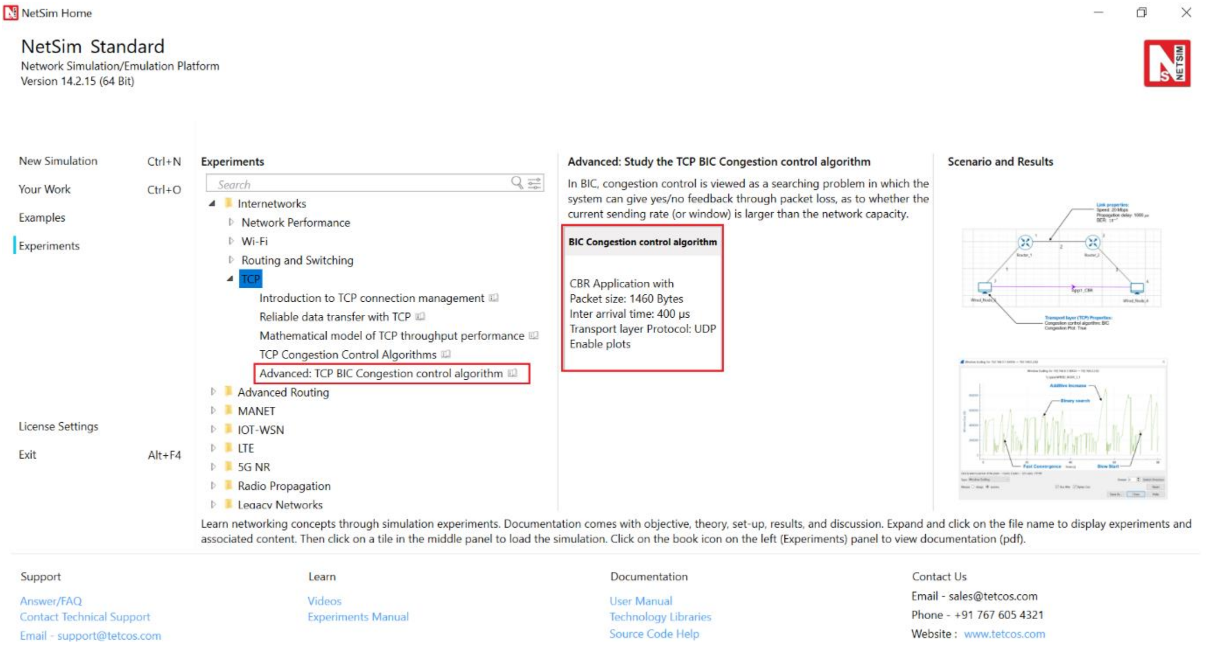


Figure 4-27: List of scenarios for the example of Advanced TCP BIC Congestion control algorithm

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 4-28.

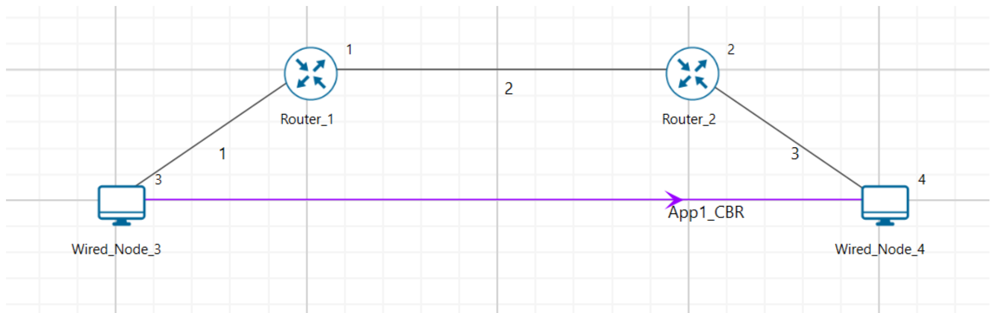


Figure 4-28: Network set up for studying the Advanced TCP BIC Congestion control algorithm

4.5.3 Procedure

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 3 i.e., Source, Wireshark Capture is set to Online and in the TRANSPORT LAYER Properties, Window Scaling is set as TRUE.

To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned in steps.

Step 3: For all the devices, in the TRANSPORT LAYER Properties, Congestion Control Algorithm is set to BIC.

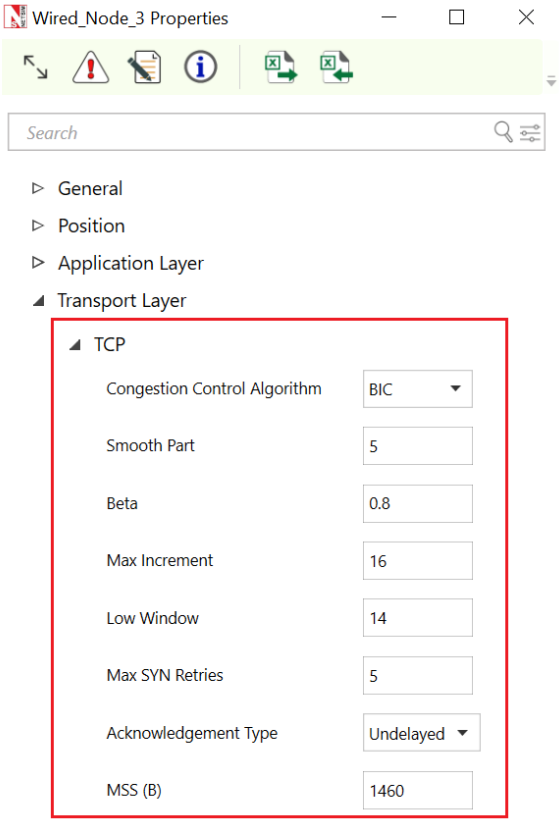


Figure 4-29: Transport Layer window

Step 4: The Link Properties are set according to the table given below Table 4-3. To configure link properties, click on link expand the link properties window on right and set as mentioned below.

Link Properties	Wired Link 1	Wired Link 2	Wired Link 3
Uplink Speed (Mbps)	100	20	100
Downlink Speed (Mbps)	100	20	100
Uplink propagation delay (µs)	5	1000	5
Downlink propagation delay (µs)	5	1000	5
Uplink BER	0.00000001	0.00000001	0.00000001
Downlink BER	0.00000001	0.00000001	0.00000001

Table 4-3: Wired Link Properties

Step 5: Configure applications between two nodes by selecting an application from Set Traffic Tab from ribbon on top. Click on the application flow **App1 CBR** and expand application property panel on right and start time to 20 seconds, Packet Size set to 1460 Bytes and Inter Arrival Time set to 400 µs.

The Packet Size and Inter Arrival Time parameters are set such that the Generation Rate equals 140 Kbps. Generation Rate can be calculated using the formula:

$$Generation\ Rate\ (Mbps)\ =\ Packet\ Size\ (Bytes)\ *\ 8/Interarrival\ time\ (\mu s)$$

Step 6: Enable the plots from configure reports tab and click on Run simulation. The simulation time is set to 100 seconds.

4.5.4 Output

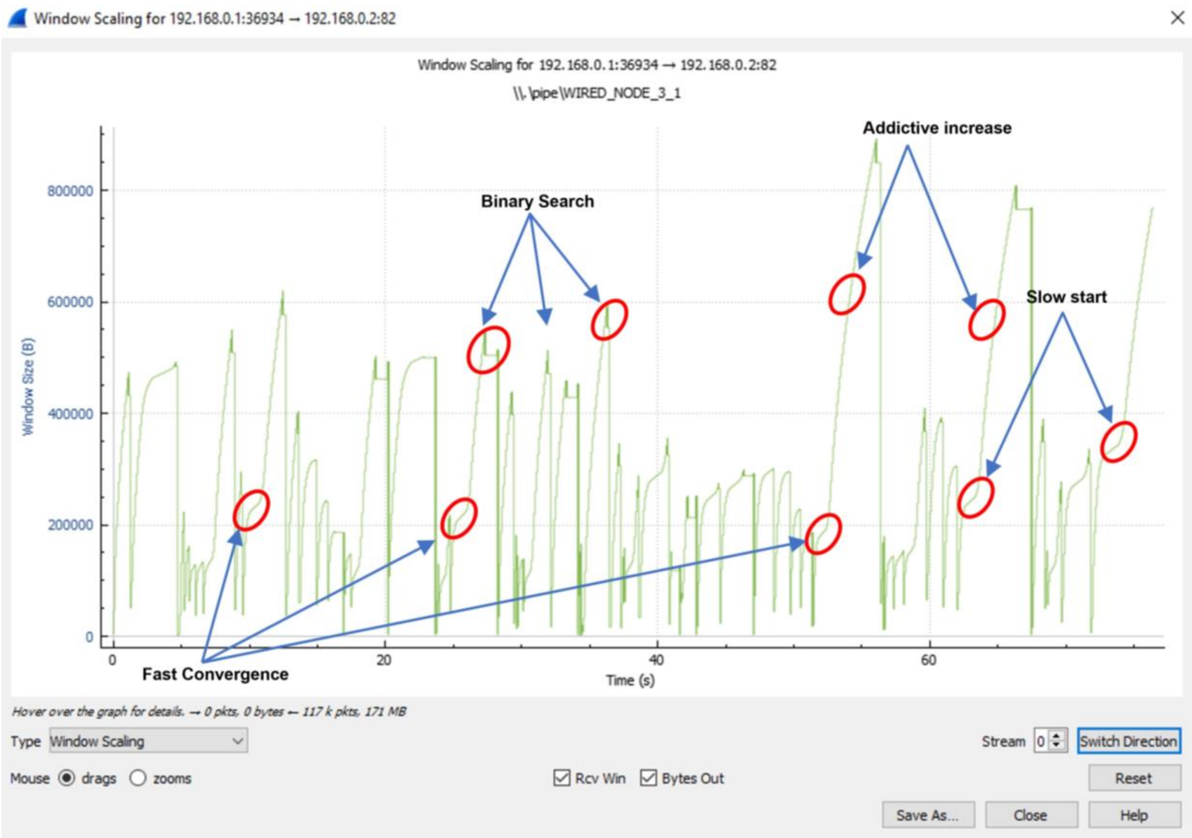


Figure 4-30: Plot of Window Scaling in Wireshark Capture

NOTE: User need to “zoom in” to get the above plot.

Go to the Wireshark Capture window.

Click on data packet i.e. <None>. **Go to Statistics → TCP Stream Graphs → Window Scaling**.

Click on **Switch Direction** in the window scaling graph window to view the graph.

(For more guidance, refer to section - 8.5.5 Window Scaling” in user manual)

The graph shown above is a plot of Congestion Window vs Time of BIC for the scenario shown above. Each point on the graph represents the congestion window at the time when the packet is sent. You can observe Binary Search, Additive Increase, Fast Convergence, Slow Start phases in the above graph.

5 Wi-Fi: IEEE 802.11

5.1 Wi-Fi: Throughput variation with distance (Level 1)

5.1.1 Introduction

In this experiment we will study how the downlink UDP throughput from an Access Point (AP) and a Station (STA) varies with the distance between these two devices. While the wireless link scheduling mechanism remains the same (as per the standard), the bit rate at which the AP and STA digital transceivers reliably operate depends on the received signal power at the receiver (i.e., the PHY rate), and the noise plus interference at the receiver. In this experiment, since we have only one AP-STA pair, there is no interference. Further, since the STA has no data to send, there is also no contention. Hence, the transfer of UDP packets from the AP to the STA comprises alternating back-off periods at the AP and packet transmission times. The back-off periods are random but from the initial back-off distribution. The transmission times depend on the PHY rate. Thus, effective transmission time will vary as per the following express.

$$MeanEffectiveTransmissionTime = MeanInitialBackoff + PacketOverhead + \frac{PacketLength}{PHYRate}$$

Of the three terms on the right, the third term depends on the AP-STA distance, since the PHY rate decreases for increasing distance. Since we are interested in the packet throughput, we can write:

$$PacketThroughput = 1/MeanEffectiveTransmissionTime$$

The above expression is correct only if every packet sent by the AP is correctly decoded by the STA. Even if the PHY rate is correctly chosen so that the SNR at the mean received power renders every packet decodable, in practice, the received power varies randomly due to the phenomenon of shadowing and multipath fading. Due to these phenomena, while the mean received power may be adequate, the actual received power can drop so much that there are packet losses. Although, NetSim can model the phenomenon of shadowing and fading for this experiment we turn this feature off and consider a constant pathloss model.

5.1.2 Simplified pathloss model

The complexity of signal propagation makes it difficult to obtain a single model that characterizes path loss accurately across a range of different environments. For general tradeoff analysis of various system designs it is sometimes best to use a simple model that captures the essence of signal propagation without resorting to complicated path loss models, which are only approximations to the real channel anyway. The following simplified model for path loss as a function of distance is commonly used for system design:

$$P_r = P_t \times c_0 \times \left(\frac{d_0}{d}\right)^{\eta}$$

In this approximation, P_r is the received power sometimes called received signal strength (RSS), P_t is the transmit power, c_0 is the path loss at the "reference" distance, d_0 (usually 1m), η is the path-loss exponent and d is the distance between the transmitter and the receiver. The dB attenuation is thus

$$P_r(dBm) = P_t(dBm) + c_0\ (dB) - 10 \times \eta \times \log_{10}\left(\frac{d}{d_0}\right)$$

As d increases, the received power decreases, e.g., doubling the distance reduces the received power by approximately 3η, since $\log_{10} 2 \approx 0.3$. Typical values of η, indoors, could be 3 to 5, resulting in 9 dB to 15 dB additional path loss for doubling the value of d.

5.1.3 The IEEE 802.11g PHY Rates Table

IEEE 802.11g utilizes OFDM over the entire 20 MHz channel. There are 52 OFDM carriers, of which 48 carriers are used for data transmission and 4 are used for control. The OFDM symbol rate is 250 Ksps. With 48 symbols being sent together (across the 48 carriers), we obtain 12 Msps. In principle, each symbol can be modulated according to any of the modulation and coding schemes (MCS) shown in table 1.

The bit rate shown in the last column of the table assumes the situation in which all the symbols are modulated using the same MCS. Thus, for example, the MCS in the row indexed by 4 uses 16 QAM (i.e., 4 bits per symbol) with a coding rate of 1/2 (i.e., half the bits are data bits, and the rest are channel error protection bits), yielding 2 data bits per symbol, and, therefore, 24 Mbps overall bit rate, assuming that all OFDM symbols use the MCS.

MCS stands for modulation and coding scheme. The MCS defines the numbers of useful bits which can carried by one symbol. In Wi-Fi IEEE 802.11g standard, the MCS depends on the received signal strength (RSS). The higher the signal strength the higher the MCS and more useful bits can be transmitted in a symbol. Thus, the PHY bit rate depends on the MCS chosen. IEEE 802.11g devices can transmit at speeds of 6, 9, 12, 18, 24, 36, 48 and 54Mbps as shown in the table below.

Index	Rx Sensitivity (dBm)	Modulation	Code Rate	Bit Rate
0	-82	BPSK	$1/2$	6 Mbps
1	-81	BPSK	$3/4$	9 Mbps
2	-79	QPSK	$1/2$	12 Mbps
3	-77	QPSK	$3/4$	18 Mbps
4	-74	16 QAM	$1/2$	24 Mbps
5	-70	16 QAM	$3/4$	36 Mbps
6	-66	64 QAM	$2/3$	48 Mbps
7	-65	64 QAM	$3/4$	54 Mbps

Table 5-1: 802.11g bit rates for different modulation schemes, and the minimum received signal power for achieving each bit rate.

In the above table, Rx Sensitivity is the minimum RSS. A simulation assumption in NetSim is that the transmitter knows the RSS at the receiver. Thus, the transmitter chooses the MCS by comparing the RSS against the Receiver-Sensitivity for different MCSs. The highest possible MCS is then chosen.

5.1.4 Calculating distances at which the PHY rate changes

In this section, we predict the AP-STA distance thresholds for the different PHY rates. We know that

$$P_r = P_t - c_0 - 10\,\eta\, log_{10}\,(d)$$

At 2.4 GHz, c_0 is $40.09\ dB$. For $P_t = 100\ mW\ (20\ dBm)$, $\eta = 3.5$, and setting P_r as equal to the receive sensitivity (Ref: Table 5-1), and we get the following inequality for the 6 Mbps PHY rate

$$-82 \le 20 - 40.09 - 35 \times \log(d) < \ -81$$

This gives $54.99m < \ d \le 58.72\ m$. Similarly, we compute the AP-PHY distance for all the rates and arrive at the table below

Rx Sensitivity (dBm)	Bit Rate	$d_{max}\ (m)$	$d_{min}\ (m)$
-82	6 Mbps	58.72	55.00
-81	9 Mbps	54.99	48.22
-79	12 Mbps	48.21	42.28
-77	18 Mbps	42.27	34.69
-74	24 Mbps	34.68	26.68
-70	36 Mbps	26.67	20.52
-66	48 Mbps	20.50	19.20
-65	54 Mbps	19.19	1.00

Table 5-2: We see the maximum and minimum AP-STA distances for different 801.11g PHY bit rates. The PHY rate is 0 for $\boldsymbol{d \ge 58.72m}$. We have chosen $\boldsymbol{P_t = 100\ mW}$ and $\boldsymbol{\eta = 3.5}$.

Table 5-2 can be visualized as shown below.

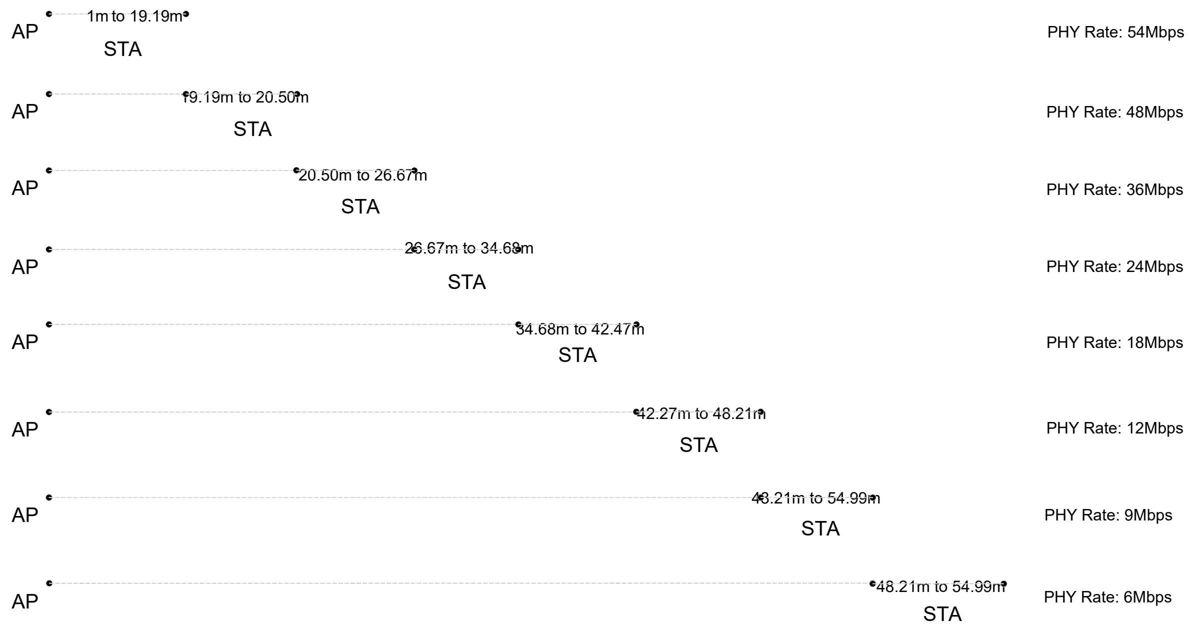


Figure 5-1: Illustration of variation in AP data (PHY) rate vs. distance for $\boldsymbol{P_t = 100\ mW}$ and $\boldsymbol{\eta = 3.5}$

5.1.5 Predicting the throughput

From the experiment Wi-Fi-UDP-Download-Throughput we know that application throughput, θ is

$$\theta = \frac{Application\ Payload\ in\ Packet\ (bits)}{Average\ Time\ per\ Packet(\mu s)}$$

$$Average\ time\ per\ packet\ \ (\mu s)$$
$$= DIFS\ +\ Average\ Backoff\ time\ +\ Packet\ Transmission\ Time\ +\ SIFS$$
$$+\ ACK\ Transmission\ Time$$

Therefore,

$$\theta = \frac{L_{pkt} \times 8}{T_{DIFS} + \left(\frac{CW_{min}}{2} \times T_{slot}\right) + \left(T_{preamble} + \frac{(L_{pkt} + OH) \times 8}{PHYRate}\right) + T_{SIFS} + \left(T_{preamble} + \frac{L_{ACK} \times 8}{PHYRate_{min}}\right)}$$

In the above formula θ is in Mbps as the time in the denominator is in μs.

The predicted application throughput for a 1450B packet, with 68B overheads, ACK size of 14B, and PHY Rate of 54 Mbps is

$$\theta = \frac{1450 \times 8}{34 + \left(\frac{15}{2} \times 9\right) + \left(20 + \frac{(1450 + 68) \times 8}{54}\right) + 16 + \left(20 + \frac{14 \times 8}{6}\right)} = \frac{11600}{401.04} = 28.92\ Mbps$$

Doing the same computation for the different PHY rates leads to the following application throughput predictions. Users just need to replace 54 in the above equation with the appropriate PHY rate.

PHY rate (Mbps)	Predicted Application Throughput (Mbps)
54	28.92
48	27.02
36	22.59
24	17.00
18	13.63
12	9.76
9	7.60
6	5.27

Table 5-3: Predicted application throughput for various PHY rates

In the following section we create scenarios with varying AP-STA distances in NetSim, model the same pathloss equation and compare simulation results against predictions.

5.1.6 Network Setup

Open NetSim and click on **Experiments> Internetworks> Wi-Fi> Wi-Fi Throughput variation with distance** then click on the tile in the middle panel to load the example as shown in below Figure 5-2.

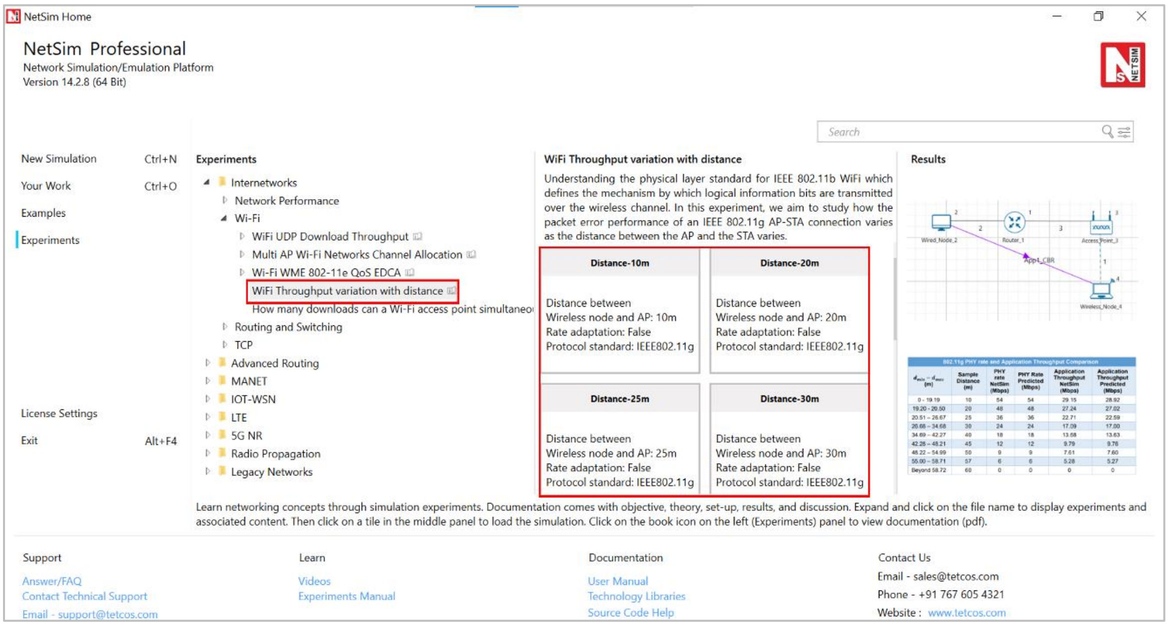


Figure 5-2: List of scenarios for the example of Impact of distance on Wi Fi throughput

NetSim UI displays the configuration file corresponding to this experiment as shown below in Figure 5-3.

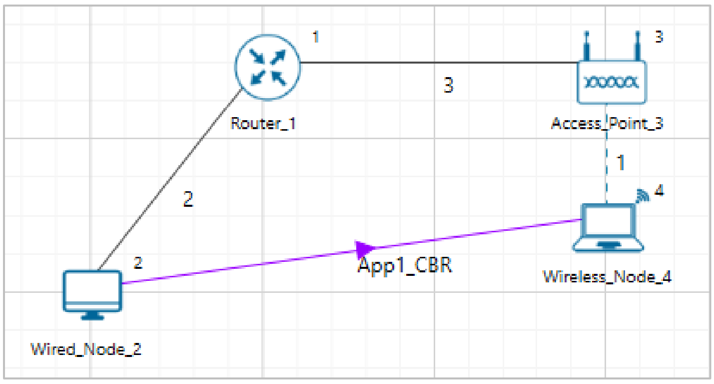


Figure 5-3: Network set up for studying the Impact of distance on Wi-Fi throughput

5.1.7 Procedure

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

Step 1: A network scenario is created in NetSim GUI comprising of 1 Wired Node, 1 Router, 1 Access Point and 1 Wireless Node in the **"Internetworks"** Network Library.

Step 2: In Access Point and Wireless Node 4, the Interface 1 (WIRELESS) > Physical Layer, Protocol Standard is set to IEEE802.11g and in the Interface 1 (WIRELESS) > Datalink Layer, Rate Adaptation is set to False.

To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned above.

Step 3: The position of the Wireless Node and the Access Point in the grid environment is set according to the values given in the below table see Table 5-4. The properties are configured in position layer of device properties.

Device Positions		
	Wireless Node 4	**Access Point**
X	60	60
Y	20	10

Table 5-4: Device Positions

Step 4: Click on Wireless node and expand the property panel on the right, set the media access protocol to DCF in datalink layer of Interface (Wireless) layer. Similarly, set the same in Access point.

Step 5: Click on the link and expand property panel on right and parameters are set according to the values given in the below Table 5-5/Table 5-6.

Wireless Link Properties	
Channel	Path Loss Only
Path Loss Model	Log Distance
Path Loss Exponent	3.5

Table 5-5: Wireless Link Properties

Wired Link Properties	
Max Uplink Speed (Mbps)	100
Max Downlink Speed (Mbps)	100
Uplink BER	0
Downlink BER	0
Uplink Propagation Delay (µs)	0
Downlink Propagation Delay (µs)	0

Table 5-6: Wired Link properties

Step 6: Configure CBR application from Wired Node 2 i.e., Source to Wireless Node 4 i.e., Destination by clicking on the set traffic tab from the ribbon at the top. Click on created application and expand the property panel on right and set packet size to 1450 Bytes, inter arrival time to 200 µs, and transport protocol is set to UDP.

Step 7: Packet Trace is enabled from Configure Reports tab in NetSim GUI. At the end of the simulation, a large .csv file contains all the packet information and is available for the users to perform packet level analysis.

Step 8: Run simulation for 10 sec.

Go back to the scenario and change the distance between Access Point and Wireless Node (i.e., Change the Y axis of Wireless Node (STA)) as 20, 25, 30, 35, 45, 50, 55 and 60 in position layer.

5.1.8 Simulation Output

Data rate can be calculated from packet trace by using the formula given below:

$$PHYRate\ (Mbps) = \frac{PHYLayerPayload\ (B) \times 8}{PHYEndTime\ (\mu s) - PHYArrivalTime(\mu s) - 20(\mu s)}$$

$20\mu s$ is the preamble time for 802.11g.

Given below are the steps to calculate the PHY rate for a 10m AP-STA distance of STA using the packet trace.

Step 1: Make sure that the packet trace is enabled before the simulation, and then run the simulation for a desired time.

Step 2: Click on ‘Packet Trace’.

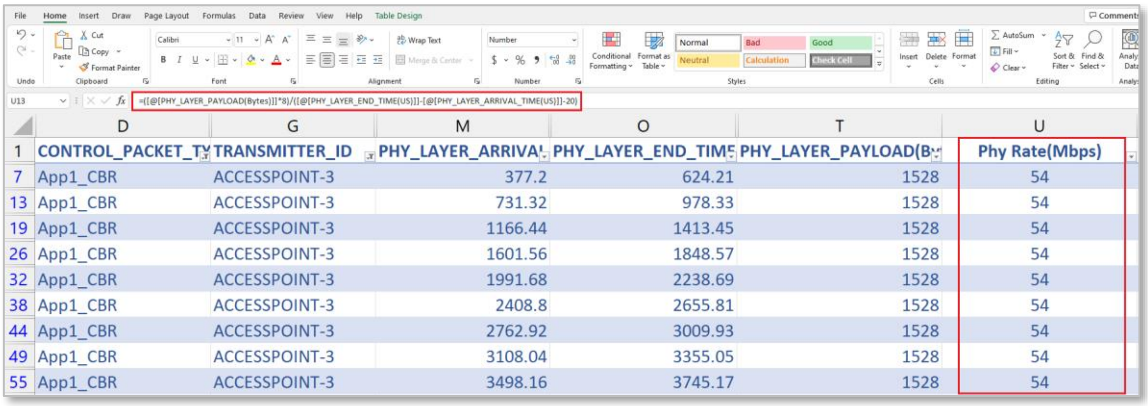
Step 3: Filter CONTROL PACKET TYPE/APP NAME as App1 CBR (Only data packets) and TRANSMITTER ID as ACCESSPOINT-3 (Only wireless channel).

Step 4: Add a column ‘Phy Rate (Mbps)’ after PHY LAYER PAYLOAD(Bytes)

Step 5: Add formula in the first cell of newly added column.

*=([@[PHY_LAYER_PAYLOAD(Bytes)]]*8)/([@[PHY_LAYER_END_TIME(US)]]-*
[@[PHY_LAYER_ARRIVAL_TIME(US)]]-20)

Step 6: We get the Phy Rate as 54Mbps. (The column Phy Rate is formatted to zero decimal places)


=([@[PHY_LAYER_PAYLOAD(Bytes)]]*8)/([@[PHY_LAYER_END_TIME(US)]]-[@[PHY_LAYER_ARRIVAL_TIME(US)]]-20)

	D	G	M	O	T	U
1	CONTROL_PACKET_TY	TRANSMITTER_ID	PHY_LAYER_ARRIVA	PHY_LAYER_END_TIM	PHY_LAYER_PAYLOAD(B	Phy Rate(Mbps)
7	App1_CBR	ACCESSPOINT-3	377.2	624.21	1528	54
13	App1_CBR	ACCESSPOINT-3	731.32	978.33	1528	54
19	App1_CBR	ACCESSPOINT-3	1166.44	1413.45	1528	54
26	App1_CBR	ACCESSPOINT-3	1601.56	1848.57	1528	54
32	App1_CBR	ACCESSPOINT-3	1991.68	2238.69	1528	54
38	App1_CBR	ACCESSPOINT-3	2408.8	2655.81	1528	54
44	App1_CBR	ACCESSPOINT-3	2762.92	3009.93	1528	54
49	App1_CBR	ACCESSPOINT-3	3108.04	3355.05	1528	54
55	App1_CBR	ACCESSPOINT-3	3498.16	3745.17	1528	54

Figure 5-4: Calculating the Phy Rate using its formula in packet trace.

5.1.9 Results

Simulation results with PHY rates and application throughputs are tabulated in Table 5-7.

802.11g PHY rate and Application Throughput Comparison					
$d_{min} - d_{max}$ (m)	Sample Distance (m)	PHY rate NetSim (Mbps)	PHY Rate Predicted (Mbps)	Application Throughput NetSim (Mbps)	Application Throughput Predicted (Mbps)
0 - 19.19	10	54	54	29.15	28.92
19.20 - 20.50	20	48	48	27.24	27.02
20.51 – 26.67	25	36	36	22.71	22.59
26.68 – 34.68	30	24	24	17.09	17.00
34.69 – 42.27	40	18	18	13.68	13.63
42.28 – 48.21	45	12	12	9.79	9.76
48.22 – 54.99	50	9	9	7.61	7.60
55.00 – 58.71	57	6	6	5.28	5.27
Beyond 58.72	60	0	0	0	0

Table 5-7: We see how PHY rate, Application Throughput varies with AP-STA distance

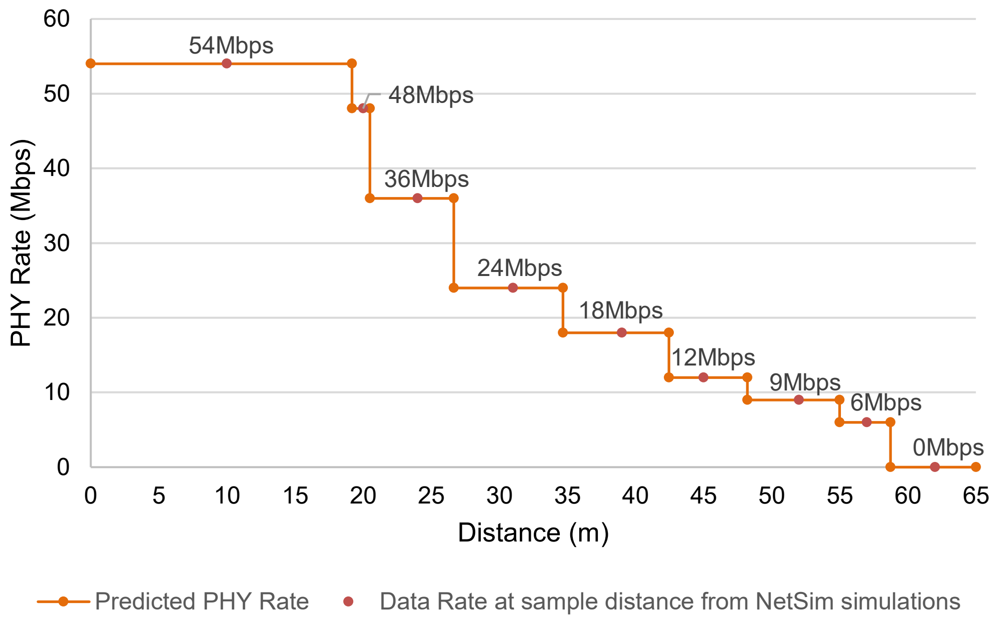


Figure 5-5: Data Rate vs. Distance

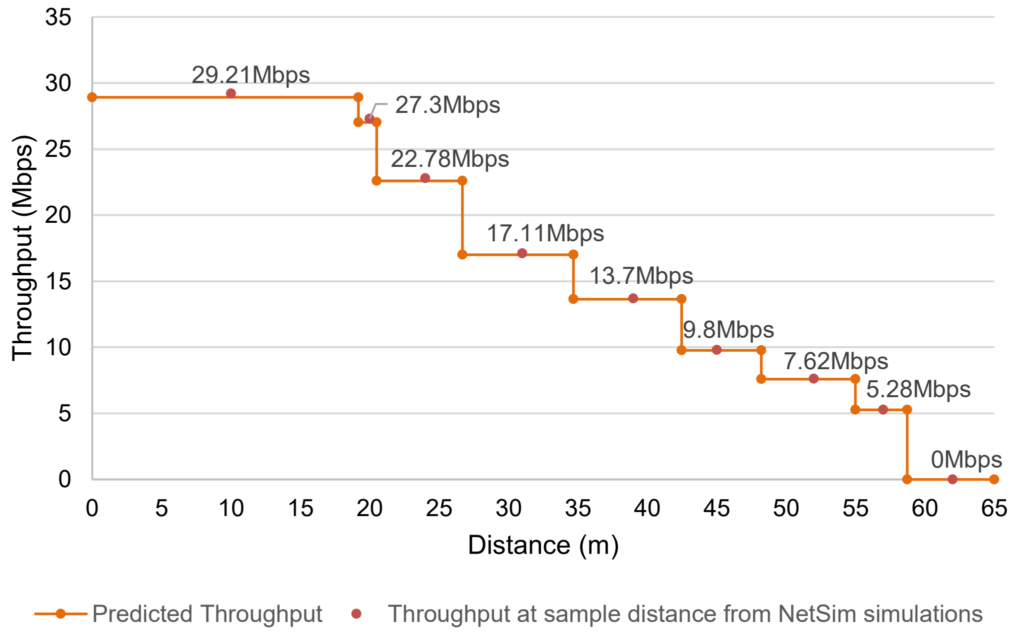


Figure 5-6: Application Throughput vs. Distance

5.1.10 Discussion

It is interesting to note that for 54Mbps PHY rate the application throughput is 29.21 Mbps, which is about 54.1% of the PHY rate. However, for 6 Mbps PHY rate the application throughput is 5.28 Mbps which is 88% of the PHY rate. Why is this so?

Let us go back to the average time per packet – the denominator of the throughput expression - which is

$$T_{DIFS} + \left(\frac{CW_{min}}{2} \times T_{slot}\right) + \left(T_{preamble} + \frac{(L_{pkt} + OH) \times 8}{PHYRate}\right) + T_{SIFS} + \left(T_{preamble} + \frac{L_{ACK} \times 8}{PHYRate_{min}}\right)$$

Per the 802.11g standards the 14 Byte MAC ACK is always sent at the control rate or $PHYRate_{min}$ which is 6 Mbps. Therefore, all terms in the above expression except $\frac{(L_{pkt}+OH)\times 8}{PHYRate}$ do not vary with the PHY rate.

We observe that, since all the "overheads" do not decrease with PHY rate, the lower PHY rates are more "efficient" than the higher PHY rates. This motivates the aggregation of packets in the newer standards (11n, 11ac, etc.), so that the fixed overheads are amortized over substantial number of data bits.

5.1.11 Conclusion

To summarize, we understood the log distance pathloss model and saw how Wi-Fi PHY rates and Application throughputs vary with AP-STA distance. Then we predicted the Wi-Fi PHY rate and application throughputs. Simulation results show excellent agreement with theory.

5.1.12 References

Some of the theoretical content is from the books (i) Wireless Communications by Andrea Goldsmith, and (ii) Wireless Networking by Anurag Kumar.

5.1.13 Exercises

1. Keeping other variables fixed, change the transmit power (P_t) to a different value. Predict d for different PHY rates. Compare against simulation
2. Keeping other variables fixed, change the pathloss exponent η (generally, in the range of 2 to 5). Predict d for different PHY rates. Compare against simulation.
3. Keeping other variables fixed, change the packet size L_{pkt} (from between 100B to 1460B). Take care to ensure the generation rate is sufficiently high to ensure full buffers (saturation) at the transmitting node. Predict θ for different PHY rates. Compare against simulation

5.2 Wi-Fi: UDP Download Throughput (Level 1)

5.2.1 The Setup and Motivation

The most basic packet transfer service offered by the Internet is called the "datagram" service, in which a series of packets are transmitted to a receiver without any packet loss recovery, flow control, or congestion control. The Internet's UDP protocol implements the datagram service. In this experiment, we will study the performance of UDP transfers from a server on a wireline local area network to Wi-Fi Stations (STA), via Wi-Fi Access Points (AP). The schematic of the network that we will be simulating in NetSim is shown in the figure below Figure 5-7.

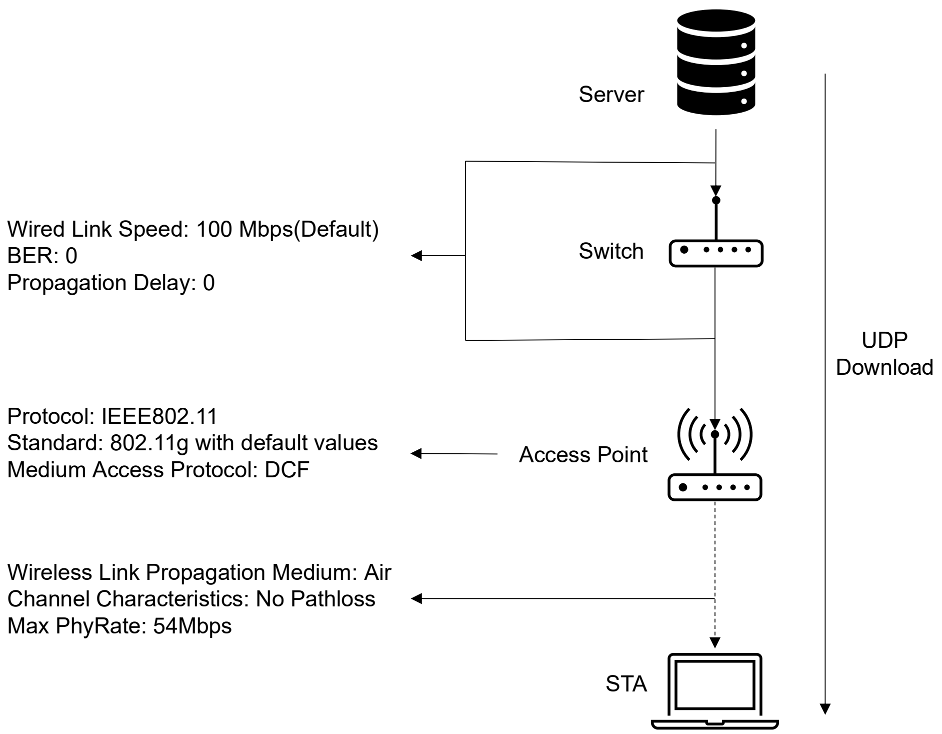


Figure 5-7: Network topology and application flow from server to STA via a single AP

The server, which contains the data that needs to be transferred to the STAs (say, laptops), is connected by a 100 Mbps switched Ethernet link to an Ethernet switch, which is, in turn, connected to the Wi-Fi APs. Each AP is associated (i.e., connected) at a phy rate of 54Mbps (802.11g standard) to a single STA. The objective is to transfer many packets (say, constituting a video) from the server to each of the STAs, the packet stream to each of the STAs being different (e.g., each STA is receiving a different video from the server). In this experiment, we are interested in studying the limitation that the Wi-Fi link places on the data transfers. We assume that the server transmits the packets at a saturation rate of wireless link (i.e., above 54Mbps) so that the queues at the APs fill up, and the rate of the UDP transfers is therefore, governed by the Wi-Fi link. It may be noted that, in practice, there will be a flow control mechanism between each STA and the server, that will control the rate at which the server releases packets, to prevent buffer overflow at the APs.

In this setting, this experiment will ask one precise question. With the buffers at the AP full, at what rate will the Wi-Fi protocol transfer the packets from the APs to the STAs over the wireless link. We will study two cases:

1. A single AP and a single STA: Since there is only one transmitter in this wireless network (namely, the AP), there is no contention, and the rate of packet transfer over the link will be governed by the basic overheads in the protocol, such as the interframe spacings, packet header overheads, transmit-receive turn-around times, and acknowledgement times. We will begin by a simple calculation (essentially timing book-keeping) that will predict the UDP throughput, and then we will verify our calculation using the NetSim simulator.
2. Multiple APs and one STA for each AP: This is the more common situation (for example neighboring apartments in a building, each with one AP and one laptop, all drawing data from the Internet service provider). The performance of such a system depends on the wireless propagation path-loss between the various APs. A predictive analysis is difficult in the general case. For deriving some insight, we will study the case where all the APs are close to each other (i.e. setting channel characteristics as No Pathloss), and thus exactly one transmission from AP to an STA can be successful at any time. If two or more APs transmit together, then all the transmissions are not successful. Even in this case, the analysis mathematically complex and is available in, Anurag Kumar, D. Manjunath and Joy Kuri. 2008: Wireless Networking. Sec 7.4

5.2.2 Predicting the UDP Throughput

5.2.2.1 One AP and one STA

As stated above, in the setup described, the AP queue is full. Thus, after a packet is completely transmitted over the wireless link, immediately the process for transmitting the next packet starts. This is illustrated by the upper part of the Figure 5-8, where the successive packets from the AP are shown as being sent back-to-back. The time taken to send a packet is, however, not just the time to clock out the physical bits corresponding to the packet over the Wi-Fi medium. After the completion of a packet transfer, the AP's Wi-Fi transmitter waits for a Distributed Coordination Function Inter-Frame Space (DIFS), followed by a backoff that is chosen randomly between 1 and 32 slots. Upon the completion of the backoff, the packet transmission starts. Each packet carries physical layer overheads, MAC layer overheads, and IP overheads. After the transmission of the packet, there is a Short Inter-Frame Space (SIFS), which gives time to the receiver (namely, the STA) to transition from the listening mode to the transmit mode. After the SIFS, the STA sends back a MAC acknowledgement (ACK). This completes the transmission of one UDP packet from the AP to the STA. Immediately, the process for sending the next packet can start. The details of the various timings involved in sending a single UDP packet are shown in the lower part of the figure below Figure 5-8.

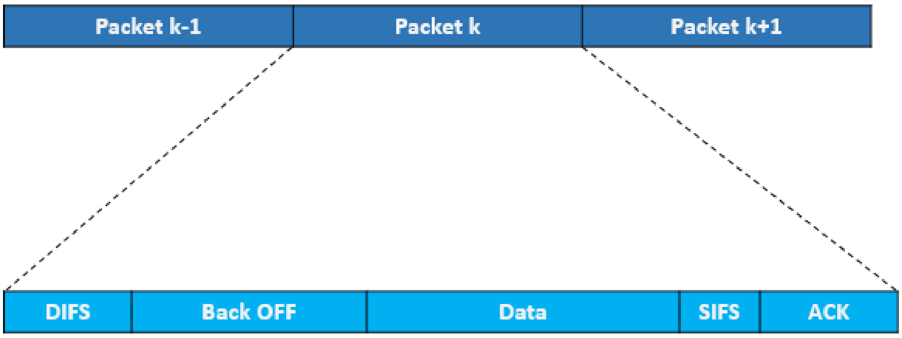


Figure 5-8: Detailed view of transmission of a single packet in Wi-Fi involving single AP-STA.

In this experiment, the payload in each packet is the same (1450 Bytes). Since the packets are sent back-to-back, and the state of the system is identical at the beginning of each packet transmission, the throughput (in Mbps) is computed by the following simple (and intuitive) relation.

5.2.2.1.1 Without RTS-CTS

$$UDP\ Throughput\ (Mbps) = \frac{Application\ Payload\ in\ Packet\ (bits)}{Average\ Time\ per\ Packet(\mu s)}$$

$$Average\ time\ per\ packet\ \ (\mu s) = DIFS\ +\ Average\ Backoff\ time\ +\ Packet\ Transmission\ Time\ +\ SIFS + Ack\ Transmission\ Time$$

$$Packet\ Transmission\ Time\ (\mu s) =\ Preamble\ time\ +\ \left(\frac{MPDUSize}{PHYrate}\right)$$

$$Average\ Backoff\ time\ (\mu s)\ =\ \left(\frac{CWmin}{2}\right)\ \times\ Slot\ Time$$

$$Ack\ Transmission\ Time\ (\mu s) =\ Preamble\ time\ +\ \left(\frac{AckPacketSize}{AckPHYRate}\right)$$

$$DIFS\ (\mu s) =\ SIFS\ +\ 2\ \times\ Slot\ Time$$

$$Average\ Backoff\ time\ (\mu s) = \left(\frac{CWmin}{2}\right)\ \times\ Slot\ Time$$

$$Preamble\ time = 20\ \mu s\ for\ 802.11g$$

$$MPDU\ Size = 1450 + 8 + 20 + 40 = 1518\ Bytes$$

$$Application\ Payload = 1450\ Bytes$$

$$SIFS = 16\ \mu s\ and\ Slot\ Time = 9\ \mu s$$

$$CW_{min} = 15\ slots\ for\ 802.11g$$

$$DIFS = SIFS + 2 \times Slot\ Time = 16\ \mu s + 2 \times 9\mu s = 34\ \mu s$$

$$Average\ Backoff\ Time = \left(\frac{CWmin}{2}\right) \times\ Slot\ Time = \left(\frac{15}{2}\right) \times 9 = 67.5\ \mu s$$

$$Packet\ Transmission\ Time = 20\mu s + \frac{1518\ (B) \times 8}{54\ (Mbps)} = 244.88\ \mu s$$

$$Ack\ Transmission\ Time = 20\mu s + \frac{14\ (B) \times 8}{6\ (Mbps)} = 38.66\ \mu s$$

$$Average\ time\ per\ packet = 34 + 67.5 + 244.88 + 16 + 38.66 = 401.04\ \mu s$$

$$UDP\ Throughput = \frac{1450 \times 8}{401.04} = 28.92\ Mbps$$

5.2.2.1.2 With RTS-CTS

$$UDP\ Throughput\ (Mbps) = \frac{Application\ Payload\ in\ Packet\ (bits)}{Average\ Time\ per\ Packet (\mu s)}$$

$$\begin{aligned} Average\ time\ per\ packet\ \ (\mu s) & \\ &= DIFS + \ RTS\ Packet\ Transmission\ Time + SIFS \\ &+ \ CTS\ Packet\ Transmission\ Time + SIFS + \ Average\ Backoff\ time \\ &+ \ Packet\ Transmission\ Time\ + \ SIFS\ + \ Ack\ Transmission\ Time \end{aligned}$$

$$RTS\ packet\ transmission\ time = Preamble\ time + \left(\frac{RTS\ Packet\ payload}{Data\ Rate}\right) = 20 + \left(\frac{20 \times 8}{6}\right)$$
$$= 46.66\ \mu s$$

$$CTS\ packet\ transmission\ time = Preamble\ time + \left(\frac{CTS\ Packet\ payload}{Data\ Rate}\right) = 20 + \left(\frac{14 \times 8}{6}\right)$$
$$= 38.66\ \mu s$$

$$Average\ time\ per\ packet = 34 + 46.66 + 16 + \ 38.66 + 16 + \ 67.5 + 244.88 + 16 + 38.66$$
$$= 518.36\ \mu s$$

$$UDP\ throughput = \frac{1450 \times 8}{518.36} = 22.37\ Mbps$$

5.2.2.2 Multiple APs (near each other) and one STA per AP

Since the AP queues are full, on the Wi-Fi medium the packet transmission can still be viewed as being back-to-back as shown in the upper part of the figure.

Figure 5-9. However, since there are multiple contending AP-STA links, there are two differences between this figure and the one shown above (for the single AP and single STA case Figure 5-8).

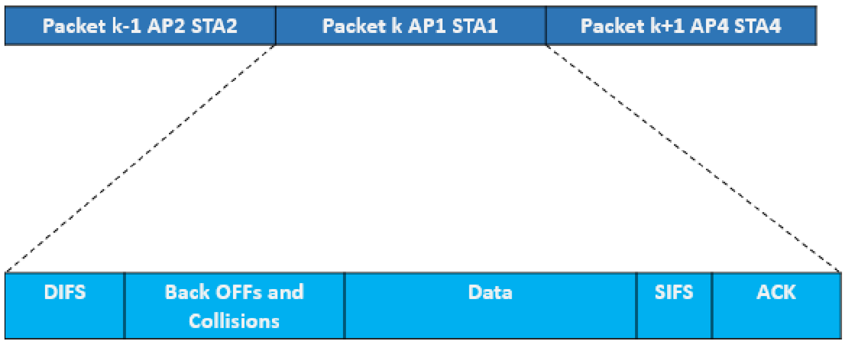


Figure 5-9: Detailed view of transmission of a single packet in Wi-Fi involving Multiple AP-STA

a. Within each transmission period, there is now a “backoffs and collisions” period, where in the figure above we only showed a “backoff” period. Access to the channel is by contention, collision, and backoff, and this “backoffs and collisions” duration is the time taken to select one transmitting AP.

b. The other difference is that, after each “backoffs and collisions” period, any one AP-STA pair “wins” the contention, and the corresponding AP can then send a packet. It turns out that the contention mechanism is such that each of the AP-STA pairs can succeed with equal probability, independent of the pair that has previously been successful. Thus, if there are, say, 5 AP-STA pairs, then each successful packet transmission will be from any of these pairs with a probability of 0.2.

With reference to the figure above, note that, all the APs are contending to send their packets to their respective STAs, and the “Backoffs and Collisions” time is due to all the APs. However, finally, only one packet transmission succeeds. We will attribute all the contention overheads to the successful transmission of this packet. Thus, we will call the time duration from the beginning of a DIFS until the end of the ACK for the transmitted packet as the “effective” time taken to transmit that packet on the wireless medium. The average of these effective packet transmission times can be called the “Average time per Packet.”

With this discussion, and the upper part of the figure above, it follows that the following expression still holds.

$$Total\ UDP\ Throughput\ (Mbps) = \frac{Application\ Payload\ in\ Packet\ (bits)}{Average\ Time\ per\ Packet(\mu s)}$$

We observe from the figure that the average time per packet will be larger than when there is a single AP-STA pair. Hence, the total UDP throughput will be smaller when there are multiple AP-STA pairs (since the “Application Payload in the Packet” is the same in both cases.

Having obtained the total throughput over all the AP-STA pairs in this manner, by the fact that each packet transmission is with equal probability from any of the AP-STA pairs, the UDP throughput for each AP-STA pair (for N pairs) is just $\frac{1}{N}$ of the total throughput.

5.2.3 Network Setup

Open NetSim and click on **Experiments> Internetworks> Wi-Fi> Wi-Fi UDP Download Throughput** then click on the tile in the middle panel to load the example as shown in below Figure 5-10.

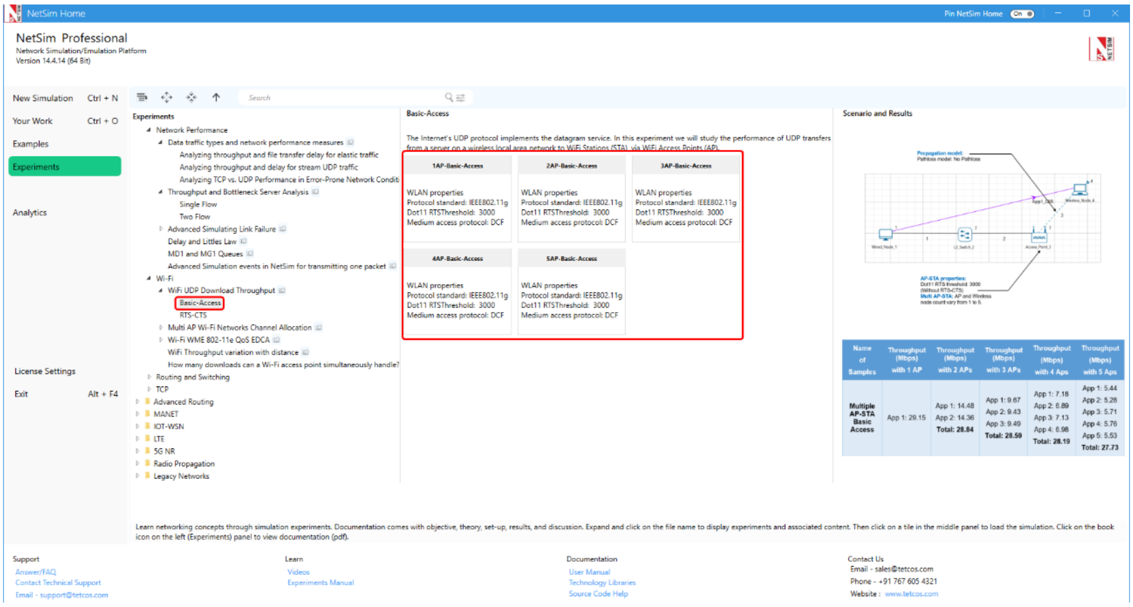
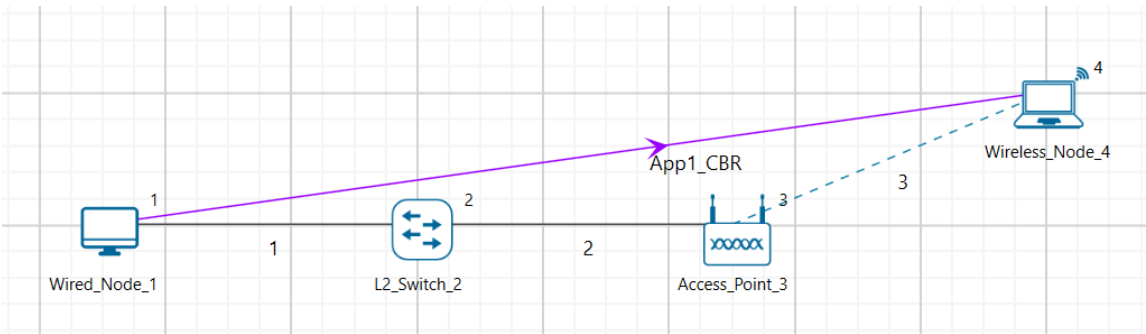


Figure 5-10: List of scenarios for the example of Wi-Fi UDP Download Throughput

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 5-11.



:

Figure 5-11: Network set up for studying the A single AP-STA Without RTS-CTS

5.2.4 Procedure

5.2.4.1 Basic Access

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

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

Step 2: In the Interface Wireless > Physical Layer Properties of Wireless Node 4, Protocol Standard is set to IEEE 802.11g. In the Interface Wireless > Data Link Layer Properties of Wireless Node and Access Point, RTS Threshold is set to 3000. Medium Access Protocol is set to DCF for all nodes.

Step 3: In the Wired Link Properties, Bit Error Rate and Propagation Delay is set to the default value.

Step 4: In the Wireless Link Properties, Channel Characteristics is set to NO PATH LOSS.

Step 5: Configure a CBR application from Wired Node 1 (Source) to Wireless Node 4 (Destination). Click on the created application, expand the property panel on the right, and set the packet size to 1450 bytes, the inter-arrival time to 200 μs, and the transport protocol to UDP.

The Packet Size and Inter Arrival Time parameters are set such that the Generation Rate equals 58 Mbps. Generation Rate can be calculated using the formula:

$$Generation\ Rate\ (Mbps)\ =\ Packet\ Size\ (Bytes)\ *\ 8/Interarrival\ time\ (\mu s)$$

Step 6: Run the Simulation for 10 Seconds and note down the throughput.

5.2.4.1.1 With RTS-CTS

The following changes in settings are done from the previous sample:

Step 1: In the Interface Wireless > Data Link Layer Properties of Wireless Node and Access Point, RTS Threshold is set to 1000.

Step 2: Run the Simulation for 10 Seconds and note down the throughput.

5.2.4.2 Multiple AP-STA Without RTS-CTS: 2APs

The following changes in settings are done from the previous sample Figure 5-12.

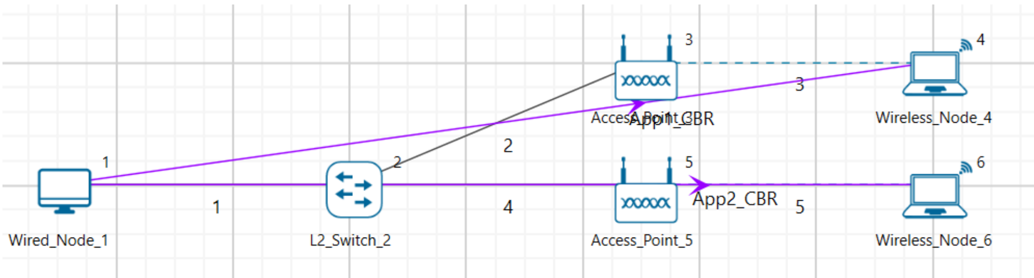


Figure 5-12: Network set up for studying the Multiple AP-STA Without RTS-CTS

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

Step 2: In the Interface Wireless > Data Link Layer Properties of Wireless Node and Access Point, RTS Threshold is set to 3000. Medium Access Protocol is set to DCF for all nodes.

Step 3: Two CBR applications are generated from Wired Node 1 i.e., Source to Wireless Node 4 and Wireless Node 6 i.e., Destination with a Generation Rate of 58 Mbps.

Step 4: Run the Simulation for 10 Seconds and note down the throughput.

Similarly, the subsequent samples are carried out with 3, 4, and 5 Access Points and Wireless Nodes by increasing the number of AP and STA with same PHY and MAC Layer properties set in step 2 and step 3 with applications configured.

5.2.4.3 Multiple AP-STA With RTS-CTS: 2APs

The following changes in settings are done from the previous sample:

Step 1: In the Interface Wireless > Data Link Layer Properties of Wireless Node and Access Point, RTS Threshold is set to 1000. Medium Access Protocol is set to DCF for all nodes.

Step 2: Run the Simulation for 10 Seconds and note down the throughput.

Similarly, the subsequent samples are carried out with 3, 4, and 5 Access Points and Wireless Nodes.

NOTE: For the Next sample Newly added devices and links change the properties are per Without RTS/CTS example except RTS Threshold value.

5.2.5 Output Without and with RTS-CTS

After running simulation, check throughput in Application metrics as shown in the below screenshot Figure 5-13.

Application Metrics

End-to-end performance of applications running across the network.

Application ID	Application Name	Source ID	Destination ID	Throughput (Mbps	Delay (µs)	Jitter (µs)
1	App1_CBR	1	4	29.158920	2253732.120203	156.131221

Figure 5-13: Results for Single AP-STA

Name of Samples	Predicted Throughput (Mbps)	Simulated Throughput (Mbps)
Without RTS-CTS	28.92	29.15
With RTS-CTS	22.37	22.49

Table 5-8:UDP throughput for a single AP-STA, with and without RTS-CTS

5.2.6 Output of Multiple AP-STA Without and with RTS-CTS

After running simulation, check throughput in Application metrics as shown in the below screenshot Figure 5-14.

Application Metrics
End-to-end performance of applications running across the network.

Application ID	Application Name	Source ID	Destination ID	Throughput (Mbps	Delay (µs)	Jitter (µs)
1	App1_CBR	1	4	12.001360	3953219.658422	765.926364
2	App2_CBR	1	6	11.852880	3986073.311487	778.049521

Figure 5-14: Results for Multiple AP-STA

Name of Samples	Throughput (Mbps) with 1 AP	Throughput (Mbps) with 2 APs	Throughput (Mbps) with 3 APs	Throughput (Mbps) with 4 Aps	Throughput (Mbps) with 5 Aps
Multiple AP-STA Basic Access	App 1: 29.15	App 1: 14.48 App 2: 14.36 **Total: 28.84**	App 1: 9.67 App 2: 9.43 App 3: 9.49 **Total: 28.59**	App 1: 7.18 App 2: 6.89 App 3: 7.13 App 4: 6.98 **Total: 28.19**	App 1: 5.44 App 2: 5.28 App 3: 5.71 App 4: 5.76 App 5: 5.53 **Total: 27.73**
Multiple AP-STA With RTS-CTS	App 1: 22.49	App 1: 12.00 App 2: 11.85 **Total: 23.85**	App 1: 9.36 App 2: 8.02 App 3: 6.78 **Total: 24.18**	App 1: 8.20 App 2: 6.51 App 3: 4.91 App 4: 4.58 **Total: 24.21**	App 1: 7.45 App 2: 5.44 App 3: 4.37 App 4: 3.82 App 5: 3.09 **Total: 24.17**

Table 5-9: UDP throughput for 2, 3, 4, and 5 AP-STA pairs, with and without RTS-CTS.

5.2.7 Discussion

Table **5-8** shows the AP-STA UDP throughput (predicted and simulated) for a single AP-STA. Table 5-9 shows the UDP throughputs for 2, 3, 4, and 5 AP-STA pairs; the total throughput is shown along with the individual AP-STA throughputs. We can make the following observations, along with explanations (as bulleted comments) for the observations.

1. The UDP throughput with RTS/CTS turned off is larger than when RTS/CTS is used.
   - The reduction in throughput with RTS/CTS is due the RTS/CTS overheads. The RTS/CTS mechanism aims at alerting “hidden” nodes that a transmission is about to start and can reduce collisions if there are hidden nodes. Since in this experiment all nodes can directly hear each other’s transmissions, the Basic Access mode suffices, whereas RTS/CTS only adds overhead.
2. In RTS-CTS case, the UDP throughput increases slightly with 2, 3 and 4 AP-STA pairs than with just one.
   - With just one AP-STA pair, there is wastage of time due to backoffs, even when there is no possibility of contention. When one more AP-STA is added some of this wastage

is compensated by two APs attempting, with the possibility that one of them might finish its backoff early and grab the channel, thus reducing the backoff overhead. There is, of course, the additional time wasted due to collisions, but the balance between these two opposing phenomena is such that there is a small gain in throughput.

3. Further increase in the number AP-STA pairs leads to a decrease in throughput, but the decrease is small.

    - The IEEE 802.11 Distributed Coordination Function (DCF) manages the sharing of the Wi-Fi channel in a distributed manner. If there was a centralized scheduler than each AP could be scheduled by turn, without any backoff and collision overheads, and the total throughput would have been just that due to sending UDP packets back-to-back: $\frac{1450\times 8}{244.88} = 47.37$ Mbps. Thus, the total throughput with DCF is smaller than if the UDP packets were being sent back-to-back, about 28 Mbps rather than 47.37 Mbps. However, DCF implements an adaptive attempt rate mechanism, which causes nodes to attempt less aggressively as the number of contending nodes increases. It is this mechanism that prevents the total throughput from dropping steeply as the number of AP-STA pairs increases.

4. The total throughput is distributed roughly equally between the AP-STA pairs.

    - This is another feature of DCF. The contending nodes obtain fair access at the packet level, i.e., each successful packet is from any of the contending nodes with equal probability. The downside of this feature is that if an AP-STA is using long packets, then that UDP flow will get a larger throughput. In this experiment, all the AP-STA UDP flows are using the same packet lengths.

5.3 How many downloads can a Wi-Fi access point simultaneously handle?(Level 2)

5.3.1 Motivation

Wi-Fi has become the system of choice for access to Internet services inside buildings, offices, malls, airports, etc. In order to obtain access to the Internet over Wi-Fi a user connects his/her mobile device (a laptop or a cellphone, for example) to a nearby Wi-Fi access point (AP). A popular use of such a connection is to download a document, or a music file; in such an application, the user's desire is to download the file as quickly as possible, i.e., to get a high throughput during the download. It is a common experience that as the number of users connected to an AP increase, the throughput obtained by all the users decreases, thereby increasing the time taken to download their files. The following question can be asked in this context.

If during the download, a user expects to get a throughput of at least θ bytes per second, what is the maximum number of users (say, n_θ) up to which the throughput obtained by every user is at least θ. We can say that n_θ is the *capacity* of this simple Wi-Fi network for the *Quality of Service (QoS)* objective θ. [4]

5.3.2 Objective

In this experiment we will learn how to obtain n_θ in a simple Wi-Fi network where the packet loss due to channel errors is 0. In this process we will understand some interesting facts about how Wi-Fi networks perform when doing file transfers.

5.3.3 Theory

In NetSim, we will set up a network comprising a server that carries many large files that the users would like to download into their mobile devices. The server is connected to a Wi-Fi AP, with the IEEE 802.11b version of the protocol, via an Ethernet switch. Several mobile devices (say, N) are associated with the AP, each downloading one of the files in the server. The Ethernet speed is 100Mbps, whereas the mobile devices are connected to the AP at 11Mbps, which is one of the IEEE 802.11b speeds.

We observe, from the above description, that the file transfer throughputs will be limited by the wireless links between the AP and the mobile devices (since the Ethernet speed is much larger

[4] It may be noted that the term capacity has several connotations in communications. Our use of the word here must not be confused with the notion of information theoretic capacity of a communication channel.

than the Wi-Fi channel speed). There are two interacting mechanisms that will govern the throughputs that the individual users will get:

1. The Wi-Fi medium access control (MAC) determines how the mobile devices obtain access to the wireless medium. There is one instance of the Wi-Fi MAC at each of the mobile devices.
2. The end-to-end protocol, TCP, controls the sharing of the wireless bandwidth between the ongoing file transfers. In our experiment, there will be one instance of TCP between the server and each of the mobile devices.

For simplicity, the default implementation of TCP in NetSim does not implement the delayed ACK mechanism. This implies that a TCP receiver returns an ACK for every received packet. In the system that we are simulating, the server is the transmitter for all the TCP connections, and each user's mobile device is the corresponding receiver.

Suppose each of the N TCP connection transmits one packet to its corresponding mobile device; then each mobile device will have to return an ACK. For this to happen, the AP must send N packets, and each of the N mobile devices must send back N ACKs. Thus, for the file transfers to progress, the AP needs to N packets for each packet (i.e., ACK) returned by each mobile device. We conclude that, in steady state, the AP must send as many packets as all the mobile devices send, thus requiring equal channel access to the AP as to all the mobile devices together.

At this point, it is important to recall that when several nodes (say, an AP and associated mobile devices) contend for the channel, the Wi-Fi medium access control provides fair access at the packet level, i.e., each contending device has an equal chance of succeeding in transmitting a packet over the channel. Now consider the system that we have set up in this present experiment. There are N mobile devices associated with one AP. Suppose, for example, 10 of them ($N \geq 10$) all have a packet to transmit (and none other has a packet). By the fair access property of the Wi-Fi MAC, each of these 10 nodes, along with the AP, has an equal probability of successfully transmitting. It follows, by the packet level fair access property, that each node will have a probability of $\frac{1}{11}$ of succeeding in transmitting its packet. If this situation continues, the channel access ratio to the AP will be inadequate and the equal channel access argued in the previous paragraph will be violated. It follows from this that, on the average, roughly only one mobile device will have an ACK packet in it; the AP will contend with one other node, thus getting half the packet transmission opportunities.

With the just two nodes contending, the collision probability is small (~ 0.06) and the probability of packet discard is negligibly small. Thus, the TCP window for every transfer will grow to the maximum window size. The entire window worth of TCP data packets for the N sessions will be in the AP buffer, except for a very small number of packets (averaging to about 1) which will appear as ACKs in the mobile devices.

It follows that, in steady state, the system will look like two contending Wi-Fi nodes, one with TCP data packets and the other with TCP ACK packets. This will be the case no matter how many downloading mobile devices there are. The total throughput can be obtained by setting up the model of two saturated nodes, one with TCP data packets, and the other with TCP ACK packets. The data packets of all the TCP connections will be randomly ordered in the AP buffer, so that the head-of-the-line packet will belong to any particular mobile device with probability $\frac{1}{N}$. This throughput is shared equally between the N mobile devices.

Now suppose that the TCP data packet throughput with the two-node model is Θ. Then

$$n_\theta = \lfloor \frac{\Theta}{\theta} \rfloor$$

where the $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. Use NetSim to verify that for an 11Mbps Wi-Fi speed, with RTS/CTS enabled the total TCP throughput is 3.4 Mbps. If $\theta = 0.65\ Mbps$, then $n_\theta = \left\lfloor \frac{3.4}{0.65} \right\rfloor = 5$. In this example, if $N = 5$ the download throughput obtained by each of them will be $0.68 Mbps$, but if one more downloading device is added then each will get a throughput less than $\theta = 0.65\ Mbps$. We say that the capacity of this network for a target throughput of $0.65 Mbps$ is 5.

5.3.4 Network Setup

Open NetSim and click on **Experiments>Internetworks> Wi-Fi> How many downloads can a Wi-Fi access point simultaneously handle?** then click on the tile in the middle panel to load the example as shown in below Figure 5-15.

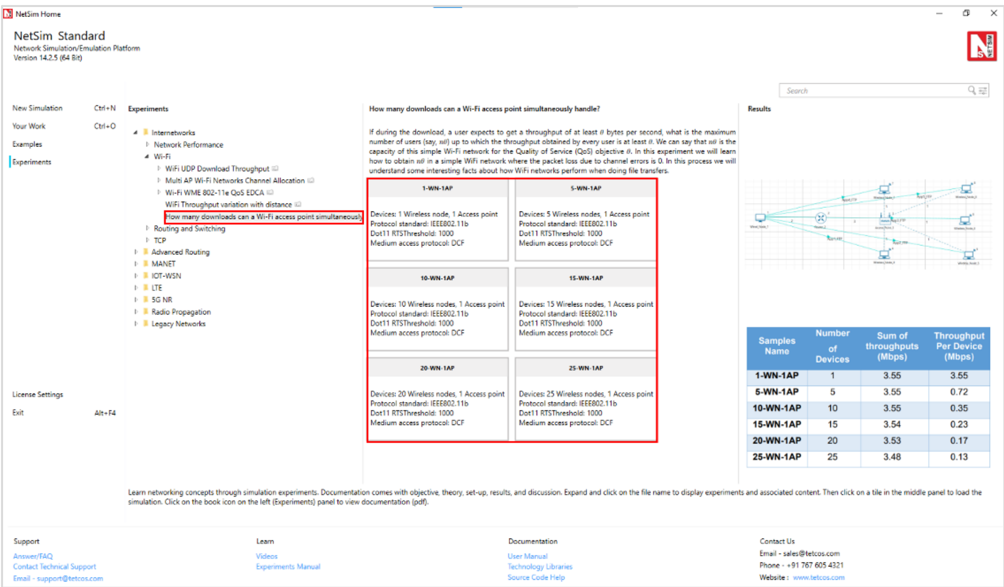


Figure 5-15: List of scenarios for the example of How many downloads can a Wi-Fi access point simultaneously handle

NetSim UI displays the configuration file corresponding to this experiment Figure 5-16.

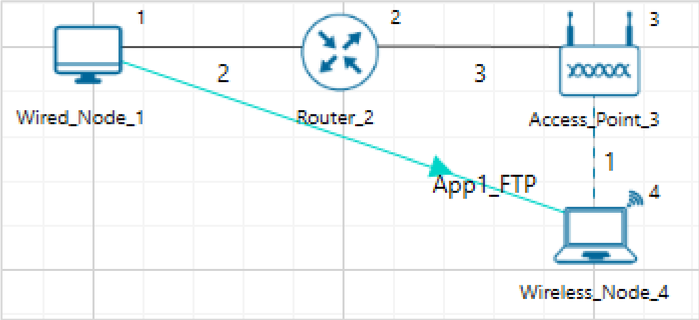


Figure 5-16: Network set up for studying the Wi-Fi Network with Single TCP Download

5.3.5 Procedure

1-WN-1AP

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

Step 1: A network scenario is designed in the NetSim GUI comprising of 1 Wired Node, 1 Wireless Node, 1 Access Point, and 1 Router in the **"Internetworks"** Network Library.

Step 2: Click on the wireless node, expand the property panel on the right, go to the data link layer of the interface (wireless), and set the short retry limit to 7, the long retry limit to 4, the RTS threshold to 1000 bytes, and the medium access protocol to DCF, as shown in Figure 5-17.Similarly, set the same properties in the access point.

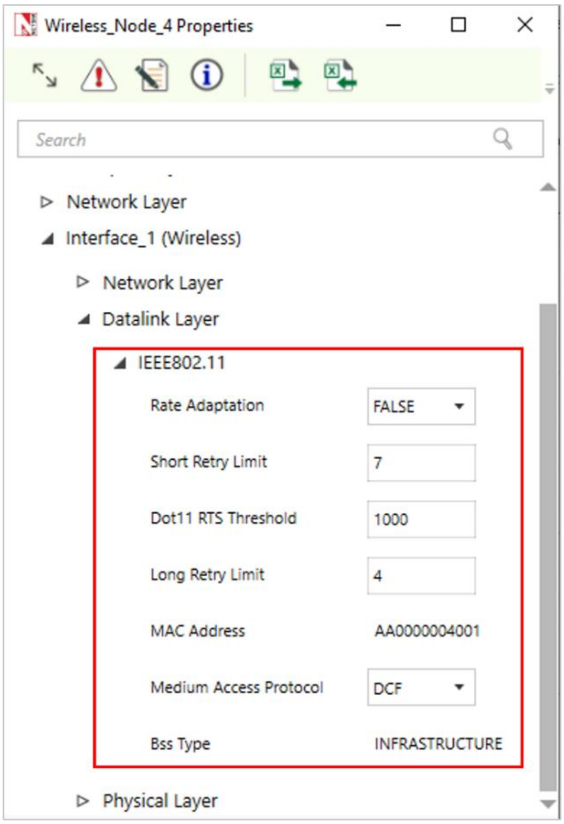


Figure 5-17: Data Link Layer Properties

Step 3: Click on the link and expand the link properties on the right and set the below properties.

Wired Link	
Max Uplink Speed (Mbps)	100
Max Downlink Speed (Mbps)	100
Uplink BER	0
Downlink BER	0
Uplink Propagation Delay	0
Downlink Propagation Delay	0

Table 5-10: Wired link properties

Wireless Link	
Channel Characteristics	No path loss

Table 5-11: Wireless Link properties.

Step 4: Configure FTP application between Wired node 2 and Wireless node 4 by clicking on the set traffic tab from the ribbon on the top. Click on the created application and expand the property panel on right and set the File Size set to 10,000,000 Bytes and Inter Arrival Time set to 20 s.

Step 5: Run the Simulation for 15 Seconds and note down the throughput.

5-WN-1AP

The following changes in settings are done from the previous sample:

Step 1: The number of Wireless Nodes is increased to 5 and FTP applications are generated from Wired Node 2 to each of the Wireless Nodes as shown below Figure 5-18.

No. of wireless nodes = 5

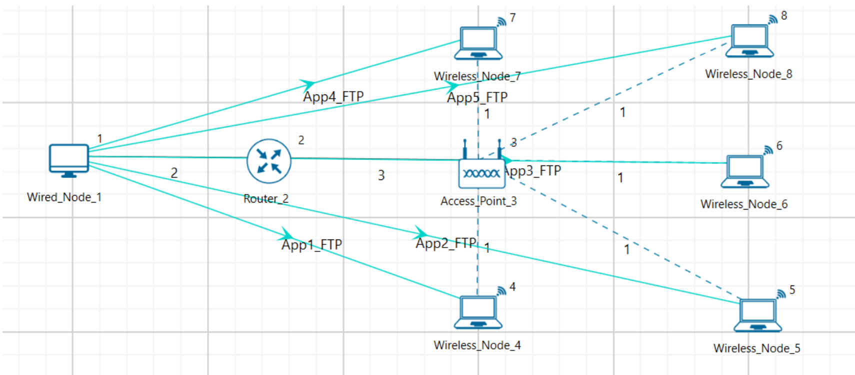


Figure 5-18: Wi-Fi Network with Multiple TCP Download

Application Properties

Properties	App1	App2	App3	App4	App5
Application Type	FTP	FTP	FTP	FTP	FTP
Source Id	1	1	1	1	1
Destination Id	4	5	6	7	8
File size (Bytes)	10,000,000	10,000,000	10,000,000	10,000,000	10,000,000
File Inter arrival time	20 s	20 s	20 s	20 s	20 s

Table 5-12: Detailed Application properties

Step 2: Run the Simulation for 15 Seconds and note down the throughput.

NOTE: Follow the same procedure for next samples with wireless nodes 10, 15, 20, 25 and note down the sum of throughputs for all applications.

5.3.6 Measurements and Output

Aggregated download throughput with different values of N (wireless nodes) is shown below Table 5-13.

$$Throughput\ Per\ Device\ (Mbps) = \frac{Sum\ of\ throughpots\ (Mbps)}{Number\ of\ Devices}$$

Samples Name	Number of Devices	Sum of throughputs (Mbps)	Throughput Per Device (Mbps)
1-WN-1AP	1	3.55	3.55
5-WN-1AP	5	3.55	0.71
10-WN-1AP	10	3.55	0.35
15-WN-1AP	15	3.54	0.23
20-WN-1AP	20	3.53	0.17
25-WN-1AP	25	3.51	0.14

Table 5-13: Aggregated download throughput for different number of wireless nodes

Plot

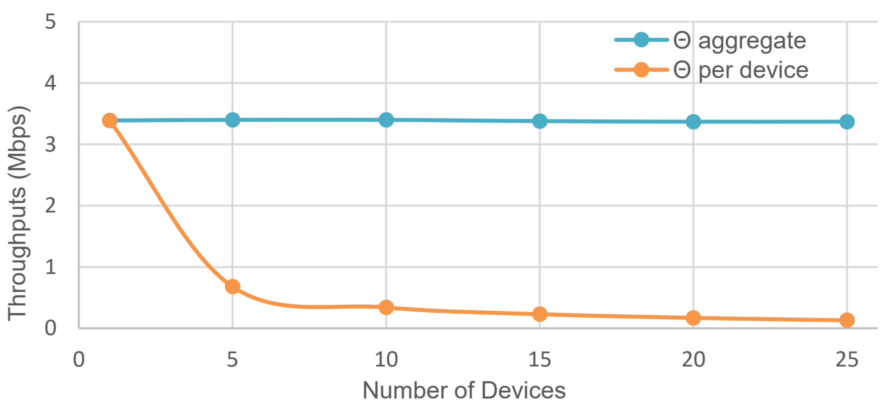


Figure 5-19: Plot of Number of devices vs. Throughputs (Mbps)

NOTE: In the referred paper we see that, a throughput value for 11 Mbps WLAN is 3.8 Mbps. Please note that this is the aggregate PHY throughput of the AP. However, in NetSim, we are calculating the total Application throughput.

To derive the PHY layer throughput from the APP layer throughput, we need to add overheads of all layers observe Table 5-14.

Layer	Overhead (Bytes)
Transport Layer	20
Network Layer	20
MAC Layer	40
PHY layer	48µs = (11*48)/8 = 66
Total Overhead	146

Table 5-14: Overhead of different layers

$$PHY_{Throughput} = APP_{Throughput} * \frac{1606}{1460} = \frac{3.42 * 1606}{1460} = 3.76\ Mbps$$

5.3.7 Observations

We see that as the number of devices increase, the aggregate (combined) throughput remains constant, whereas the throughput per user decreases.

As discussed earlier, our goal was to identify that if during the download, a user expects to get a throughput of at least θ bytes per second, what is the maximum number of users (say, n_θ)?

If we set θ to be 650 Kbps, then we see that from the output table that the maximum number of users who can simultaneously download files is 5 (n_θ)

5.3.8 Reference Documents

1. Analytical models for capacity estimation of IEEE 802.11 WLANs using DCF for internet applications. George Kuriakose, Sri Harsha, Anurag Kumar, Vinod Sharma.

5.4 Multi-AP Wi-Fi Networks: Channel Allocation (Level 2)

5.4.1 Introduction

A single Wi-Fi Access Point (AP) can connect laptops and other devices that are a few meters distance from the AP, the actual coverage depending on the propagation characteristics of the building in which the Wi-Fi network is deployed. Thus, for large office buildings, apartment complexes, etc., a single AP does not suffice, and multiple APs need to be installed, each covering a part of the building. We will focus on 2.4GHz and 5GHz systems. In each of these systems the available bandwidth is organized into channels, with each AP being assigned to one of the channels. For example, 2.4GHz Wi-Fi systems operate in the band 2401MHz to 2495MHz, which has 14 overlapping channels each of 22MHz. There are 3 nonoverlapping channels, namely, Channels 1, 6, and 11, which are centered at 2412MHz, 2437MHz, and 2462MHz. Evidently, if neighboring APs are assigned to the same channel or overlapping channels they will interfere, thereby leading to poor performance. On the other hand, since there are only three nonoverlapping channels, some care must be taken in assigning channels to APs so that nearby APs have nonoverlapping channels, whereas APs that are far apart can use the same or overlapping channels.

In this experiment we will understand some basic issues that arise in multi-AP networks, particularly with attention to channel allocation to the APs.

5.4.2 Network Setup

Open NetSim and click on **Experiments>Internetworks> Wi-Fi> Multi AP Wi-Fi Networks Channel Allocation> APs on the Same channel** then click on the tile in the middle panel to load the example as shown in below

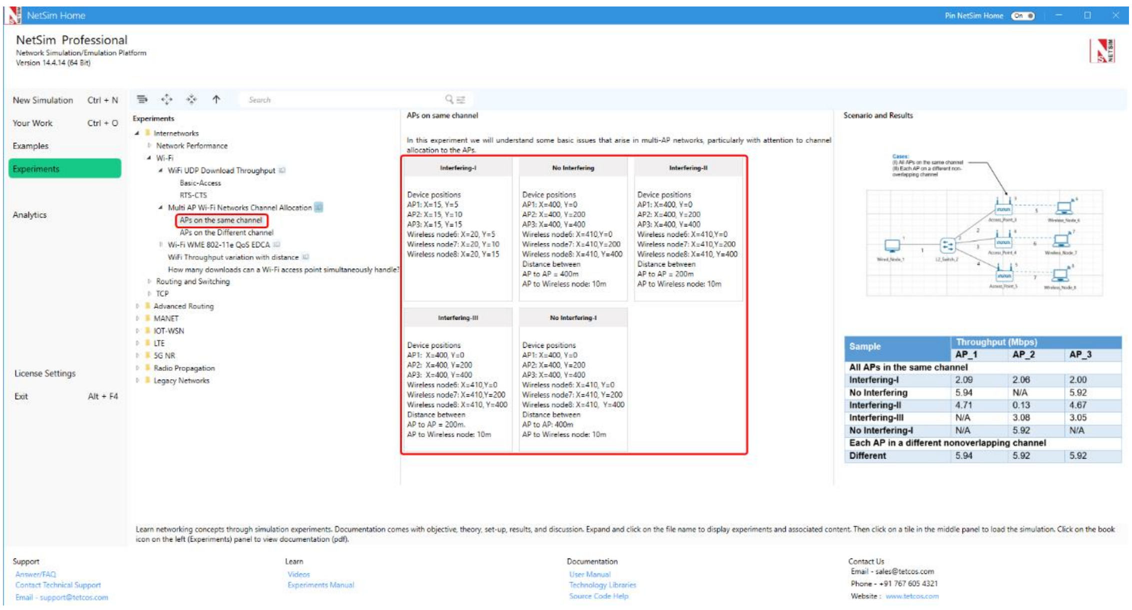


Figure 5-20: List of scenarios for the example of Multi AP Wi-Fi Networks Channel Allocation

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 5-21.

APs in the same channel

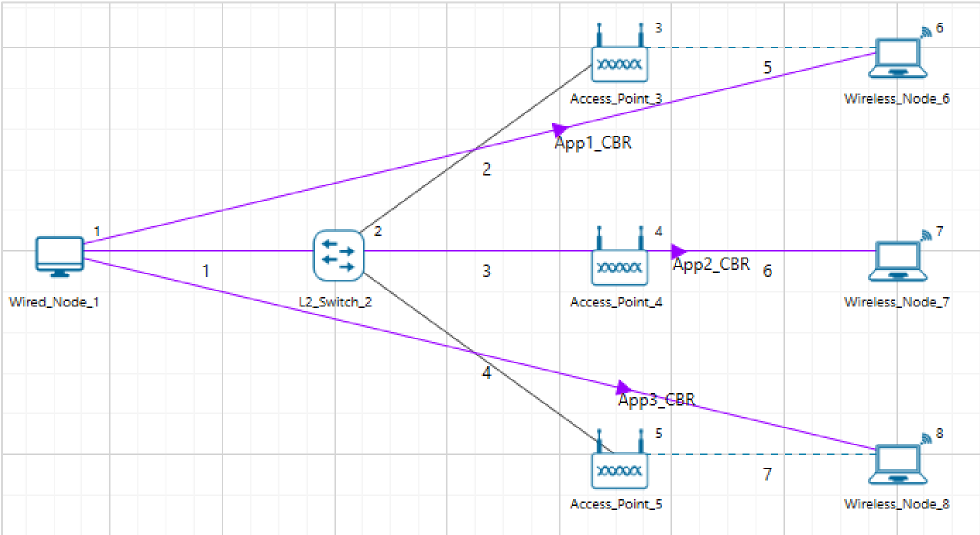


Figure 5-21: Network set up for studying the Multiple APs-Wi-Fi Networks with APs in same Channel - Interfering-I

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

Step 1: Environment Grid length: 40m x 20m.

Step 2: A network scenario is designed in NetSim GUI comprising of 1 Wired Node, 1 L2 Switch, 3 Wireless Nodes and 3 Access Points in the **"Internetworks"** Network Library.

Step 3: The device positions are set as per the table given below Table 5-15.

General Properties		
Device Name	**X / Lon**	**Y / Lat**
AP 3	15	5
AP 4	15	10
AP 5	15	15
Wireless Node 6	20	5
Wireless Node 7	20	10
Wireless Node 8	20	15

Table 5-15: Device positions for APs-STA - Interfering-I

Step 4: In the INTERFACE (WIRELESS) > PHYSICAL LAYER Properties of all the Wireless Nodes and Access Points, the Protocol Standard is set to IEEE 802.11 b. In the INTERFACE (WIRELESS) > DATALINK LAYER Properties of all the Wireless Nodes and Access Points, Medium Access Protocol is set to DCF.

Step 5: Right-click the link ID (of a wired link) and select Properties to access the link's properties. For all the Wired Links, Bit Error Rate and Propagation Delay is set to 0.

Step 6: The Wireless Link Properties are set according to the values given in the below Table 5-16.

Channel Characteristics	PATH LOSS ONLY
Path Loss Model	LOG DISTANCE
Path Loss Exponent	3.5

Table 5-16: Wireless Link Properties

Step 7: Configure the application between any two nodes by selecting an application from the Set Traffic tab**.** Right click on the Application and select properties.

A CBR Application is generated from Wired Node 1 i.e., Source to Wireless Node 6 i.e., Destination with Packet Size set to 1460 Bytes and Inter Arrival Time set to 1168µs.

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

The Packet Size and Inter Arrival Time parameters are set such that the Generation Rate equals 10 Mbps. Generation Rate can be calculated using the formula:

$$Generation\ Rate\ (Mbps)\ =\ Packet\ Size\ (Bytes)\ *\ 8/Interarrival\ time\ (\mu s)$$

Similarly, two more CBR applications are generated from Wired Node 1 to Wireless Node 7 and Wired Node 1 to Wireless Node 8.

No Interfering: The following changes in settings are done from the previous sample:

Step 1: Before we start designing the network scenario, the Grid Length is set to 1000m x 500 m. This can be set by choosing the Menu **Options>Change Grid/Map settings>Grid/Map settings** from the GUI.

Step 2: From the previous sample, we have removed App2 CBR (i.e., from Wired Node1 to Wireless Node7), set distance between the other 2 Access Points (AP 1 and AP 3) as 400m and distance between APs and Wireless nodes as 10m as shown below Figure 5-22.

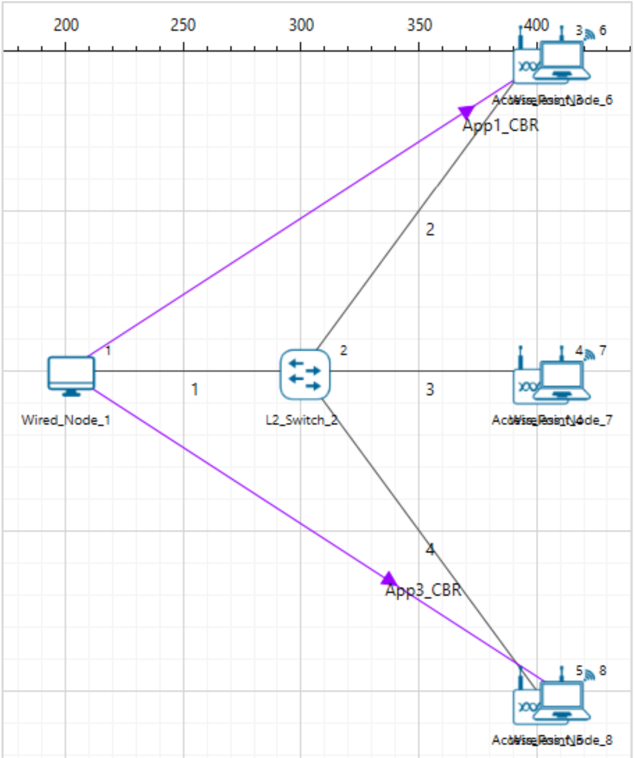


Figure 5-22: Network set up for studying the Multiple APs-Wi-Fi Networks with APs in same Channel No-Interfering

Step 3: The device positions are set according to the table given below Table **5-17**.

General Properties		
Device Name	**X / Lon**	**Y / Lat**
AP 3	400	0
AP 4	400	200
AP 5	400	400
Wireless Node 6	410	0
Wireless Node 7	410	200
Wireless Node 8	410	400

Table 5-17: Device positions for APs-STA – No Interfering

Interfering-II: The following changes in settings are done from the previous sample:

Step 1: From the previous sample, Add App2 CBR (i.e., from Wired Node1 to Wireless Node7), The distance between the Access Points (AP 1 and AP 3) is set to 400m and distance between APs and Wireless nodes as 10m as shown below Figure 5-23.

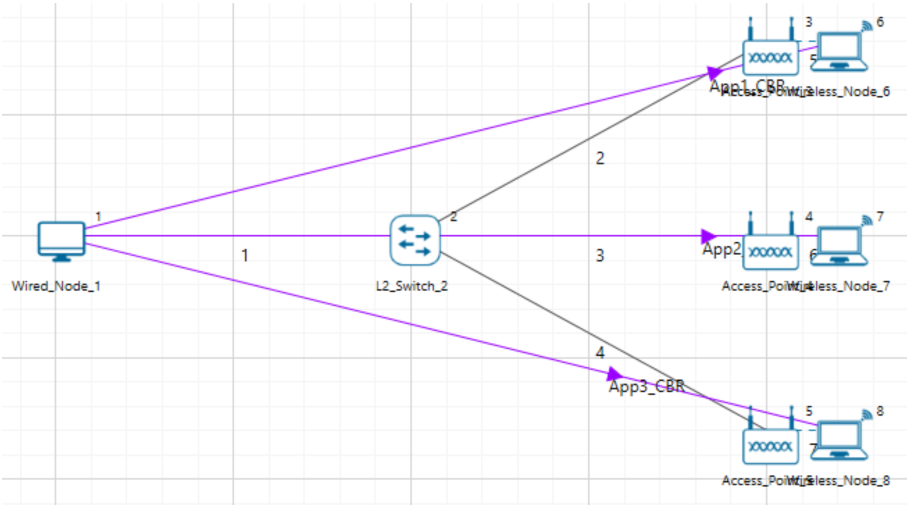


Figure 5-23: Network set up for studying the Multiple APs-Wi-Fi Networks with APs in same Channel Interfering-II

Step 2: The device positions are set according to the table given below Table **5-18**.

General Properties		
Device Name	**X / Lon**	**Y / Lat**
AP 3	400	120
AP 4	400	200
AP 5	400	280
Wireless Node 6	420	120
Wireless Node 7	420	200
Wireless Node 8	420	280

Table 5-18: Device positions for APs-STA - Interfering-II

Interfering-III: The following changes in settings are done from the previous sample:

Step 1: From the previous sample, we have removed App1 CBR (i.e., from Wired Node 1 to Wireless Node 6), set distance between the other 2 Access Points (AP 2 and AP 3) as 200m and distance between APs and Wireless nodes as 10m as shown below Figure 5-24.

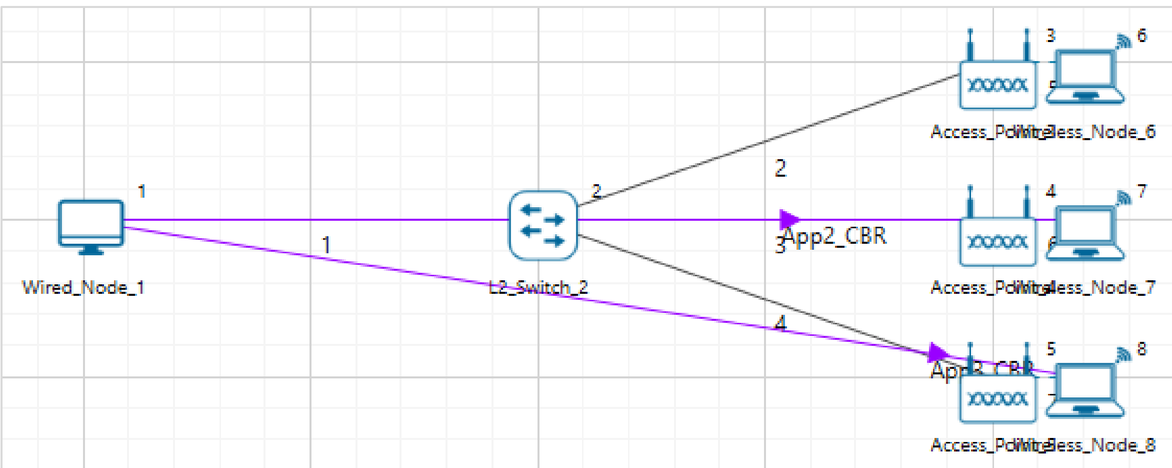


Figure 5-24: Network set up for studying the Multiple APs-Wi-Fi Networks with APs in same Channel Interfering - III

Step 2: The device positions are set according to the table given below Table 5-19.

General Properties		
Device Name	**X / Lon**	**Y / Lat**
AP 3	400	85
AP 4	400	135
AP 5	400	185
Wireless Node 6	420	85
Wireless Node 7	420	135
Wireless Node 8	420	185

Table 5-19: Device positions for APs-STA – Interfering III

No Interfering-I: The following changes in settings are done from the previous sample:

Step 1: From **Interfering-II**, we have removed first, and third applications as shown below Figure 5-25.

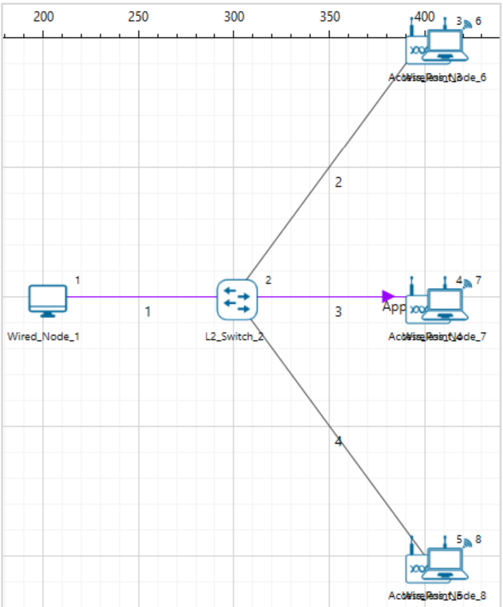


Figure 5-25: Network set up for studying the Multiple APs-Wi-Fi Networks with APs in same Channel No Interfering-I

Step 2: The device positions are set according to the table given below Table 5-20.

General Properties		
Device Name	**X / Lon**	**Y / Lat**
AP 3	400	0
AP 4	400	200
AP 5	400	400
Wireless Node 6	410	0
Wireless Node 7	410	200
Wireless Node 8	410	400

Table 5-20: Device positions for APs-STA - No Interfering-I

APs on the different channel: The following changes in settings are done from the previous sample:

Step 1: From previous sample, we have changed standard channel to **11_2462** under INTERFACE (WIRELESS) >PHYSICAL LAYER Properties of AP 2.

5.4.3 Output

After running simulation, check throughput in Application metrics as shown in the below screenshot Figure 5-26.

Application Metrics

End-to-end performance of applications running across the network.

Application ID	Application Name	Source ID	Destination ID	Throughput (Mbps	Delay (µs)	Jitter (µs)
1	App1_CBR	1	6	2.098896	3932013.621686	4397.871297
2	App2_CBR	1	7	2.069696	4005000.786608	4467.587679
3	App3_CBR	1	8	2.001952	3991927.103512	4664.399586

Figure 5-26: Application Metrics Table in Result window

Sample	Throughput (Mbps)		
	AP1	AP2	AP3
All APs in the same channel			
Interfering-I	2.09	2.06	2.00
No	5.94	N/A	5.92
Interfering-II	5.50	0.03	5.46
Interfering-III	N/A	3.08	3.05
No	5.92	N/A	N/A
Each AP in a different nonoverlapping channel			
Different	5.94	5.92	5.92

Table 5-21: Throughput for APs on the Same and different Channels

NOTE: Please refer “Wi-Fi UDP Download Throughput” experiment for theoretical WLAN throughput calculations in NetSim Experiment Manual.

5.4.4 Discussion

We recall that each AP is associated with one station (STA; e.g., a laptop). All the APs are connected to the same server which is sending separate UDP packet streams to each of the STAs via the corresponding AP. The packet transmission rate from the server is large enough so that the AP queue in permanently backlogged, i.e., the rate at which the server transmits packets is larger than the rate at which the AP can empty the packet queue.

5.4.4.1 All APs in the same channel

Interfering-I: All the APs and their associated STAs are close together, so that all devices (APs and STAs) can sense every other device.

- The table shows that all the AP-STA links achieve the same UDP throughput. This is because all the AP-STA links are equivalent (since all interfere with each other), and only one can be active at one time. The throughput for this scenario can be predicted from the analysis in Section 7.4 of the book Wireless Networking by Anurag Kumar, D. Manjunath and Joy Kuri.

No Interfering: AP1 and AP3 are close to their associated STAs but are 400m apart. The link from AP2 to its STA is half-way between the other two APs and is not carrying any traffic.

- The table shows that both the links from AP1 and AP3 to their respective STAs carry the same throughput, of 5.94Mbps and 5.92Mbps. These are also the throughputs that each link would have if the other was not present, indicating that the two links are far enough apart that they do not interfere.

Interfering-II: This is the same scenario as **No Interfering**, but the AP2-STA link is now carrying traffic.

- We find that, in comparison with **No Interfering**, the AP1-STA and AP3-STA carry slightly lower throughputs of about 5.21Mbps, whereas the AP2-STA link carries a small throughput of 0.92Mbps. Comparing **Interfering-I** and **II** we conclude that in these networks there can be severe unfairness depending on the relative placement of the AP-STA links. In **Interfering-I**, all the links could sense each other, and each got a fair chance. In **Interfering-II**, we have what is called the “link-in-the-middle problem.” The AP2-STA link is close enough to interfere with the AP1-STA link and the AP3-STA link, whereas the AP1-STA link and the AP3-STA link do not “see” each other. The AP2-STA link competes with the links on either side, whereas the other links compete only with the link in the center, which thereby gets suppressed in favour of the outer links.

Interfering-III: Here we stop the traffic to AP1 but send the traffic to the AP2-STA link and the AP3-STA link.

- The two active links interfere with each other, but the situation is symmetric between them (unlike in **Interfering-II**), and they obtain equal throughput. Again, the throughput obtained by these two links can be predicted by the analysis mentioned earlier in this section.

No Interfering-I: Now we send traffic only to AP2.

- The throughput is now 5.92Mbps, since the AP2-STA link can transmit without interference; there are no collisions. The reason that this throughput is less than the sum of the two throughputs in **Interfering-III** is that the single link acting by itself, with all the attendant overheads, is unable to occupy the channel fully.

5.4.4.2 Each AP in a different nonoverlapping channel

There is only one case here. Having observed the various situations that arose in the previous subsection when all the APs are in the same channel, now we consider the case where all the AP-STA pairs are each on a different nonoverlapping channel. As expected, every AP-STA pair gets the same throughput as when they are alone on the network.

5.5 Wi-Fi Multimedia Extension (IEEE 802.11 EDCA) (Level 3)

5.5.1 Introduction

In this experiment, we will study an enhancement to Wi-Fi that enables APs and STAs to handle various packet flows in such a way so as to provide differentiated quality of service (QoS) to these flows. In the original Wi-Fi standard (IEEE 802.11 DCF), every device in the network has one output buffer (queue) for all packets to be transmitted onto the wireless channel. The consequence of this would be that a packet stream with strict delivery constraints and another will relatively loose delivery objectives are queued in the same output buffer at every device. Each such queue is scheduled by the DCF CSMA/CA (see the experiment "Wi-Fi: UDP Download Throughput"), and when a queue gets its transmission opportunity the first (head-of-the-line (HOL)) packet is transmitted. This might result in packets with strict delivery constraints being kept waiting, while other, less urgent, packets get transmitted.

For example, an interactive voice call might have 200-byte packets being transmitted periodically, with a period of 20 ms. Ideally, for perfect voice playout at the receiver, this voice stream must arrive exactly as it was transmitted: every 200-byte packet must arrive, and with the gaps of 20 ms intact. If the voice packets are delayed excessively, or if the delay is highly variable, the playout is affected, and voice quality (and speaker interaction) is affected. On the other hand, TCP controlled file transfers can adapt to network delay and delay variability. Evidently, the solution is to create multiple output buffers in each device, of different transmit priorities, queue the more urgent packets in higher priority buffers, and create a mechanism for preferential transmission of the packets with tighter QoS requirements.

In this experiment we will study the EDCAF mechanism, an extension to DCF, which implements service differentiation in Wi-Fi.

5.5.2 EDCAF: Access Categories

In the year 2005, the standard IEEE 802.11e (EDCAF-Enhanced Distributed Channel Access Function) was introduced, with the above issues in mind. In EDCAF, there are four Access Categories (AC0, AC1, AC2, and AC3), with AC3 being the highest priority and AC0 being the lowest. The assignment of application usage to these ACs was.

Access Category	Application	Example
AC0	Background	Background print job
AC1	Best Effort	TCP File Transfer
AC2	Video	Video Conference
AC3	Voice	Video Conference

Table 5-22: EDCA 4 access categories for 802.11 both AP and STA

Now, if a device (an AP or a STA) is sending interactive voice packets then these packets can be queued in the AC3 buffer of the device, whereas packets of a simultaneous TCP file transfer can be queued in the AC1 buffer. The human brain is less sensitive to video packet losses, and delays in the rendering of video frames (than the human hearing system is of voice corruption), hence video is given a priority between voice and TCP. The lowest category, AC0, can be used for any application whose packets need to just be delivered, without any well-defined quality of service, for example, a low urgency bulk printing service.

Having created buffers into which the various priority packets are queued, a mechanism is needed to schedule transmissions from these buffers so that service differentiation is achieved. Ideally, strict priority service could be enforced, i.e., assuming that there is only AC3 and AC2 traffic in the network, if any device has a nonempty AC3 buffer, all packets from AC3 category should be served before any AC2 traffic is served. Further, ideally, the video and the TCP file transfers could have been assigned a guaranteed service rate, to meet their QoS requirements. Such strict priorities and guaranteed service would belong to the concept of Integrated Services. However, the IEEE 802.11 wireless access mechanism is distributed, and there is no central entity that has the instantaneous state of all the buffers in all the devices. Hence, strict priority or a guaranteed service rate is not possible to. Instead, the IEEE 802.11 series of standards adopted EDCAF (an extension to DCF) for scheduling the service of the access category queues at the contending devices. The EDCAF mechanism achieves Differentiated Services. How does the MAC layer in a device know which access category buffer to queue a packet in? This is achieved by the corresponding application using the DSCP (Differentiated Service Code Point) field in the IPV4 header to indicate the Differentiated Services class of the packet. The MAC layer of the device would have a table that maps the DSCP value to the access category.

5.5.3 EDCAF: Service Differentiation Mechanisms

We begin by recalling how the basic EDCF works, since EDCAF is built as an extension of EDCF. In EDCF, after handling the HOL packet in its (single) buffer (which could result in the packet being transmitted successfully or discarded (due to exceeding the maximum number of reattempts)), a device waits for DIFS, samples an initial back-off for the next packet in the buffer, and begins to count down (at slot boundaries) until the back-off counter reaches zero, at which instant the first attempt for the next packet is made. A collision leads to a new back-off being sampled from a distribution with a larger mean. All nodes behave in exactly the same manner, thus getting opportunities to transmit packets whenever their back-off counters reach zero. Thus, all devices (STAs and APs) have the same behavior (statistically), and there is no service differentiation.

Now consider an AP with AC3 packets to be transmitted (say, voice), and an STA with AC1 packets (say, TCP). After the AP transmits a voice packet, in EDCAF, the AP's MAC waits for AIFS3 (Arbitration Inter Frame Space for Category 3) which is 2 slots, and samples a back-off from a uniform distribution over 1 slot to 2^7slots. On the other hand, at this point the STA waits for AIFS1, which is 3 slots. In addition, after a TCP packet has been transmitted (or dropped) the STA samples a back-off for the next packet from a uniform distribution over 1 slot to 2^{31}slots. Thus, the HOL packet waiting in the AC3 buffer has two advantages over the HOL packet waiting in the AC1 buffer:

i. The back-off counter of the AC3 category starts counting down one slot earlier than the AC1 category.
ii. The back-off counters are smaller for AC3 than for AC1.

These two mechanisms conspire to differentiate the wireless access service in favour of AC3. Note that we do not get strict priority. For example, if a voice packet has been transmitted by the AP, and after AIFS3, the back-off sampled is 3 slots, whereas the residual back-off of the TCP transfer (at the STA) was 2 slots, then the TCP packet will be transmitted next. However, the service differentiation is significant as the simulation results from NetSim will demonstrate later in this chapter. The following is the table of all EDCAF parameters as specified by the standard.

Access Category	CWmin	CWmax	AIFSN	Max TXOP (μs)
Background (AC_BK)	31	1023	7	3264
Best Effort (AC_BE)	31	1023	3	3264
Video (AC_VI)	15	31	2	6016
Voice (AC_VO)	7	15	2	3264

Table 5-23: EDCA access parameters for 802.11 b for both AP and STA

5.5.4 The Experimental Plan

- Voice over DCF: We first want to understand the limitation when carrying interactive voice over DCF.
    - With this in mind, we will set up several full-duplex voice calls between several STAs and the wired network, one such call for each STA. Each full-duplex voice call will be modelled by a periodic stream of 200-byte UDP packets (160B voice plus 40B of UDP/IP headers), generated at 20 ms intervals, from the STA to the wired network, and another such, independent stream from the wired network to the STA. We will increase the number of STAs, thereby increasing the number full-duplex voice calls, and will determine the number of calls that can be handled.
    - Then we will add on TCP controlled file transfer from the wired network to another STA. Due to reasons explained earlier in this chapter, the voice performance should degrade, leading to fewer calls being possible to handle along with the TCP transfer.

- Voice over EDCAF: Next we repeat the above two experiments with the EDCAF mechanism enabled. We should find that it is possible to maintain a substantial number of voice conversations even while running the TCP file transfer. Next we will study what happens if the number of TCP file transfers is increased, the question being whether the number of voice conversations that can be handled gets affected.

5.5.5 Simulation Experiments to Study IEEE 802.11 EDCAF

Open NetSim and click on **Experiments> Internetworks> Wi-Fi> Wi Fi WME 802.11e QoS EDCA>DCF Voice Only** then click on the tile in the middle panel to load the example as shown in below Figure 5-27.

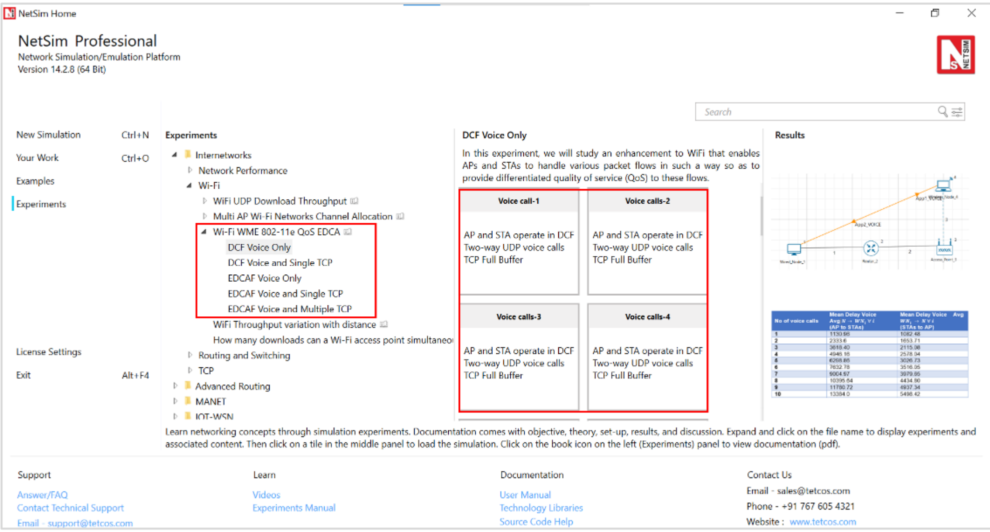


Figure 5-27: List of scenarios for the example of Wi Fi WME 802.11e QoS EDCA

Case 1: DCF with full-duplex voice calls only

Network Scenario

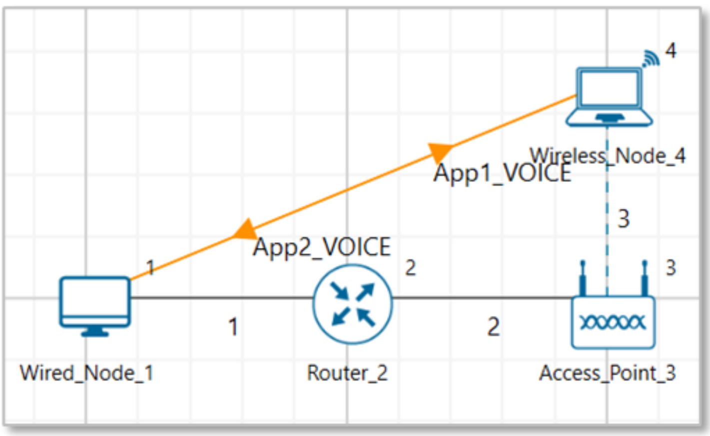


Figure 5-28: Network set up for studying the DCF with full-duplex voice calls only

Network Setup

- AP and STA operate in DCF.

- There is no error in all wired and wireless links.

Applications

- Two-way UDP voice calls from Node to Wireless Node i, with i being incremented. Each voice calls in NetSim is modelled as 2 one-way voice applications. The voice modelling option in NetSim UI currently allows transfer in one direction only. Hence, we model a two-way voice call as 2 one-way voice applications.
  - $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i, while N represents the wired node or remote host.
  - Each voice call runs G.711 at 64 Kbps.

Case 2: DCF with full-duplex voice calls and a single TCP download

Network Scenario

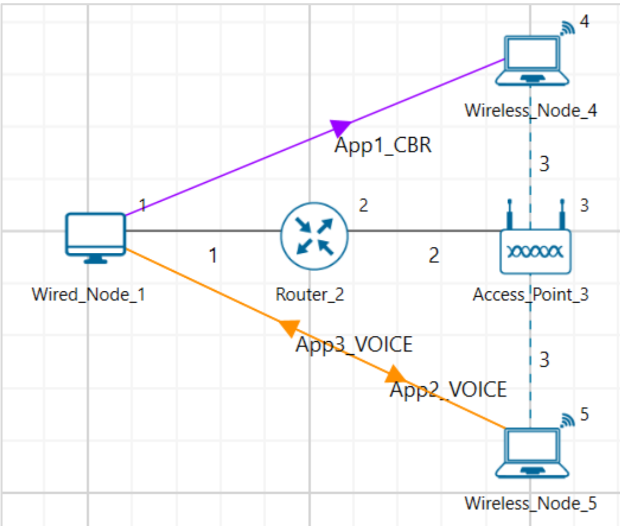


Figure 5-29: Network set up for studying the DCF with full-duplex voice calls and a single TCP download

Network Setup

- AP and STA operate in DCF.
- There is no error in all wired and wireless links.

Applications

- TCP full buffer (or saturated case) download from N to WN_1. The saturation is generated by using a CBR application with Packet Size of 1460 Bytes and Inter packet arrival time of 1000 μs. Since this generation rate is higher than the Wi-fi link it rate as a saturated (full buffer) TCP flow is achieved. The reason for emulating a TCP download using a CBR session, is that a TCP file download would take a longer time to simulate.

- Voice calls from N to WN_{i+1} with i being incremented. Two-way UDP voice calls from Node to Wireless_Node_i. Each voice calls in NetSim is modelled as 2 one-way voice applications
  - $WN_{i+1} \rightarrow N$ and $N \rightarrow WN_{i+1}$. WN_i represent wireless node i, while N represents the wired node or remote host.
  - Each voice call runs G.711 at 64 Kbps.

Case 3: EDCAF with full-duplex voice calls only

Network Scenario

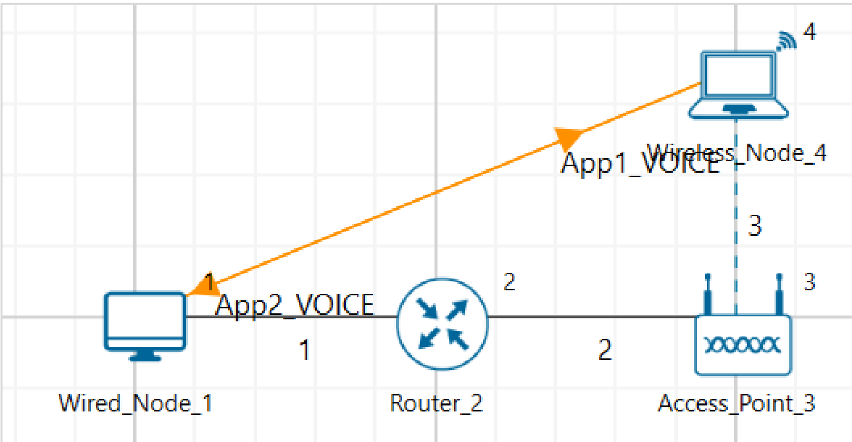


Figure 5-30: Network set up for studying the EDCAF with full-duplex voice calls only

Network Setup

- AP and STA operate in EDCAF, with EDCAF parameters set per reference paper
- There is no error in all wired and wireless links.

Applications

- Two-way UDP voice calls from Node to Wireless Node i, with i being incremented. Each voice calls in NetSim is modelled as 2 one-way voice applications. The voice modelling option in NetSim UI currently allows transfer in one direction only. Hence, we model a two-way voice call as 2 one-way voice applications.
  - $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i, while N represents the wired node or remote host.
  - Each voice call runs G.711 at 64 Kbps.

Case 4: EDCAF with full-duplex voice calls and a single TCP download

Network Scenario

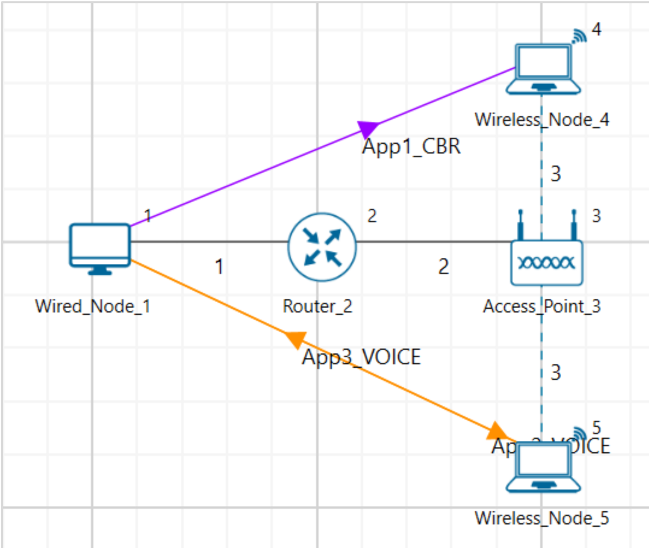


Figure 5-31: Network set up for studying the EDCAF with full-duplex voice calls and a single TCP download.

Network Setup

- AP and STA operate in EDCAF, with EDCAF parameters set per reference paper. In the AP, TCP would be queued in AC_BE while Voice packets would be queued in AC_VO.
- There is no error in all wired and wireless links.

Applications

- TCP full buffer (or saturated case) download from N to WN_1. The saturation is generated by using a CBR application with Packet Size of 1460 Bytes and Inter packet arrival time of 1000 μs. Since this generation rate is higher than the Wi-fi link it rate as a saturated (full buffer) TCP flow is achieved.
- Voice calls from N to WN_{i+1} with i being incremented. Two-way UDP voice calls from Node to Wireless Node i. Each voice calls in NetSim is modelled as 2 one-way voice applications
  - $WN_{i+1} \rightarrow N$ and $N \rightarrow WN_{i+1}$. WN_i represent wireless node i, while N represents the wired node or remote host.
  - Each voice call runs G.711 at 64 Kbps.

Case 5: EDCAF with full-duplex voice calls and multiple TCP downloads

Network Scenario

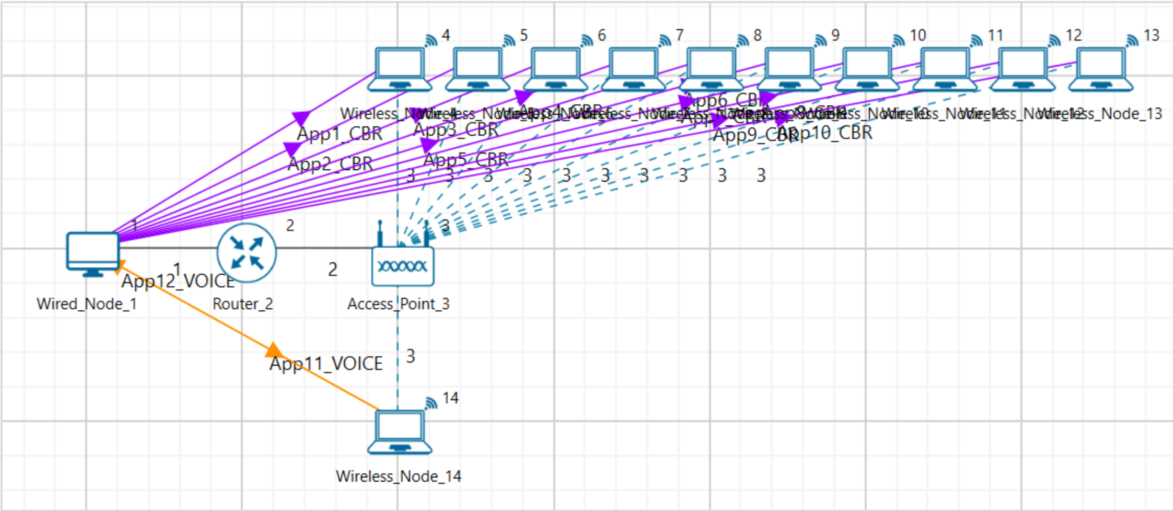


Figure 5-32: Network set up for studying the EDCAF with full-duplex voice calls and multiple TCP downloads

Network Setup

- AP and STA operate in EDCAF mode.
- There is no error in all wired and wireless links.

Applications

- 10 TCP downloads from N to WN_1 through WN_{11}
  - TCP full buffer (or saturated case) download from $N \rightarrow WN$. The saturation is generated by using a CBR application with Packet Size of 1460 B and Inter packet arrival time of 1000 μs. Since this generation rate is higher than the Wi-fi link it rate as a saturated (full buffer) TCP flow is achieved.
- UDP Voice calls from N to WN_{11+i} with i being incremented.
- Each voice calls in NetSim is modelled as 2 one-way voice applications
  - $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i, while N represents the wired node or remote host.
  - Each G.711 at 64 Kbps.

5.5.6 Simulation Results

Case 1: DCF with full-duplex voice calls only

- Tabulated separately for applications going $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i, while N represents the wired node or remote host.
- A mean delay of 20,000 μs is considered as a threshold.
- For the case $N \rightarrow WN_i$ this threshold is crossed at 11 calls
- For the case $WN_i \rightarrow N$ this threshold is crossed at 20 calls

No of voice calls	Mean Delay Voice	Mean Delay
1	1130.97	1082.48
2	2333.61	1653.72
3	3618.41	2115.09
4	4946.16	2578.05
5	6298.87	3026.74
6	7632.79	3516.05
7	9004.97	3979.66
8	10395.65	4434.80
9	11780.73	4937.34
10	13384	5498.42
11	3569361	6354
12	11267144	6985
13	17921294	7568
14	23235201	8232
15	28389497	9221
16	32526177	10326
17	36062048	12060.73
18	39589934	14635.82
19	42859790	21222.59
20	44960901	39404.71
21	45684937.21	210834.19
22	46312147.7	2102284.78

Table 5-24: DCF with full-duplex voice calls only. $\boldsymbol{N} \rightarrow \boldsymbol{WN_i}$ and $\boldsymbol{WN_i} \rightarrow \boldsymbol{N}$, where N represents the wired node and $\boldsymbol{WN_i}$ represents the $\boldsymbol{i}$-th wireless node. Therefore $\boldsymbol{N} \rightarrow \boldsymbol{WN_i}$ represents the flows from AP to STAs while $\boldsymbol{WN_i} \rightarrow \boldsymbol{N}$ represents the flows from STAs to AP.

* Mean Delay Voice $WN_i \rightarrow N$ = Average of the delay of all the applications flowing $WN_i \rightarrow N$

Case 2: DCF with full-duplex voice calls and single TCP download

- Tabulated separately for applications going $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i, while N represents the wired node or remote host.
- A mean delay of 20,000 μs is considered as a threshold.
- For the case $N \rightarrow WN_i$ this threshold is crossed at 1 call itself

No of TCP Connections	No of voice calls	Mean Delay Voice $N \rightarrow WN_i$	Mean Delay Voice $WN_i \rightarrow N$
1	1	106287.82	3313.02
	2	139670.90	3574.49

Table 5-25: DCF with full-duplex voice calls and single TCP download $\boldsymbol{N} \rightarrow \boldsymbol{WN_i}$ and $\boldsymbol{WN_i} \rightarrow \boldsymbol{N}$

Case 3: EDCAF with full-duplex voice calls

- Tabulated separately for applications going $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i, while N represents the wired node or remote host.
- A mean delay of 20,000 μs is considered as a threshold.

- For the case $N \rightarrow WN_i$ this threshold is crossed at 13 calls, for the case $WN_i \rightarrow N$ this threshold is crossed at 21 calls

No of voice calls	Mean Delay Voice $N \rightarrow WN_i$	Mean Delay Voice $WN_i \rightarrow N$
1	1000.70	704.67
2	1720.33	1575.46
3	2460.84	2464.80
4	3231.19	3356.52
5	4084.78	4274.85
6	5092.33	5068.09
7	6090.03	5872.13
8	7083.65	6648.27
9	8193.41	7373.58
10	9226.28	8115.11
11	10508.87	8671.12
12	11697.43	9477.25
13	473970.92	12243.88
14	509835	12656.72
15	513948.31	13302.09
16	512255.59	13890.68
17	507689	14612
18	502026	15251
19	498552	16611
20	493342	17863
21	488076.26	20317.288

Table 5-26: EDCAF with full-duplex voice calls $\boldsymbol{N \rightarrow WN_i}$ and $\boldsymbol{WN_i \rightarrow N}$

Case 4: EDCAF with full-duplex voice calls and single TCP download

- Tabulated separately for applications going $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i, while N represents the wired node or remote host.
- A mean delay of 20,000 μs is considered as a threshold.
- For the case $N \rightarrow WN_i$ this threshold is crossed at 13 calls

No of TCP Connections	No of Voice calls	Mean Delay Voice $N \rightarrow WN_i$	Mean Delay Voice $WN_i \rightarrow N$
1	1	2669.87	2969.59
	2	3742.81	4204.35
	3	4653.75	5087.94
	4	5599.46	5954.40
	5	6631.17	6799.18
	6	7877.11	7551.42
	7	9081.65	8221.35
	8	10353.74	8932.09
	9	11761.94	9617.73
	10	13781.53	10379.25

	11	16629.97	10723.95
	12	28475.08	9966.17
	13	489968.54	12295.56
	14	512965.29	12765.90

Table 5-27: EDCAF with full-duplex voice calls and single TCP download TCP download $\boldsymbol{N \rightarrow WN_i}$ and $\boldsymbol{WN_i \rightarrow N}$

Case 5: EDCAF with full-duplex voice calls and multiple TCP downloads

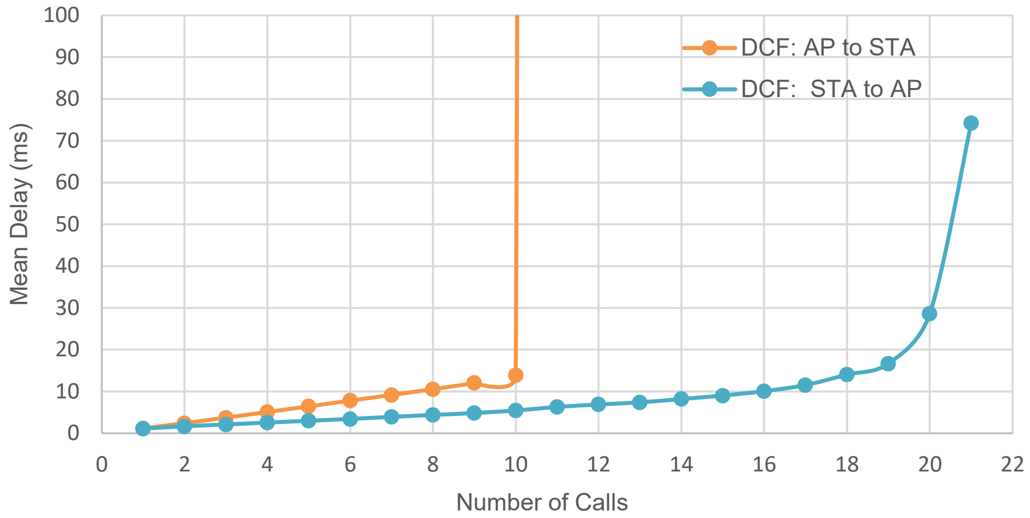
- Tabulated separately for applications going $WN_i \rightarrow N$ and $N \rightarrow WN_i$. WN_i represent wireless node i, while N represents the wired node or remote host.
- A mean delay of 20,000 μs is considered as a threshold.
- For the case $N \rightarrow WN_i$ this threshold is crossed at 13 calls

No of TCP Connections	No of voice calls	Mean Delay Voice $N \rightarrow WN_i$	Mean delay Voice $WN_i \rightarrow N$
10	1	2631.65	2922.68
	2	3640.44	4079.23
	3	4624.32	5079.30
	4	5386.83	5838.45
	5	6447.83	6602.43
	6	7186.98	7472.08
	7	8801.21	8097.06
	8	10052.13	8645.98
	9	11660.75	9466.26
	10	13281.74	9931.90
	11	16146.43	10759.76
	12	43567.50	11347.94
	13	506122.33	12439.36
	14	511198.25	12980.59

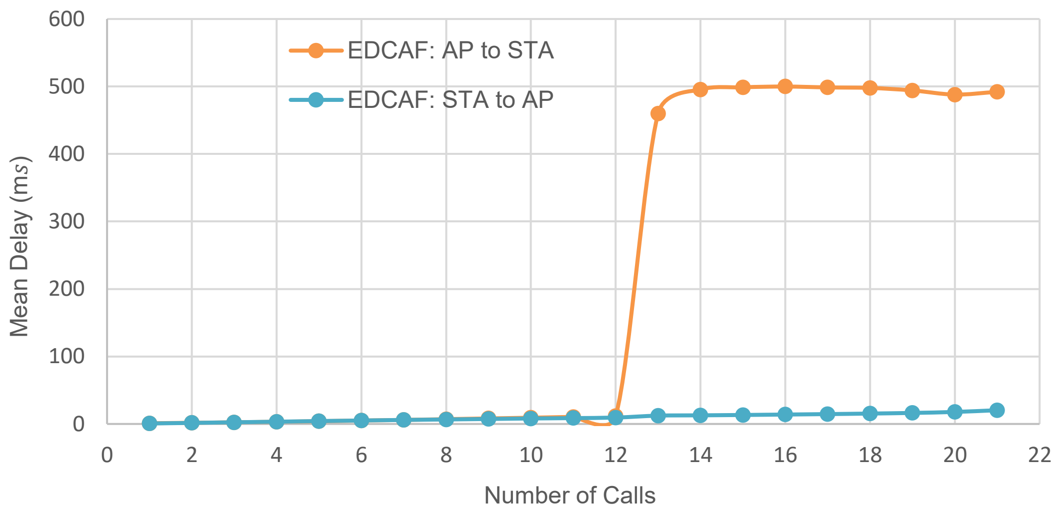
Table 5-28: EDCAF with full-duplex voice calls and multiple TCP downloads

$\boldsymbol{N \rightarrow WN_i}$ and $\boldsymbol{WN_i \rightarrow N}$

Comparison Charts



(a)



(b)

Figure 5-33: The plot of Mean Delay Vs. Number of Calls for DCF and EDCAF (a & b)

5.5.7 Discussion of the Simulation Results

5.5.7.1 Results for DCF

1. Observations

- Only voice calls: With one voice conversation, the mean packet delay for the wired-to-wireless (downlink) direction (i.e., these packets queue in the AP) is 1.18161 ms, whereas in the wireless-to-wired (uplink) direction (i.e., these packets queue in the STA) the mean packet delay is 1.11924 ms (we will, henceforth, report these delays in milliseconds and round-off to two decimal places). These mean delays increase as the number of voice conversations increase. We notice that with 10 conversations the downlink mean packet delay is 13.69 ms, whereas the uplink packet delay is 5.47 ms. An increase of one more call result in the downlink mean packet delay becoming 4027.85 ms and the uplink mean packet delay being 6.28 ms.

- One TCP download to one STA from a wired node and increasing number of full-duplex voice calls: With just one voice call the mean delay is 116.00 ms in the downlink and 3.18 ms in the uplink. These values should be compared with 1.18 ms and 1.12 ms, respectively. Thus, with just one TCP connection, a single voice call experiences substantially larger mean delay.

2. Discussion

- With increasing number of voice calls (without any simultaneous TCP download) the dramatic change in the downlink delay, when the number of voice calls is increased from 10 to 11 is due to the downlink queue becoming unstable, i.e., the arrival rate of bits exceeds the rate at which the DCF wireless access mechanism can service the bits queued at the AP. The sudden transition from low delays to very high delays is typical of a queue going a stable regime to an unstable regime.
- It is interesting to note that at this transition point, the uplink delays have not increased as dramatically. In fact, in the uplink direction the transition from stability to instability appears in going from 22 calls to 23 calls. This difference in the downlink and uplink directions is because all the downlink voice packet streams are handled at one queue (the AP's Wi-Fi buffer), with one contention process, whereas each uplink voice packet stream has its own buffer with its own contention process. If all the uplink voice streams had also been from one STA then the same phenomenon would have been observed in both directions.
- Next, we saw that with a single downlink TCP transfer the downlink mean delay of a single voice call is almost 100 times that without TCP. This occurs because the TCP transfer over a the local area network builds up a large window, most of which resides in the AP buffer. The TCP file transfer packets are large (about 1500 bytes). A single voice stream generates 200-byte packets at 20 ms intervals. The downlink voice packets see a very large buffer, due to the TCP packets queued in the AP buffer. It may be noted here, that with this kind of delay, even a single interactive voice call will not be supported.

Results for EDCAF

1. Observations

- With voice calls alone the transition in downlink delay occurs in going from 12 to 13 calls.
- With TCP downloads (1 or 10 downloads) the transition in downlink voice packet delay does not change as compared to without TCP

2. Discussion

- EDCAF creates different buffers for voice and for TCP file transfers (AC3 and AC1, respectively). The service differentiation mechanism between these buffers is described earlier in this chapter. The experimental results show that voices call performance is not seriously affected by the TCP controlled file transfers.
- As before, and for the same reasons, the voice capacity is limited by the service rendered to the AP buffers.

6 Internet of Things (IOT) and Wireless Sensor Networks

6.1 One Hop IoT Network over IEEE 802.15.4 (Level 2)

6.1.1 Introduction

In most situations, due to the practical difficulty of laying copper or optical cables to connect sensors and actuators, digital wireless communication has to be used. In such applications, since the energy available in the sensor and actuator devices is small, there is a need for keeping costs low, and the communication performance requirement (in terms of throughput and delay is limited) several wireless technologies have been developed, with the IEEE 802.15.4 standard being one of the early ones.

The IEEE Standard 802.15.4 defines PHY and MAC protocols for low-data-rate, low-power, and low-complexity short-range radio frequency (RF) digital transmissions.

In this experiment, we will study the simplest IEEE 802.15.4 network with one wireless node transmitting packets to an IEEE 802.15.4 receiver, from where the packets are carried over a high-speed wireline network to a compute server (where the sensor data would be analyzed).

6.1.2 The IEEE 802.15.4 PHY and MAC

We will study the IEEE 802.15.4 standard that works in the 2.4 GHz ISM band, in which there is an 80 MHz band on which 16 channels are defined, each of 2 MHz, with a channel separation of 5 MHz. Each IEEE 802.15.4 network works in one of these 2 MHz channels, utilizing spread spectrum communication over a chip-stream of 2 million chips per second. In this chip-stream, 32 successive chips constitute one symbol, thereby yielding 62,500 symbols per second (62.5 Ksps; $\frac{(2\times10^6)}{32} = 62{,}500$). Here, we observe that a symbol duration is $32 \times \frac{1}{2\times10^6} = 16\ \mu$sec. Binary signaling (OQPSK) is used over the chips, yielding 2^{32} possible sequences over a 32 chip symbol. Of these sequences, 16 are selected to encode 4 bits $(2^4 = 16)$. The sequences are selected so as to increase the probability of decoding in spite of symbol error. Thus, with 62.5 Ksps and 4 bits per symbol, the IEEE 802.15.4 PHY provides a raw bit rate of $62.5 \times 4 = 250$ Kbps.

Having described the IEEE 802.15.4 PHY, we now turn to the MAC, i.e., the protocol for sharing the bit rate of an IEEE 802.15.4 shared digital link. A version of the CSMA/CA mechanism is used for multiple access. When a node has a data packet to send, it initiates a random back-off with the first back-off period being sampled uniformly from 0 to $(2^{macminBE} - 1)$, where $macminBE$ is a standard parameter. The back-off period is in slots, where a slot equals 20 symbol times, or $20 \times 16 = 320\ \mu$sec. The node then performs a *Clear Channel Assessment* (CCA) to determine whether the channel is idle. A CCA essentially involves the node listening over 8 symbols times and integrating its received power. If the result exceeds a threshold, it is concluded that the channel is busy and CCA fails. If the CCA succeeds, the node does a Rx-to-Tx turn-around, which takes 12 symbol times and starts transmitting on the channel. The failure of the CCA starts a new back-off process with the back-off exponent raised by one, i.e., to macminBE+1, provided it is less than the maximum back-off value, macmaxBE. The maximum number of successive CCA failures for the same packet is governed by macMaxCSMABackoffs; if this limit is exceeded the packet is discarded at the MAC layer. The standard allows the inclusion of acknowledgements (ACKs) which are sent by the intended receivers on a successful packet reception. Once the packet is received, the receiver performs a Rx-to-Tx turnaround, which is again 12 symbol times, and sends a 22-symbol fixed size ACK packet. A successful transmission is followed by an InterFrameSpace (IFS) before sending another packet.

The IEEE 802.15.4 can operate either in a beacon enabled or a nonbeacon enabled mode. In the beacon enabled mode, the PAN coordinator works with time slots defined through a superframe structure (see Figure 6-1). This permits a synchronous operation of the network. Each superframe has active and inactive portions. The PAN Coordinator interacts with the network only during the active portion. The active portion is composed of three parts: a beacon, a contention access period (CAP), and a contention free period (CFP). The active portion starts with the transmission of a beacon and a CAP commences immediately after the beacon. All frames, except acknowledgment frames and any data frame that immediately follows the acknowledgment of a data request command (as would happen following a data request from a node to the PAN coordinator), transmitted in the CAP, must use a slotted CSMA/CA mechanism to access the channel.

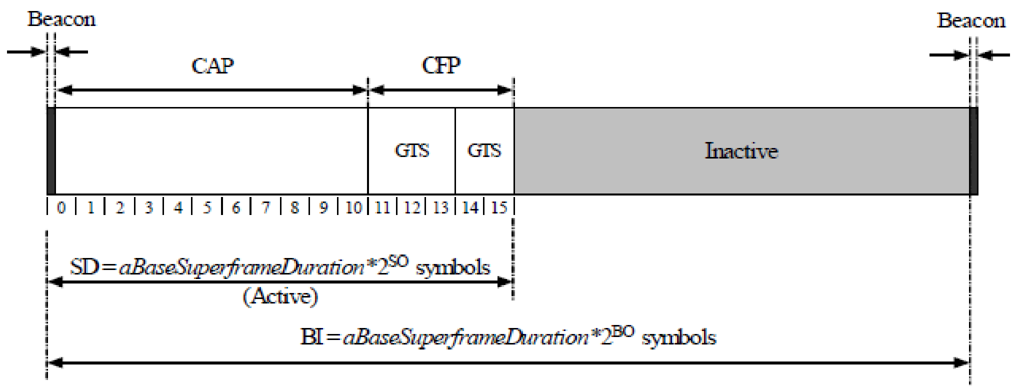


Figure 6-1: The IEEE 802.15.4 Frame Structure

When a transmitted packet collides or is corrupted by the PHY layer noise, the ACK packet is not generated which the transmitter interprets as packet delivery failure. The node reattempts the same packet for a maximum of a Max FrameRetries times before discarding it at the MAC layer. After transmitting a packet, the node turns to Rx-mode and waits for the ACK. The macAckWaitDuration determines the maximum amount of time a node must wait to receive the ACK before declaring that the packet (or the ACK) has collided. The default values of macminBE, macmaxBE, macMaxCSMABackoffs, and a Max FrameRetries are 3, 5, 4, and 3.

6.1.3 Objectives of the Experiment

In Section 6.1.2, we saw that the IEEE 802.15.4 PHY provides a bit rate of 250 Kbps, which has to be shared among the nodes sharing the 2 MHz channel on which this network runs. In the simulation experiment, the packets will have an effective length of 109 bytes (109 B = 872 bits). Thus, over a 250 Kbps link, the maximum packet transmission rate is $\frac{250\times1000}{872} = 286.70$ packets per second. We notice, however, from the protocol description in Section 6.1.2, that due to the medium access control, before each packet is transmitted the nodes must contend for the transmission opportunity. This will reduce the actual packet transmission rate well below 286.7.

In this experiment, just one node will send packets to a receiver. Since there is no contention (there being only one transmitter) there is no need for medium access control, and packets could be sent back-to-back. However, the MAC protocol is always present, even with one node, and we would like to study the maximum possible rate at which a node can send back-to-back packets, when it is the only transmitter in the network. Evidently, since there is no uncertainty due to contention from other nodes, the overhead between the packets can be calculated from the protocol description in Section 6.1.2. This has been done in Section 6.1.6.

This analysis will provide the maximum possible rate at which a node can send packets over the IEEE 802.15.4 channel. Then in Section 6.1.7, we compare the throughput obtained from the simulation with that obtained from the analysis. In the simulation, in order to ensure that the node sends at the maximum possible rate, the packet queue at the transmitting node never empties out. This is ensured by inserting packets into the transmitting node queue at a rate higher than the node can possibly transmit.

6.1.4 NetSim Simulation Setup

Open NetSim and click on **Experiments>IOT-WSN> One Hop IoT Network over IEEE 802.15.4** click on the tile in the middle panel to load the example as shown in below Figure 6-2.

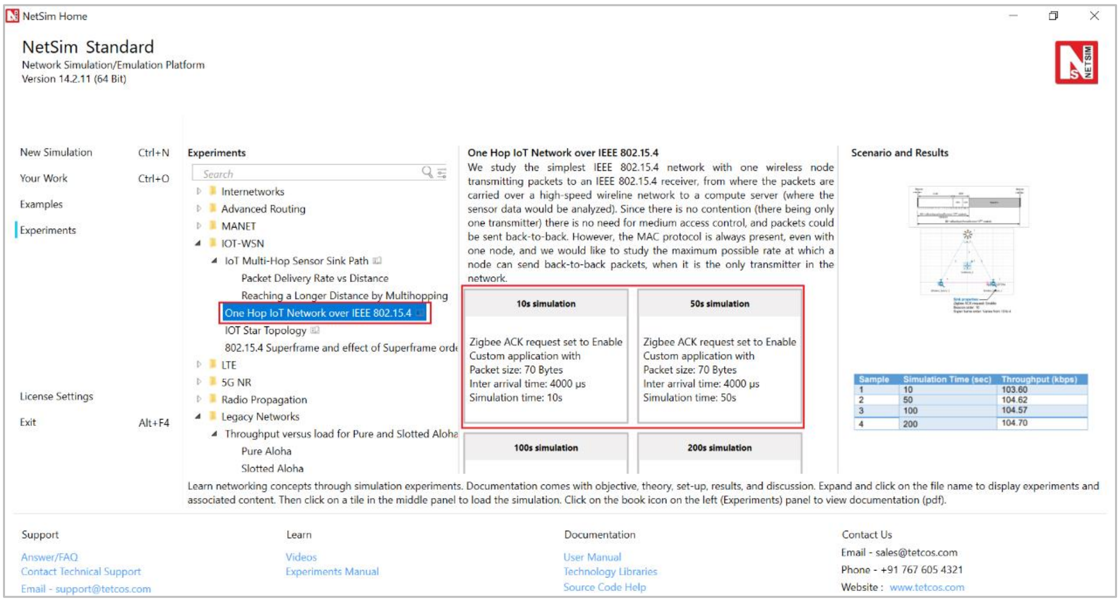


Figure 6-2: List of scenarios for the example of One Hop IoT Network over IEEE 802.15.4

10s Simulation sample

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 6-3.

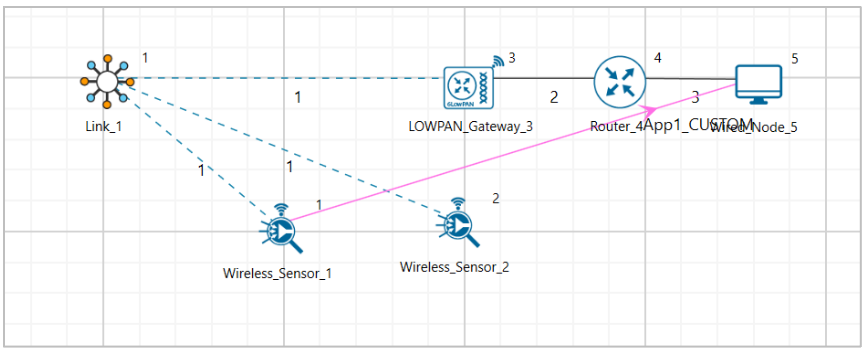


Figure 6-3: Network set up for studying the One Hop IoT Network over IEEE 802.15.4

6.1.5 Simulation Procedure

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

Step 1: A network scenario is designed in NetSim GUI comprising of 2 Wireless Sensors, a LOWPAN Gateway, 1 Router, and 1 Wired Node from IoT library.

Step 2: In the Interface Zigbee > Data Link Layer of Wireless Sensor 1, Ack Request is set to Enable, and Max Frame Retries is set to 7. These settings will be automatically applied to Wireless Sensor 2 since the parameters are global. To configure these settings, click on the wireless sensor, expand the property panel on the right, and set the parameters as mentioned.

Step 3: In the Interface Zigbee > Data Link Layer of LOWPAN Gateway, **Beacon Mode** is set to Disable by default.

Step 4: Click on Ad hoc link and expand the link property panel on right and set channel characteristics to **No pathloss**.

Step 5: Configure Custom application between Sensor 1 to Wired node 5 by clicking on set traffic tab from ribbon on the top. Click on the created application, expand the application property panel on the right, and set the transport protocol to UDP, the packet size to 70 bytes, and the inter arrival time to 4000 µs.

The Packet Size and Inter Arrival Time parameters are set such that the Generation Rate equals 140 Kbps. Generation Rate can be calculated using the formula:

$$Generation\ Rate\ (Mbps)\ =\ Packet\ Size\ (Bytes)\ *\ 8/Interarrival\ time\ (\mu s)$$

NOTE: If the size of the packet at the Physical layer is greater than 127 bytes, the packet gets fragmented. Taking into account the various overheads added at different layers (which are mentioned below), the packet size at the application layer should be less than 80 bytes.

Step 6: Run simulation for 10 Seconds and note down the throughput.

Similarly, do the other samples by increasing the simulation time to 50, 100, and 200 Seconds respectively and note down the throughputs.

6.1.6 Analysis of Maximum Throughput

We have set the Application layer payload as 70 bytes in the Packet Size and when the packet reaches the Physical Layer, various other headers get added like Table 6-1.

App layer Payload	70 bytes
Transport Layer Header	8 bytes
Network Layer Header	20 bytes
MAC Header	5 bytes
PHY Header (includes Preamble, and Start Packet Delimiter)	6 bytes

Packet Size	109 bytes

Table 6-1: Overheads added to a packet as it flows down the network stack

By default, NetSim uses Unslotted CSMA/CA and so, the packet transmission happens after a Random Back Off, CCA, and Turn-Around-Time and is followed by Turn-Around-Time and ACK Packet and each of them occupies specific time set by the IEEE 802.15.4 standard as per the timing diagram shown below Figure 6-4.

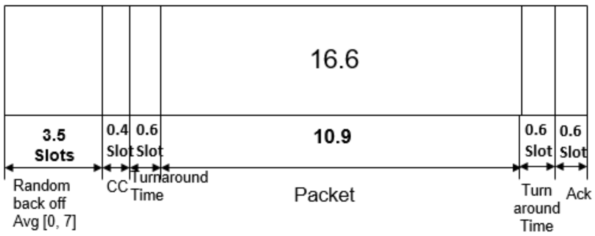


Figure 6-4: Zigbee timing diagram

From IEEE standard, each slot has 20 Symbols in it and each symbol takes 16µs for transmission.

Symbol Time	T_s	16 µs
Slot Time	$20 \times T_s$	0.32 ms
Random Backoff Average	$3.5{*}Slots$	1.12 ms
CCA	$0.4{*}Slots$	0.128 ms
Turn-around-Time	$0.6{*}Slots$	0.192 ms
Packet Transmission Time	$10.9{*}Slots$	3.488 ms
Turn-around-Time	$0.6{*}Slots$	0.192 ms
ACK Packet Time	$0.6{*}Slots$	0.192 ms
Total Time	$\mathbf{16.6{*}Slots}$	**5.312 ms**

Table 6-2: Symbol times as per IEEE standard

$$Analytical\ Application\ Throughput = \frac{70(bytes)inApplayer * 8}{5.312\ ms} = 105.42\ kbps$$

6.1.7 Comparison of Simulation and Calculation

Result Analysis	Throughput
Throughput from simulation	104.74 kbps
Throughput from analysis	105.42 kbps

Table 6-3: Results comparison form simulation and theoretical analysis

Throughput from theoretical analysis matches the results of NetSim's discrete event simulation. The slight difference in throughput is due to two facts that

- The average of random numbers generated for backoff need not be exactly 3.5 as the simulation is run for short time.

- In the packet trace one can notice that there are OSPF and AODV control packets (required for the route setup process) that sent over the network. The data transmissions occur only after the control packet transmissions are completed.

As we go on increasing the simulation time, the throughput value obtained from simulation approaches the theoretical value as can be seen from the table below Table 6-4.

Sample	Simulation Time (sec)	Throughput (kbps)
1	10	103.60
2	50	104.62
3	100	104.57
4	200	104.70

Table 6-4: Throughput comparison with different simulation times

6.2 IoT – Multi-Hop Sensor-Sink Path (Level 3)

NOTE: It is recommended to carry out this experiment in Standard Version of NetSim.

6.2.1 Introduction

The Internet provides the communication infrastructure for connecting computers, computing devices, and people. The Internet is itself an interconnection of a very large number of interconnected packet networks, all using the same packet networking protocol. The Internet of Things will be an extension of the Internet with sub-networks that will serve to connect “things” among themselves and with the larger Internet. For example, a farmer can deploy moisture sensors around the farm so that irrigation can be done only when necessary, thereby resulting in substantial water savings. Measurements from the sensors have to be communicated to a computer on the Internet, where inference and decision-making algorithms can advise the farmer as to required irrigation actions.

Farms could be very large, from a few acres to hundreds of acres. If the communication is entirely wireless, a moisture sensor might have to communicate with a sink that is 100s of meters away. As the distance between a transmitter and a receiver increase, the power of the signal received at the receiver decreases, eventually making it difficult for the signal processing algorithms at the receiver to decode the transmitted bits in the presence of the ever-present thermal noise. Also, for a large farm there would need to be many moisture sensors; many of them might transmit together, leading to collisions and interference.

6.2.2 Theory

The problem of increasing distance between the transmitter and the receiver is solved by placing packet routers between the sensors and the sink. There could even be multiple routers on the path from the sensor to the sink, the routers being placed so that any intermediate link is short enough to permit reliable communication (at the available power levels). We say that there is a multi-hop path from a sensor to the sink.

By introducing routers, we observe that we have a system with sensors, routers, and a sink; in general, there could be multiple sinks interconnected on a separate edge network. We note here that a sensor, on the path from another sensor to the sink, can also serve the role of a router. Nodes whose sole purpose is to forward packets might also need to be deployed.

The problem of collision and interference between multiple transmission is solved by overlaying the systems of sensors, routers, and sinks with a scheduler which determines (preferably in a distributed manner) which transmitters should transmit their packets to which of their receivers.

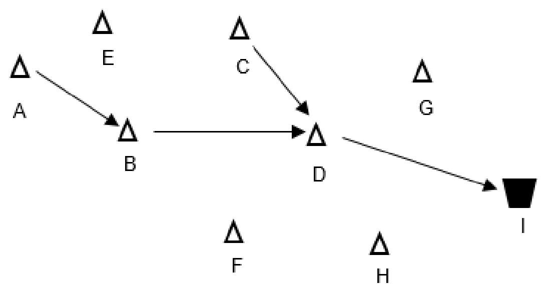


Figure 6-5: Data from Sensor A to Sink I takes the path A-B-D-I while data from sensor C to Sink I takes the path C-D-I

In this experiment, we will use NetSim Simulator to study the motivation for the introduction of packet routers, and to understand the performance issues that arise. We will understand the answers to questions such as:

1. How does packet error rate degrade as the sensor-sink distance increases?
2. How far can a sensor be from a sink before a router needs to be introduced?
3. A router will help to keep the signal-to-noise ratio at the desired levels, but is there any adverse implication of introducing a router?

6.2.3 Network Setup

Open NetSim and click on **Experiments> IOT-WSN> IoT Multi Hop Sensor Sink Path > Packet Delivery Rate and Distance** then click on the tile in the middle panel to load the example as shown in below Figure 6-6.

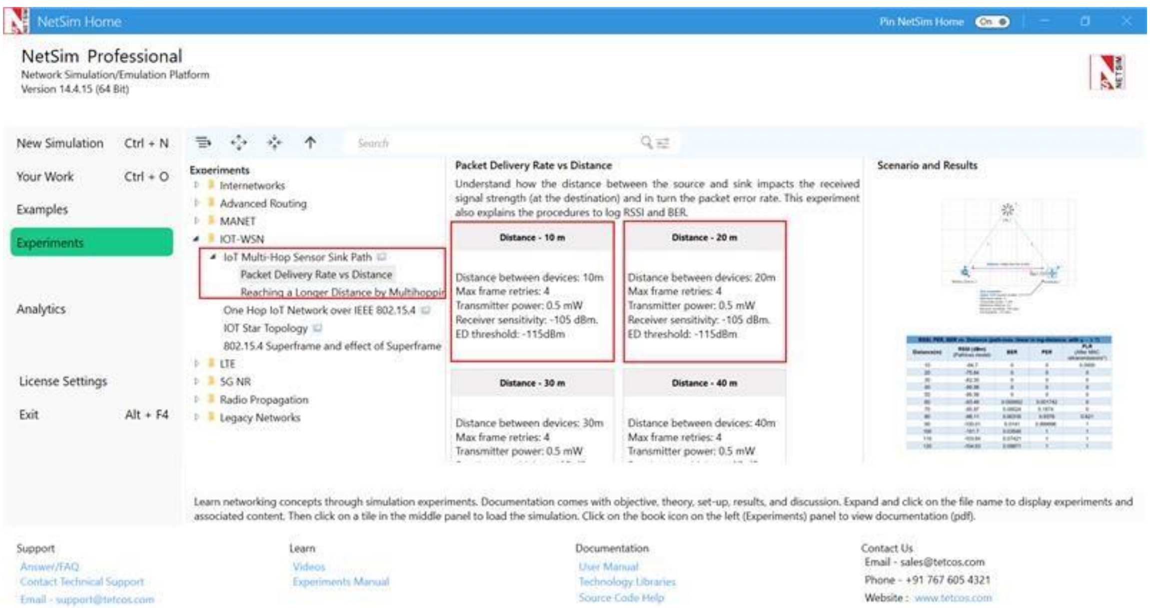


Figure 6-6: List of scenarios for the example of IoT Multi Hop Sensor Sink Path

6.2.4 Packet Delivery Rate vs. Distance

In this part, we perform a simulation to understand, **"How the distance between the source and sink impacts the received signal strength (at the destination) and in turn the packet error rate?"** We will assume a well-established path-loss model under which, as the distance varies, the received signal strength (in dBm) varies linearly. For a given transmit power (say 0dBm), at a certain reference distance (say 8m) the received power is c_0dBm and decreases beyond this point as $-10\eta\ \log_{10} d$ for a transmitter-receiver distance of d. This is called a *power-law* path loss model, since in mW the power decreases as the η power of the distance d. The value of η is 2 for free space path loss and varies from 2 to 5 in the case of outdoor or indoor propagation. Values of η are obtained by carrying out experimental propagation studies.

Distance vs BER PER and RSSI sample

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 6-7.

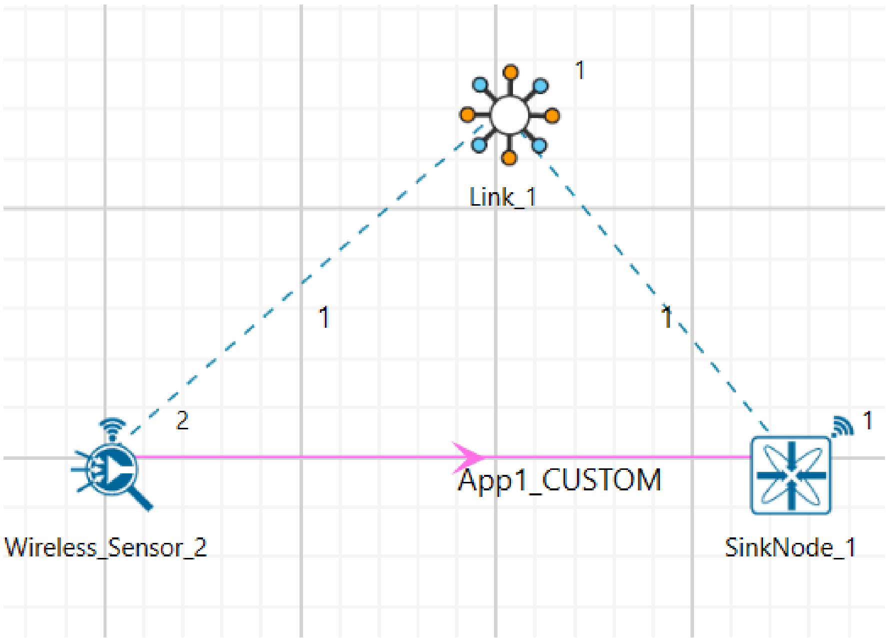


Figure 6-7: Network set up for studying the Distance vs BER PER and RSSI sample

6.2.4.1 Procedure

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

Step 1: A network scenario is designed in the NetSim GUI comprising of a WSN Sink and 1 Wireless Sensor in **Wireless Sensor Networks**.

Note: NetSim currently supports a maximum of only one device, such as a WSN Sink.

Step 2: Set grid length as 150 x 200 from grid setting property panel on the right. This needs to be done before the any device is placed on the grid.

This grid length is initially configured for running a few samples, and as the distance between the sensor and the sink increases, the grid length is adjusted accordingly

Step 3: The distance between the WSN Sink and Wireless Sensor is 10 meters.

Step 4: In the Network Layer properties of Wireless Sensor 2, the Routing Protocol is set to AODV. To configure properties for any node, click on the node, expand the property panel on the right side, and adjust the settings as described in the following step.

Since the routing protocol is a global property, changing it on one node will affect the sink node as well.

Step 5: In the Interface Zigbee > Data Link Layer of Wireless Sensor 2, **Ack Request** is set to Enable and **Max Frame Retries** is set to 4. Similarly, the same settings are done in WSN Sink.

Step 6: In the Interface Zigbee > Physical Layer of Wireless Sensor 2, **Transmitter Power** is set to 0.5mW, **Receiver Sensitivity** is set to -105dBm, and **ED Threshold** is set to -115dBm.

Step 7: Click on link and expand property panel on the right and set the Channel characteristics: Path loss only; Path loss model: Log Distance; Path loss exponent: 3.7.

Step 8: Configure the CUSTOM application between the sensor and sink nodes by clicking on the 'Set Traffic' tab in the ribbon at the top. To set application properties, click on the created application, expand the application property panel on the right, and set the Transport Protocol to UDP, Packet Size to 70 bytes, and Inter Arrival Time to 4000 µs.

The Packet Size and Inter Arrival Time parameters are set such that the Generation Rate equals 140 Kbps. Generation Rate can be calculated using the formula:

$$Generation\ Rate\ (Mbps) = Packet\ Size\ (Bytes) * 8/Interarrival\ time\ (\mu s)$$

Step 9: The following procedures were followed to set Static IP:

Go to Network Layer properties of Wireless Sensor 2 Enable - Static IP Route ->Click on **Configure Via UI**.

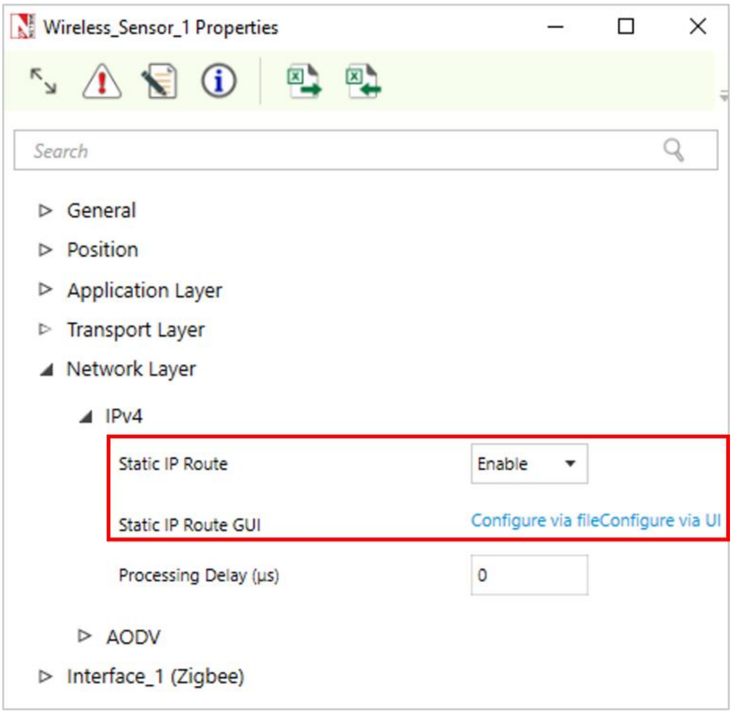


Figure 6-8: Network layer properties window

Static IP Routing Dialogue box gets open.

Enter the Network Destination, Gateway, Subnet Mask, Metrics, and Interface ID. Click on **Add**.

You will find the entry added to the below Static IP Routing Table as shown below.

Click on **OK**.

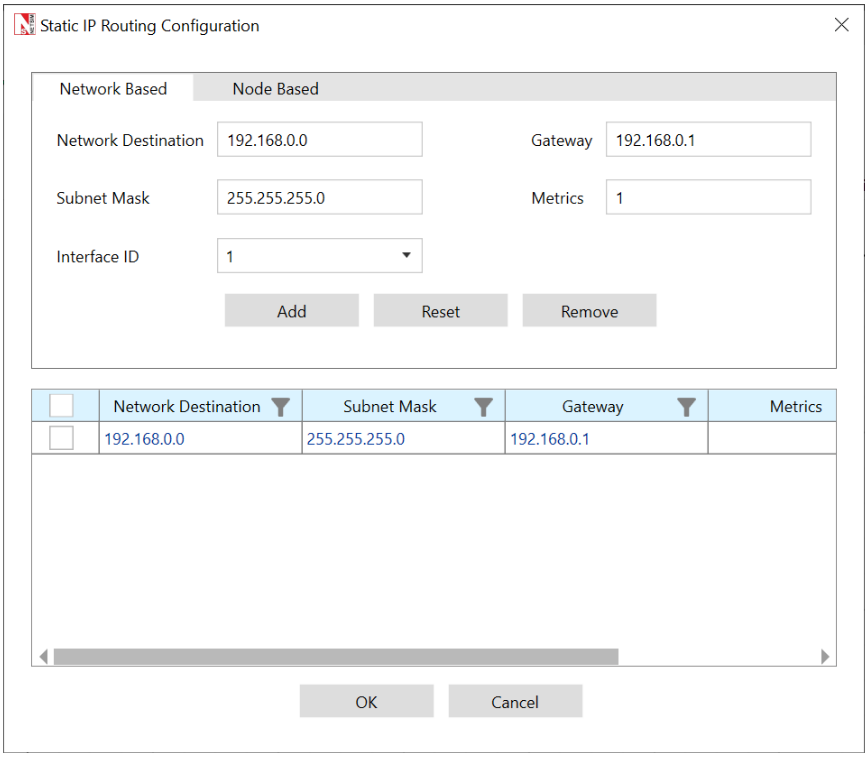


Figure 6-9: Static Route Configuration Window

Step 9: IEEE 802.15.4 Radio Measurements log can be enabled by clicking on Configure Reports tab and Plots as shown below.

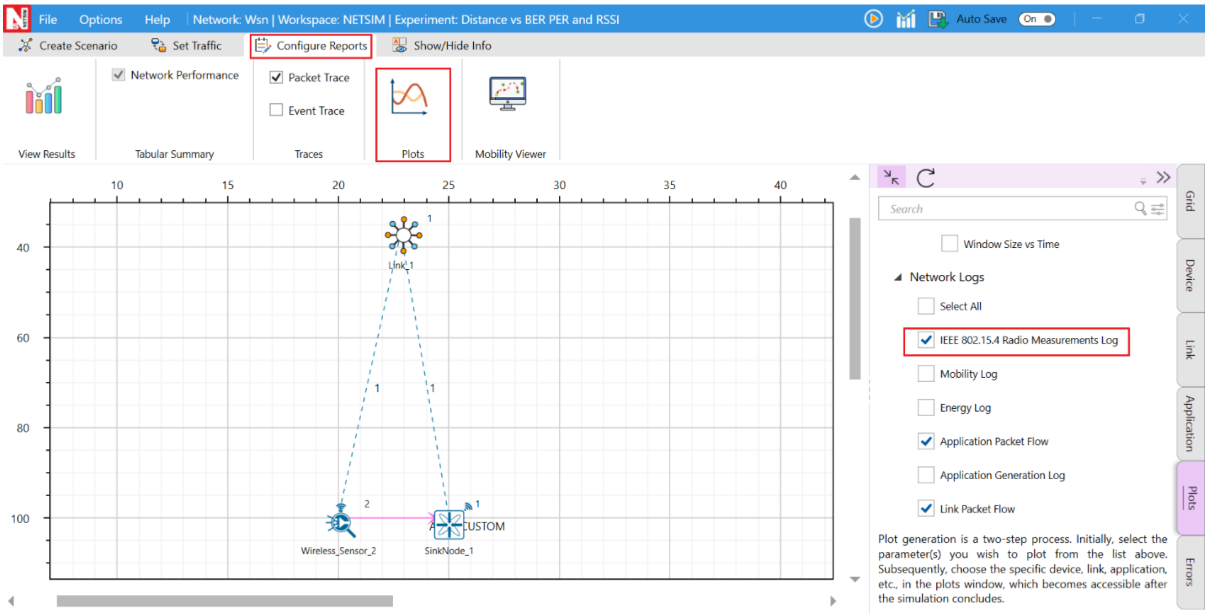


Figure 6-10: Enabling log files in NetSim GUI.

Step 10: Packet Trace is also enabled. At the end of the simulation, a very large .csv file is containing all the packet information is available for the users to perform packet level analysis.

Step 11: Enable the Application and Link plots and run the simulation for 10 seconds. Once the simulation is complete, open the IEEE 802.15.4 Radio Measurements log from the NetSim simulation results window.

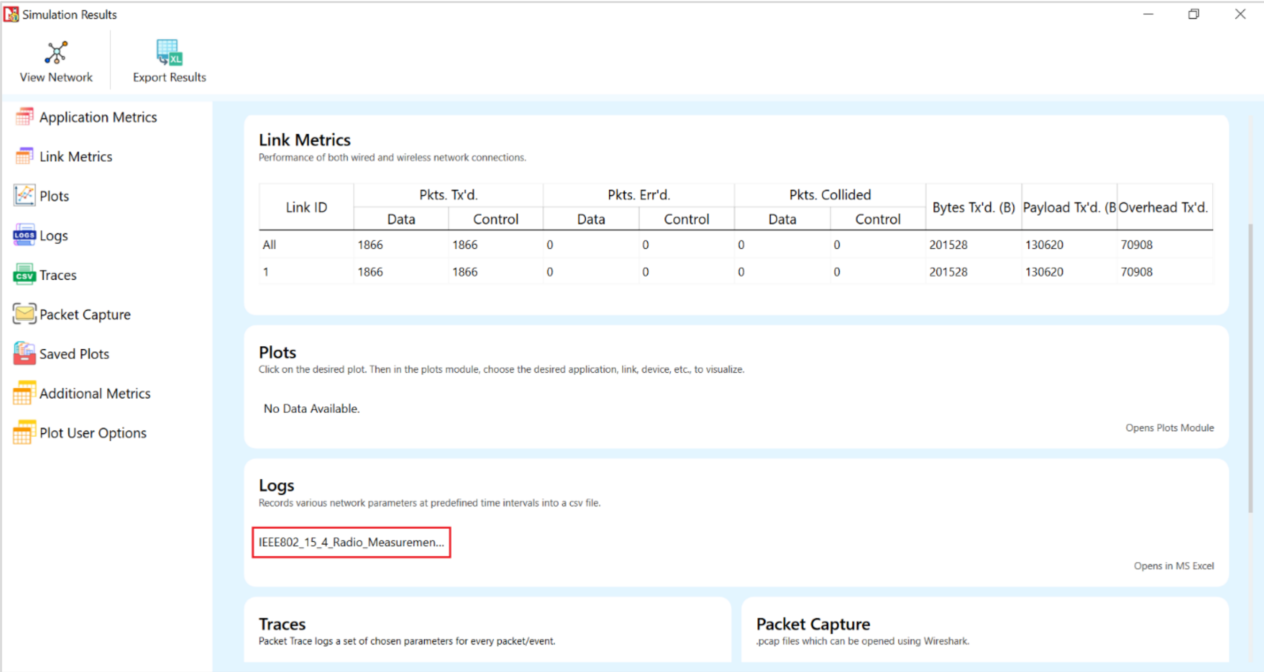
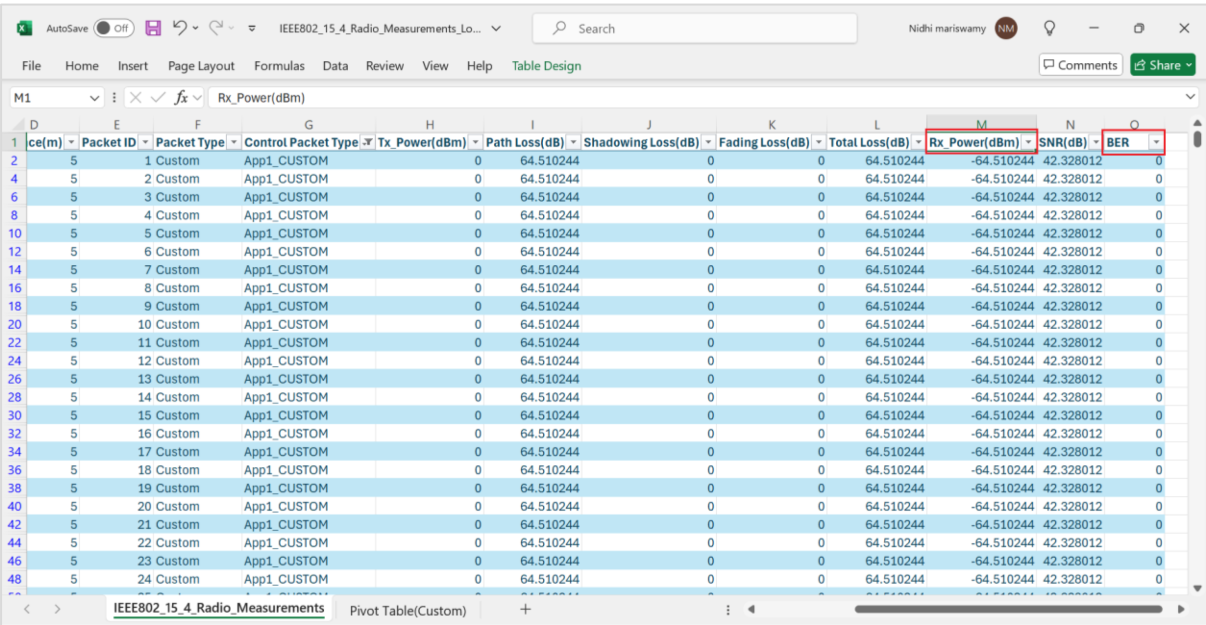


Figure 6-11: Opening IEEE 802.15.4 Radio Measurements log from NetSim Simulation results window

In the IEEE 802.15.4 Radio Measurements log, filter the Control Packet Type to App1_CUSTOM and observe the Rx Power (RSSI) and BER values.



	ce(m)	Packet ID	Packet Type	Control Packet Type	Tx_Power(dBm)	Path Loss(dB)	Shadowing Loss(dB)	Fading Loss(dB)	Total Loss(dB)	Rx_Power(dBm)	SNR(dB)	BER
2	5	1	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
4	5	2	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
6	5	3	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
8	5	4	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
10	5	5	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
12	5	6	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
14	5	7	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
16	5	8	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
18	5	9	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
20	5	10	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
22	5	11	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
24	5	12	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
26	5	13	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
28	5	14	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
30	5	15	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
32	5	16	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
34	5	17	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
36	5	18	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
38	5	19	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
40	5	20	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
42	5	21	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
44	5	22	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
46	5	23	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0
48	5	24	Custom	App1_CUSTOM	0	64.510244	0	0	64.510244	-64.510244	42.328012	0

Figure 6-12: RSSI and BER values in IEEE 802.15.4 Radio Measurements log

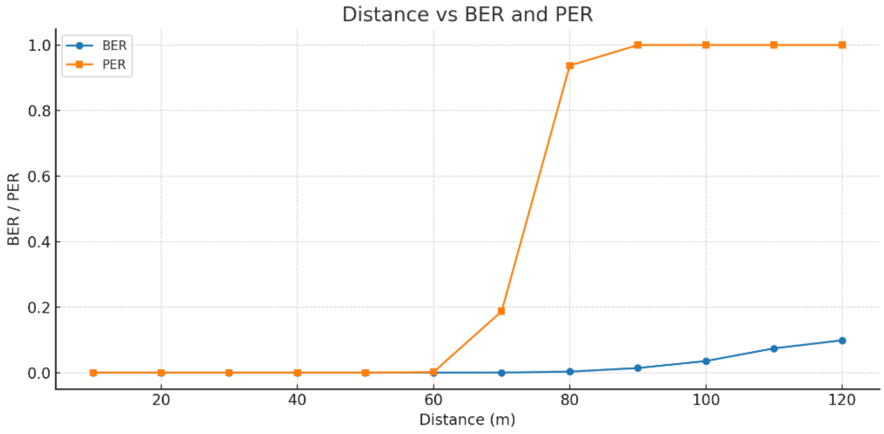
Output for Distance vs BER PER and RSSI sample

RSSI, PER, BER vs. Distance (path-loss: linear in log-distance, with $\eta = 3.7$)				
Distance(m)	**RSSI (dBm)** (Pathloss model)	**BER**	**PER**	**PLR** (After MAC retransmissions*)
10	-64.7	0	0	0

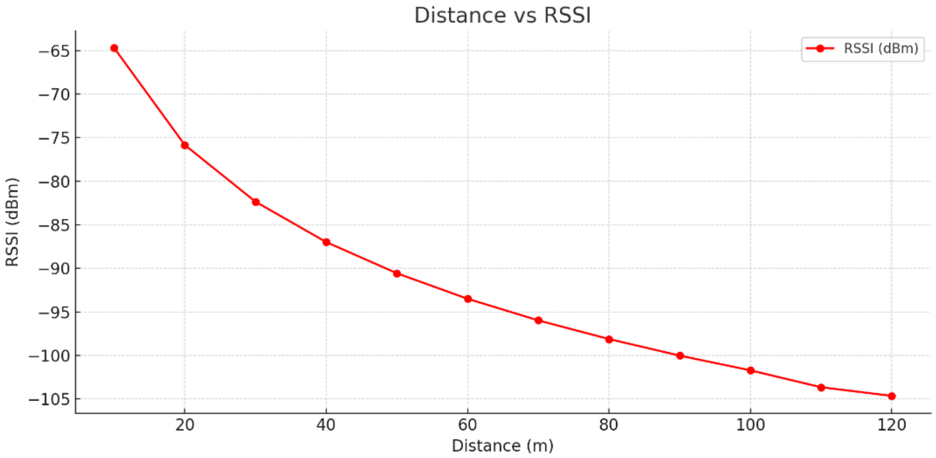
20	-75.84	0	0	0
30	-82.35	0	0	0
40	-86.98	0	0	0
50	-90.56	0	0	0
60	-93.49	0.000002	0.001742	0
70	-95.97	0.00024	0.18885	0
80	-98.11	0.00318	0.9378	0.621
90	-100.01	0.0141	0.9999958	1
100	-101.7	0.03546	1	1
110	-103.64	0.07421	1	1
120	-104.63	0.09871	1	1

Table 6-5: Table showing RSSI, BER results obtained from IEEE 802.15.4 Radio Measurement log and calculated PER value.

Comparison Charts



(a)



(b)

Figure 6-13: (a) Distance Vs. BER, PER and (b) Distance vs. RSSI

* The IEEE 802.15.4 MAC implements a retransmission scheme that attempts to recover errored packets by retransmission. If all the retransmission attempts are also errored, the packet is lost.

The table above reports the RSSI (Received Signal Strength), BER (Bit Error Rate), and Packet Error Rate (PER), and the Packet Loss Rate (PLR) as the distance between the sensor to the sink is increased from 10m to 120m with path loss exponent $\eta = 3.7$. We see that the BER is 0 until a received power of about -90dBm. At a distance of 60m the received power is -93 dBm, and we notice a small BER of 2×10^{-6}. As the distance is increased further the BER continues to grow and at 80m the BER is about 0.00318, yielding $PER = 0.94$, and $PLR = 0.62$. Here PER is obtained from the following formula (which assumes independent bit errors across a packet)

$$PER = 1 - (1 - BER)^{PL},$$

Where,

$$PL - \text{ packet length in bits at the PHY layer}$$

$$PL\ (bits) = \big(70\ (payload) + 39(overhead)\big) * 8$$

The PLR in the above table has been obtained from NetSim, which implements the details of the IEEE 802.15.4 MAC acknowledgement and reattempt mechanism. This mechanism is complex, involving a MAC acknowledgement, time-outs, and multiple reattempts. Analysis of the PLR, therefore, is not straightforward. Assuming that the probability of MAC acknowledgement error is small (since it is a small packet), the PLR can be approximated as PER^{K+1}, where K is the maximum number of times a packet can be retransmitted.

$$PLR = \frac{Total\ number\ of\ Lost\ Packet}{Total\ number\ of\ Packet\ Sent\ by\ Source\ MAC}$$

$$\begin{aligned} Total\ number\ of\ Lost\ packets \\ &= Total\ number\ of\ Packet\ Sent\ by\ SourceMAC \\ &- Packets\ Received\ at\ Destination\ MAC \end{aligned}$$

Steps to calculate Packet Loss Rate

- Open Packet Trace from the Results Dashboard. Filter the PACKET TYPE column as Custom and note down the packet id of the last packet sent from the PACKET ID column.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)	TRX_LA
421	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1680000	
421	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1680000	
421	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1680000	
421	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1680000	
421	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1680000	
421	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1680000	
422	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1684000	
422	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1684000	
422	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1684000	
422	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1684000	
422	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1684000	
422	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1684000	
423	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1688000	
423	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1688000	
423	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1688000	
423	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1688000	
424	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1692000	
424	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1692000	
425	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1696000	
425	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1696000	
425	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1696000	
425	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1696000	
425	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1696000	
425	0	Custom	App1_CUSTOM	SENSOR-2	SINKNODE-1	SENSOR-2	SINKNODE-1	1696000	

Figure 6-14: Packet Trace

This represents the total number of packets sent by the source.

- Note down the Packets Received from the Application Metrics in the Simulation results window as shown in Figure 6-15.

Application Metrics

End-to-end performance of applications running across the network.

App. ID	App. Name	Src. ID	Dest. ID	Gen. Rate Mbps	Thput Mbps	Delay µs	Jitter µs	Pkts. Gen	Pkts. Rcvd	Payload. Gen	Payload. Rcvd
1	App1_CUSTOM	2	1	0.14	0.009016	4043825.360248	51124.03125	2500	161	175000	11270

Figure 6-15: Application metrics table in Result Dashboard

This represents the total number of packets received at the destination.

- Calculate the total number of Lost Packets and PLR as follows:

For the above case,

$$Total\ number\ of\ Packet\ Sent\ by\ SourceMAC = 425$$

$$Packets\ Received\ at\ Destination\ MAC = 161$$

$$Total\ number\ of\ Lost\ packets = 425 - 161 = 264$$

$$PLR = \frac{264}{425} = 0.621$$

6.2.5 Inference

It is clear that Internet applications, such as banking and reliable file transfer, require that all the transmitted data is received with 100% accuracy. The Internet achieves this, in spite of unreliable communication media (no medium is 100% reliable) by various protocols above the network layer. Many IoT applications, however, can work with less than 100% packet delivery without affecting the application. Take, for example, the farm moisture sensing application mentioned in the introduction. The moisture levels vary slowly; if one measurement is lost, the next received measurement might suffice for the decision-making algorithm. This sort of thinking also permits the IoT applications to utilize cheap, low power devices, making the IoT concept practical and affordable.

With the above discussion in mind, let us say that the application under consideration requires a measurement delivery rate of at least 80%. Examining the table above, we conclude that the sensor-sink distance must not be more than 60 meters. Thus, even a 1 acre farm $(61m \times 61m)$ would require multi-hopping to connect sensors to a sink at the edge of the farm.

In Part 2 of this experiment, we will study the placement of a single router between the sensor and the sink, to increase the sensor-sink distance beyond 60 meters.

6.2.6 Reaching a Longer Distance by Multihopping

Direct sensor sink link sample

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 6-16.

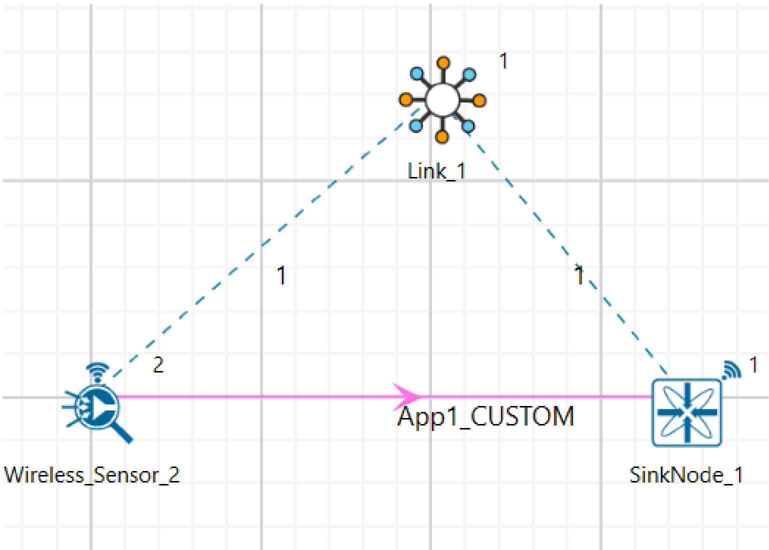


Figure 6-16: Network set up for studying the Direct sensor sink link sample

6.2.6.1 Procedure

The following changes in settings are done from the previous sample:

Step 1: The distance between the WSN Sink and Wireless Sensor is 40 meters.

Step 2: In the Interface Zigbee > Data Link Layer of Wireless Sensor 2, **Ack Request** is set to Enable and **Max Frame Retries** is set to 3. To configure properties for any node, click on the node, expand the property panel on the right side, and adjust the settings as described.

Step 3: The Ad hoc Link properties are set as follows Figure 6-17.

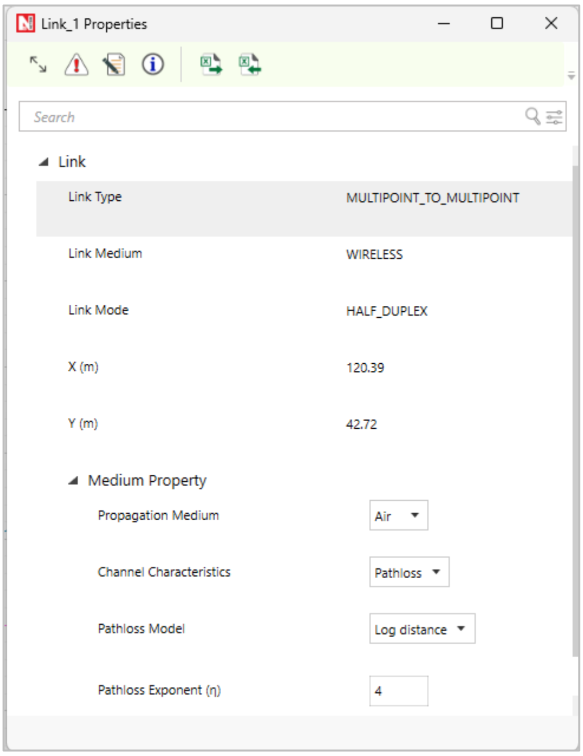


Figure 6-17: Wireless Link properties

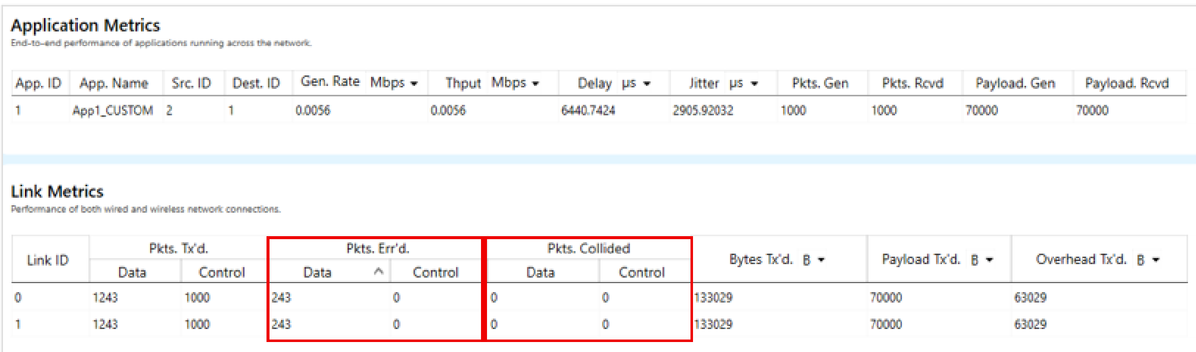
Step 4: Configure the CUSTOM application between the sensor and sink nodes by clicking on the 'Set Traffic' tab in the ribbon at the top. To set application properties, click on the created application, expand the application property panel on the right, and set the Transport Protocol to UDP, Packet Size to 70 bytes, and Inter Arrival Time to 100000 µs.

The Packet Size and Inter Arrival Time parameters are set such that the Generation Rate equals 5.6 Kbps. Generation Rate can be calculated using the formula:

$$Generation\ Rate\ (Mbps)\ =\ Packet\ Size\ (Bytes)\ *\ 8/Interarrival\ time\ (\mu s)$$

Step 5: Run the Simulation for 100 Seconds. Once the simulation is complete, note down the Packet Generated, Packet Received and Throughput values from the **Application Metrics**.

Note down the Packet Errored and Packet Collided from the **Link Metrics**.



Application Metrics

End-to-end performance of applications running across the network.

App. ID	App. Name	Src. ID	Dest. ID	Gen. Rate Mbps	Thput Mbps	Delay µs	Jitter µs	Pkts. Gen	Pkts. Rcvd	Payload. Gen	Payload. Rcvd
1	App1_CUSTOM	2	1	0.0056	0.0056	6440.7424	2905.92032	1000	1000	70000	70000

Link Metrics

Performance of both wired and wireless network connections.

Link ID	Pkts. Tx'd.		Pkts. Err'd.		Pkts. Collided		Bytes Tx'd. B	Payload Tx'd. B	Overhead Tx'd. B
	Data	Control	Data	Control	Data	Control			
0	1243	1000	243	0	0	0	133029	70000	63029
1	1243	1000	243	0	0	0	133029	70000	63029

Figure 6-18: NetSim Results window displaying the number of received, collided, and errored packets.

Router between sensor and sink sample

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 6-19.

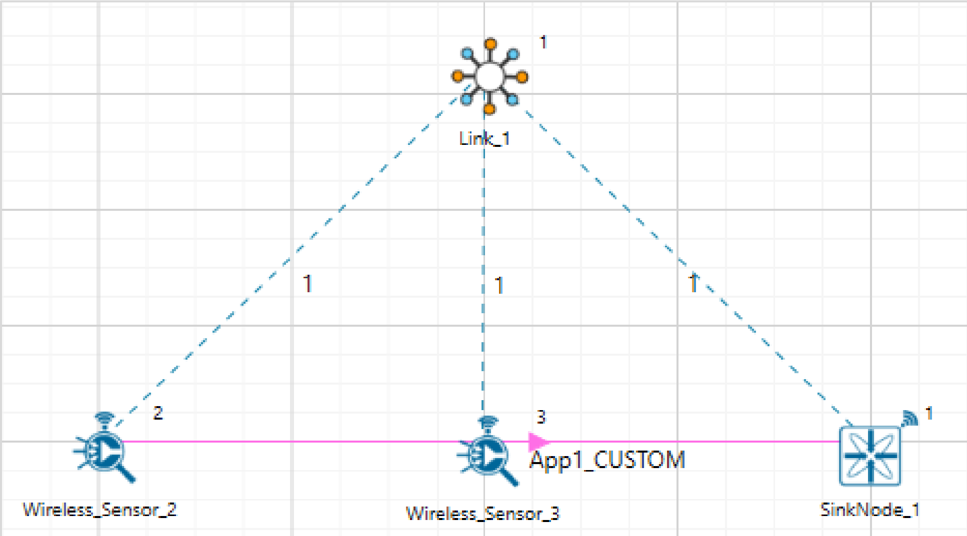


Figure 6-19: Network set up for studying the Router between sensor and sink sample

6.2.6.2 Procedure

The following changes in settings are done from the previous sample:

Step 1: One more Wireless Sensor is added to this network. The distance between Wireless Sensor 2 and Wireless Sensor 3 is 60 meters and the distance between Wireless Sensor 3 and the WSN Sink is 60 meters.

Step 2: The following procedures were followed to set Static IP:

In Network Layer properties of Wireless Sensor 2 Enable - Static IP Route ->Click on Configure via UI as shown in Figure 6-20.

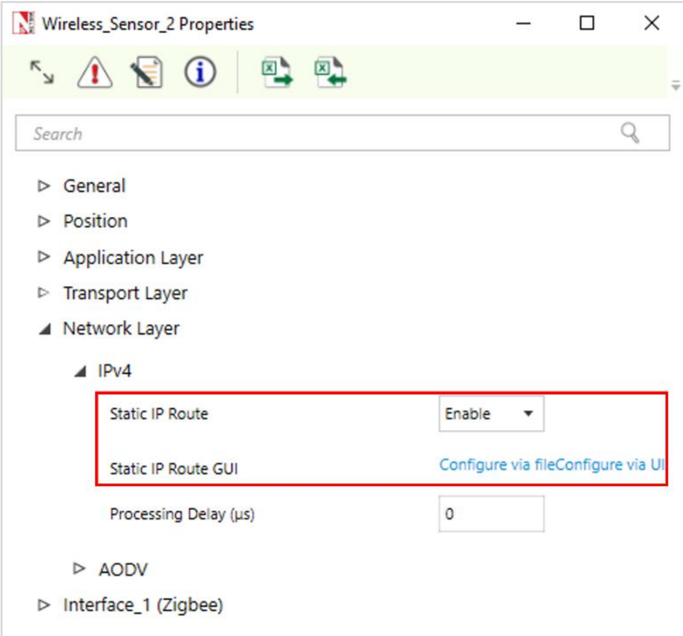


Figure 6-20: Network layer properties window

Static IP Routing Dialogue box gets open.

Static IP Routing Dialogue box gets open.

Enter the Network Destination, Gateway, Subnet Mask, Metrics, and Interface ID. Click on **Add**.

You will find the entry added to the Static IP Routing Table as shown below. Similarly, Static IP is set for Wireless Sensor 3 as shown below Figure 6-21.

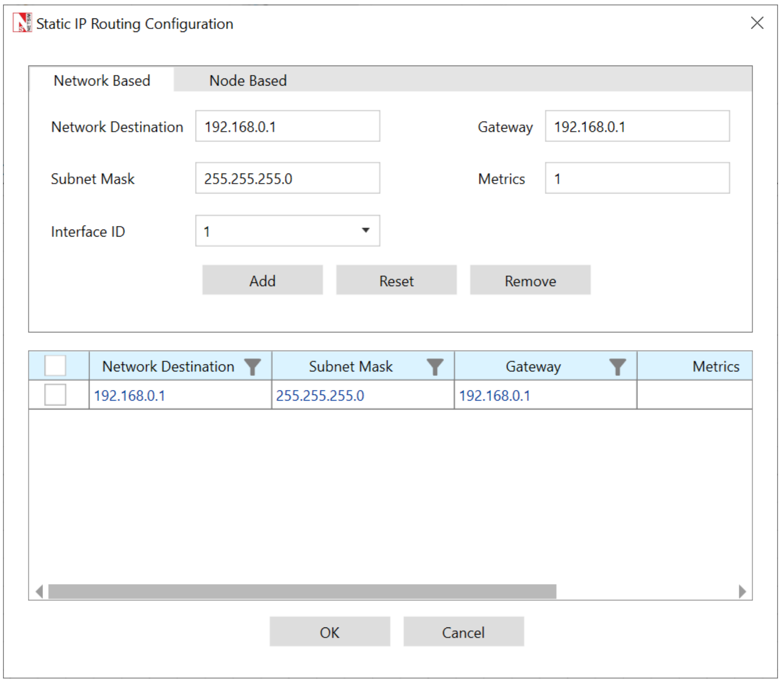
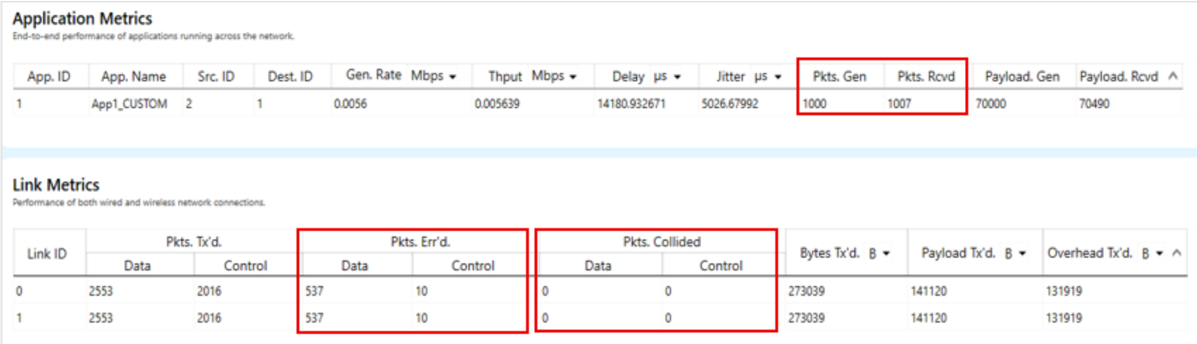


Figure 6-21: Static Route configuration for Wireless Sensor 3

Step 3: Run the Simulation for 100 Seconds. Once the simulation is complete, note down the Packet Generated, Packet Received and Throughput values from the Application Metrics.

Note down the Packet Errored and Packet Collided from the Link Metrics as shown in Table 6-6



Application Metrics
End-to-end performance of applications running across the network.

App. ID	App. Name	Src. ID	Dest. ID	Gen. Rate Mbps	Thput Mbps	Delay µs	Jitter µs	Pkts. Gen	Pkts. Rcvd	Payload. Gen	Payload. Rcvd
1	App1_CUSTOM	2	1	0.0056	0.005639	14180.932671	5026.67992	1000	1007	70000	70490

Link Metrics
Performance of both wired and wireless network connections.

Link ID	Pkts. Tx'd.		Pkts. Err'd.		Pkts. Collided		Bytes Tx'd. B	Payload Tx'd. B	Overhead Tx'd. B
	Data	Control	Data	Control	Data	Control			
0	2553	2016	537	10	0	0	273039	141120	131919
1	2553	2016	537	10	0	0	273039	141120	131919

Figure 6-22: NetSim Results window displaying the number of received, collided, and errored packets.

6.2.7 Output for Router between sensor and sink sample

	Source-Sink Distance(m)	Packets Generated	Packets Received	Packets Errored (PHY)	Packets Collided	Packet Loss (MAC)	PLR	Mean Delay (μs)
Direct sensor-sink link	60	1000	1000	243	0	0	0	6440.74
Router between sensor and sink	120 (router at midpoint)	1000	1007	537	0	-7	-0.007	14180.93

Table 6-6: Packet Generated/Received/Errored/Collided and Mean delay from result dashboard

NOTE: Packet loss (PHY) is the number of packets that were received in error and then recovered by retransmission. Packets received is slightly higher than packets generated on account of retransmissions of successful packets in case of ACK errors.

6.2.8 Inference

In **Distance vs BER PER and RSSI** sample of this experiment, we learnt that if the sensor device uses a transmit power of -3dBm (0.5mW), then for one-hop communication to the sink, the sensor-sink distance cannot exceed 60m. If the sensor-sink distance needs to exceed 60m (see the example discussed earlier), there are two options:

1. The transmit power can be increased. There is, however, a maximum transmit power for a given device. Wireless transceivers based on the CC 2420 have a maximum power of 0dBm (i.e., about 1 mW), whereas the CC 2520 IEEE 802.15.4 transceiver provides maximum transmit power of 5dBm (i.e., about 3 mW). Thus,

given that there is always a maximum transmit power, there will always be a limit on the maximum sensor-sink distance.

2. Routers can be introduced between the sensor and the sink, so that packets from the sensor to the sink follow a *multihop* path. A router is a device that will have the same transceiver as a sensor, but its microcontroller will run a program that will allow it to forward packets that it receives. Note that a sensor device can also be programmed to serve as a router. Thus, in IOT networks, sensor devices themselves serve as routers.

In this part of the experiment, we study the option of introducing a router between a sensor and the sink to increase the sensor-sink distance. We will compare the performance of two networks, one with the sensor communicating with a sink at the distance of 60m, and another with the sensor-sink distance being 120m, with a sensor at the mid-point between the sensor and the sink.

Direct sensor sink link sample simulates a one hop network with a sensor-sink distance of 60m. We recall from Part 1 that, with the transceiver model implemented in NetSim, 60m is the longest one hop distance possible for 100% packet delivery rate. In **Router between sensor and sink** sample, to study the usefulness of routing we will set up network with a sensor-sink distance of 120m with a packet router at the midpoint between the sensor and the sink.

The measurement process at the sensor is such that one measurement (i.e., one packet) is generated every 100ms. The objective is to deliver these measurements to the sink with 100% delivery probability. From Part 1 of this experiment, we know that a single hop of 90m will not provide the desired packet delivery performance.

The Table at the beginning of this section shows the results. We see that both networks are able to provide a packet delivery probability of 100%. It is clear, however, that since the second network has two hops, each packet needs to be transmitted twice, hence the mean delay between a measurement being generated and it being received at the sink is doubled. Thus, the longer sensor-sink distance is being achieved, for the same delivery rate, at an increased delivery delay.

The following points may be noted from the table:

1. The number of packets lost due to PHY errors. The packet delivery rate is 100% despite these losses since the MAC layer re-transmission mechanism is able to recover all lost packets.

2. There are no collisions. Since both the links (sensor-router and router-sink) use the same channel and there is no co-ordination between them, it is possible, in general for sensor-router and router-sink transmissions to collide. This is probable when the measurement rate is large, leading to simultaneously nonempty queues at the sensor and router. In this experiment we kept the measurement rate small such that the sensor queue is empty when the router is transmitting and vice versa. This avoids any collisions.

6.3 Performance Evaluation of a Star Topology IoT Network (Level 3)

6.3.1 Introduction

In IOT experiments 20 and 21 we studied a single IoT device connected to a sink (either directly or via IEEE 802.15.4 routers). Such a situation would arise in practice, when, for example, in a large campus, a ground level water reservoir has a water level sensor, and this sensor is connected wirelessly to the campus wireline network. Emerging IoT applications will require several devices, in close proximity, all making measurements on a physical system (e.g., a civil structure, or an industrial machine). All the measurements would need to be sent to a computer connected to the infrastructure for analysis and inferencing. With such a scenario in mind, in this experiment, we will study the performance of several IEEE 802.15.4 devices each connected by a single wireless link to a sink. This would be called a “star topology” as the sensors can be seen as the spikes of a “star.”

We will set up the experiment such that every sensor can sense the transmissions from any other sensor to the sink. Since there is only one receiver, only one successful transmission can take place at any time. The IEEE 802.15.4 CSMA/CA multiple access control will take care of the coordination between the sensor transmissions. In this setting, we will conduct a saturation throughput analysis. The IoT communication buffers of the IEEE 802.15.4 devices will always be nonempty, so that as soon as a packet is transmitted, another packet is ready to be sent. This will provide an understanding of how the network performs under very heavy loading. For this scenario we will compare results from NetSim simulations against mathematical analyses.

Details of the IEEE 802.15.4 PHY and MAC have been provided in the earlier IOT experiments 20 and 21, and their understanding must be reviewed before proceeding further with this experiment. In this experiment, all packets are transferred in a single hop from and IoT device to the sink. Hence, there are no routers, and no routing to be defined.

6.3.2 NetSim Simulation Setup

Open NetSim and click on **Experiments> IOT-WSN> IOT Star Topology** then click on the tile in the middle panel to load the example as shown in below Figure 6-23.

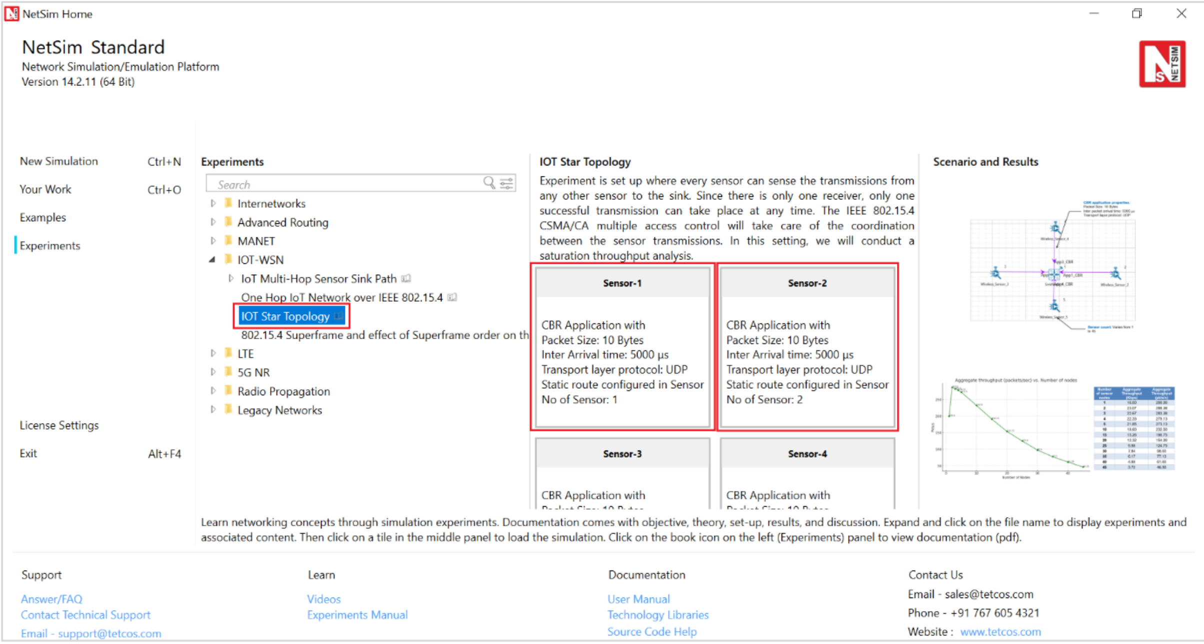


Figure 6-23: List of scenarios for the example of IOT Star Topology

The simulation scenario consists of n nodes distributed uniformly around a sink (PAN coordinator) at the center. In NetSim the nodes associate with the PAN coordinator at the start of the simulation. CBR traffic is initiated simultaneously from all the nodes. The CBR packet size is kept as 10 bytes to which 20 bytes of IP header, 7 bytes of MAC header and 6 bytes of PHY header are added. To ensure saturation, the CBR traffic interval is kept very small; each node's buffer receives packets at intervals of 5 ms

6.3.3 Procedure

Sensor-1

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 6-24.

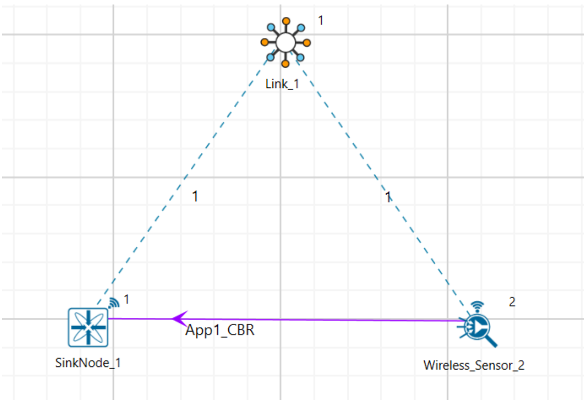


Figure 6-24: Network set up for studying the Sensor 1

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

Step 1: A network scenario is designed in NetSim GUI comprising of 1 SinkNode, and 1 wireless sensor in the **“WSN”** Network Library.

Step 2: Click on sensor, expand the property panel on right and configure the static route in network layer as shown in below Table 6-7**.**

Network Destination	Gateway	SubnetMask	Metrics	Interface ID
192.168.0.0	192.168.0.1	255.255.255.0	1	1

Table 6-7: Static route for Wireless sensor

Step 3: Click on Adhoc link, expand the link property panel on right and set channel characteristics to No pathloss.

Step 4: Configure a CBR application from Wireless Sensor 2 (Source) to Sink Node 1 (Destination) by clicking on the **Set Traffic** tab in the ribbon at the top. Then, click on the created application and set the packet size to 10 bytes, the inter arrival time to 5000 µs, the start time to 5 seconds, and the transport protocol to UDP.

Step 5: Plots are enabled from configure reports tab and Simulation time is set to 100 seconds.

Increase wireless sensor count to 2, 3, and 4 with the same above properties to design Sensor-2, 3, and 4.

Sensor-5: NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 6-25.

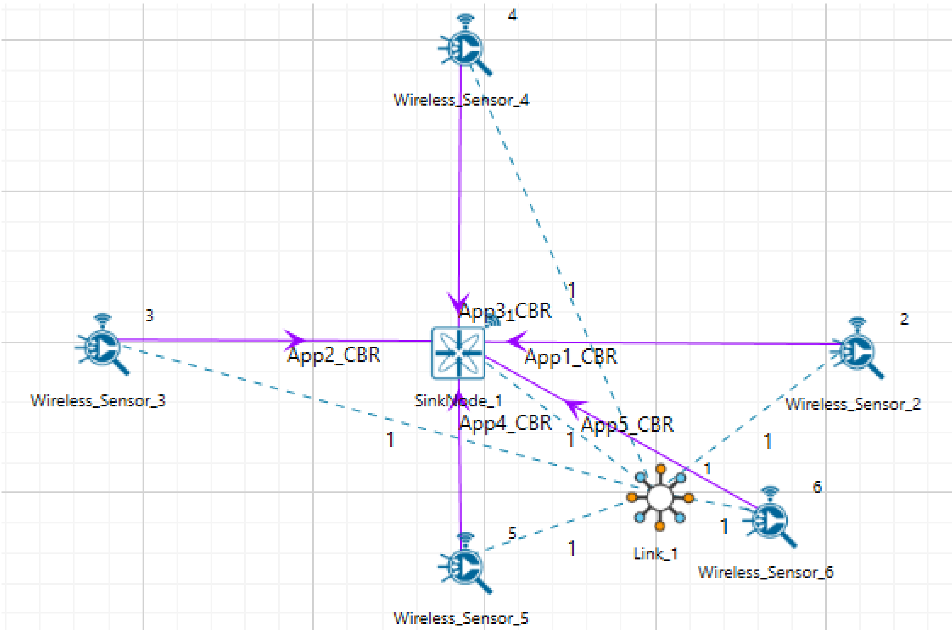


Figure 6-25: Network set up for studying the Sensor-5

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

Step 1: Click on sensor, expand the property panel on right and configure the static route in network layer as shown in below Table 6-8.

Network Destination	Gateway	SubnetMask	Metrics	Interface ID
192.168.0.0	192.168.0.1	255.255.255.0	1	1

Table 6-8: Static route for Wireless sensors

Step 2: Configure CBR application by clicking on set traffic tab from the ribbon on the top. Set the applications properties as mentioned in below Table 6-9.

	App 1	App 2	App 3	App 4	App 5
Application	CBR	CBR	CBR	CBR	CBR
Source ID	2	3	4	5	6
Destination ID	1	1	1	1	1
Start time	5s	5s	5s	5s	5s
Transport protocol	UDP	UDP	UDP	UDP	UDP
Packet Size	10bytes	10bytes	10bytes	10bytes	10bytes
Inter-Arrival Time	5000 µs	5000 µs	5000 µs	5000 µs	5000 µs

Table 6-9: Detailed Application properties

Step 3: Plots are enabled in NetSim GUI. Click on Run simulation. The simulation time is set to 100 seconds.

Increase wireless sensor count to 10, 15, 20, 25, 30, 35, 40, and 45 with the same above properties to design Sensor-6, 7, 8, 9, 10, 11, 12, and 13.

6.3.4 Output

The aggregate throughput of the system can be got by adding up the individual throughput of the applications. NetSim outputs the results in units of kilobits per second (kbps). Since the packet size is $80\ bits$ we convert per the formula

$$Aggregate\ Throughput\ (pkts\ per\ sec) = \frac{Aggregate\ Throughput\ (kbps) * 1000}{80}$$

Number of Nodes	Aggregate Throughput (Kbps)	Aggregate Throughput (pkts/s)
1	16.00	200.00
2	23.07	288.38
3	22.67	283.38
4	22.33	279.13
5	21.85	273.13
10	18.60	232.50
15	15.26	190.75
20	12.32	154.00
25	9.98	124.75
30	7.84	98.00
35	6.16	77.13

40	4.88	61.00
45	3.72	46.50

Table 6-10: Aggregate throughput for Star topology scenario

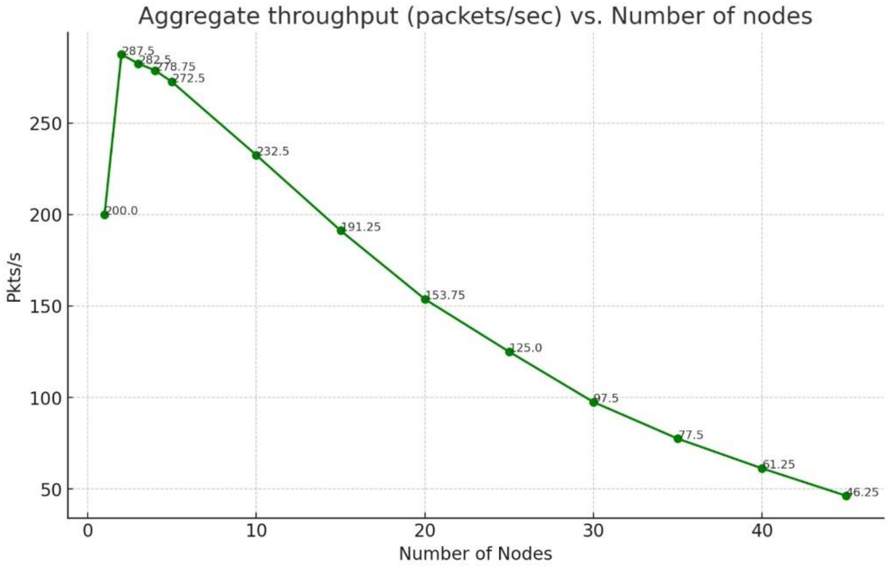


Figure 6-26: The plot illustrates the relationship between aggregate throughput (measured in packets per second) and the number of nodes within an IoT star topology network. It demonstrates how throughput varies as the network size increases

6.3.5 Discussion

In Figure 6-26 we plot the throughput in packets per second versus the number of IoT devices in the star topology network. We make the following observations:

1. Notice that the throughput for a single saturated node is 200 packets per second, which is just the single hop throughput from a single saturated node, when the packet payload is 10 bytes. When a node is by itself, even though there is no contention, still the node goes through a backoff after every transmission. This backoff is a waste and ends up lowering the throughput as compared to if the node knew it was the only one in the network and sent packets back-to-back.

2. When there are two nodes, the total throughput increases to 287.5 packets per second. This might look anomalous. In a contention network, how can the throughput increase with more nodes? The reason can be found in the discussion of the previous observation. Adding another node helps fill up the time wasted due to redundant backoffs, thereby increasing throughput.

3. Adding yet another node results in the throughput dropping to 282.5 packets per second, as the advantage gained with 2 nodes (as compared to 1 node) is lost due to more collisions.

4. From there on as the number of nodes increases the throughput drops rapidly to about 100 packets per second for about 30 nodes.

5. The above behaviour must be compared with a Experiment 12 where several IEEE 802.11 STAs, with saturated queues, were transmitting packets to an AP. The throughput increased from 1 STA to 2 STAs, dropped a little as the number of STAs increased and then flattened out. On the other hand, in IEEE 802.15.4 the throughput drops rapidly with increasing number of STAs. Both IEEE 802.11 and IEEE 802.15.4 have CSMA/CA MACs. However, the adaptation in IEEE 802.11 results in rapid reduction in per-node attempt rate, thus limiting the drop in throughput due to high collisions. On the other hand, in IEEE 802.15.4, the per-node attempt rate flattens out as the number of nodes is increased, leading to high collisions, and lower throughput. We note, however, that IoT networks essentially gather measurements from the sensor nodes, and the measurement rates in most applications are quite small.

6.3.6 References

1. Chandramani Kishore Singh, Anurag Kumar. P. M. Ameer (2007). Performance evaluation of an IEEE 802.15.4 sensor network with a star topology. Wireless Network (2008) 14:543–568.

6.4 Study the 802.15.4 Superframe Structure and analyze the effect of Superframe order on throughput (Level 3)

6.4.1 Introduction

A coordinator in a PAN can optionally bound its channel time using a Superframe structure which is bound by beacon frames and can have an active portion and an inactive portion. The coordinator enters a low-power (sleep) mode during the inactive portion.

The structure of this Superframe is described by the values of macBeaconOrder and macSuperframeOrder. The MAC PIB attribute macBeaconOrder, describes the interval at which the coordinator shall transmit its beacon frames. The value of macBeaconOrder, BO, and the beacon interval, BI, are related as follows:

For 0 ≤ BO ≤ 14, BI = aBaseSuperframeDuration * 2^{BO} symbols.

If BO = 15, the coordinator shall not transmit beacon frames except when requested to do so, such as on receipt of a beacon request command. The value of macSuperframeOrder, SO shall be ignored if BO = 15.

An example of a Superframe structure is shown in following Figure 6-27.

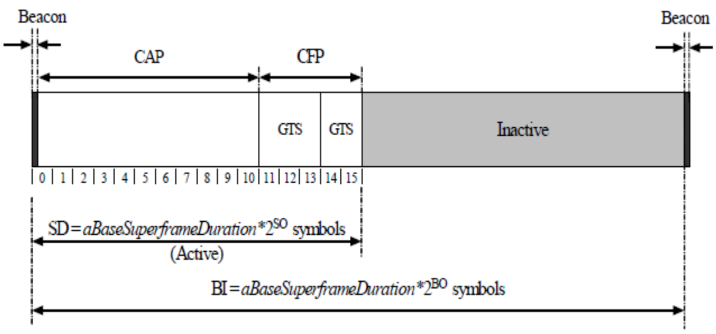


Figure 6-27: An example of the Super Frame structure

Theoretical Analysis

From the above Superframe structure,

$$Beacon\ Interval = aBaseSuperframeDuration * 2^{BO}$$

$$Active\ part\ of\ SuperFrame = aBaseSuperframeDuration * 2^{SO}$$

$$Inactive\ part\ of\ SuperFrame = aBaseSuperframeDuration * (2^{BO} - 2^{SO})$$

If Superframe Order (SO) is same as Beacon Order (BO) then there will be no inactive period and the entire Superframe can be used for packet transmissions.

If BO=10, SO=9 half of the Superframe is inactive and so only half of Superframe duration is available for packet transmission. If BO=10, SO=8 then $(3/4)^{th}$ of the Superframe is inactive and so nodes have only $(1/4)^{th}$ of the Superframe time for transmitting packets and so we expect throughput to approximately drop by half of the throughput obtained when SO=9.

Percentage of inactive and active periods in Superframe for different Superframe Orders is given below Table 6-11. This can be understood from Beacon Time Analysis section of the IoT-WSN technology library manual.

Beacon Order (BO)	Super Frame Order (SO)	Active part of Superframe(%)	Inactive part of Superframe (%)	Throughput estimated (%)
10	10	100	0	> 200% of T
10	9	50	50	Say T = 21.07 (Got from simulation)
10	8	25	75	50 % T
10	7	12.5	87.5	25 % T
10	6	6.25	93.75	12.5 % of T
10	5	3.125	96.875	6.25 % of T
10	4	1.5625	98.4375	3.12% of T
10	3	0.78125	99.21875	1.56 % of T

Table 6-11: Inactive and active periods in Superframe for different Superframe

We expect throughput to vary in the active part of the Superframe as sensors can transmit a packet only in the active portion.

6.4.2 Network Setup

Open NetSim and click on **Experiments> IOT-WSN> 802.15.4 Superframe and effect of Superframe order on throughput** then click on the tile in the middle panel to load the example as shown in below Figure 6-28.

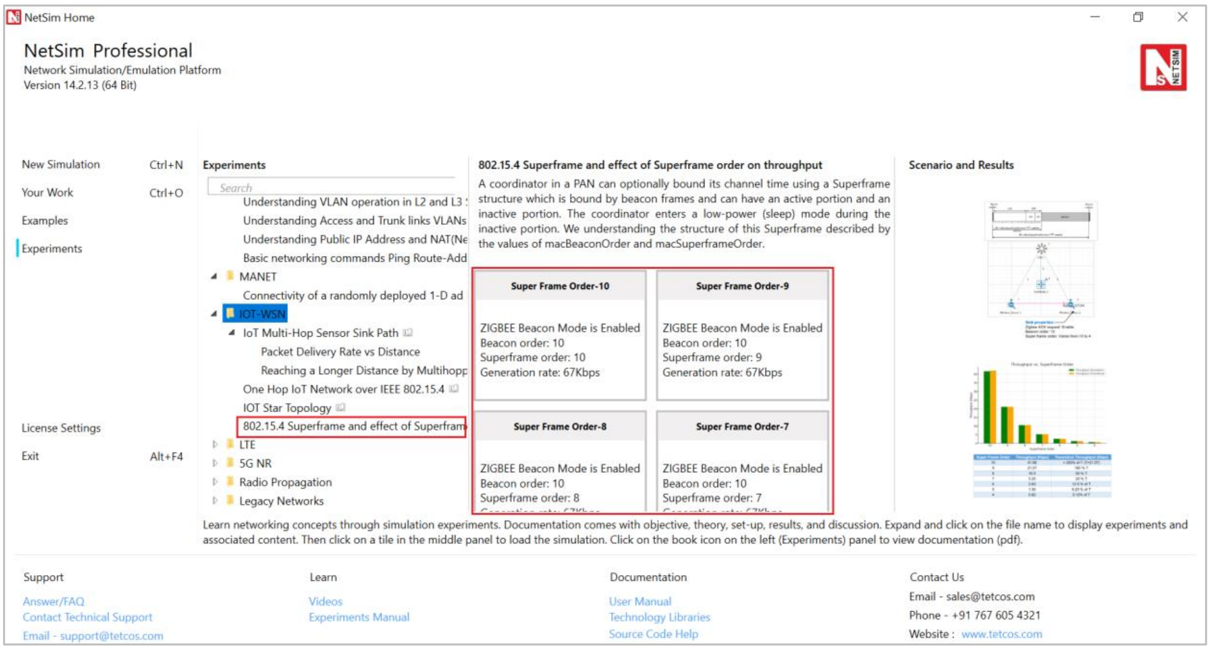


Figure 6-28: List of scenarios for the example of 802.15.4 Superframe and effect of Superframe order on throughput

Super Frame Order 10 Sample

NetSim UI displays the configuration file corresponding to this experiment as shown below Figure 6-29.

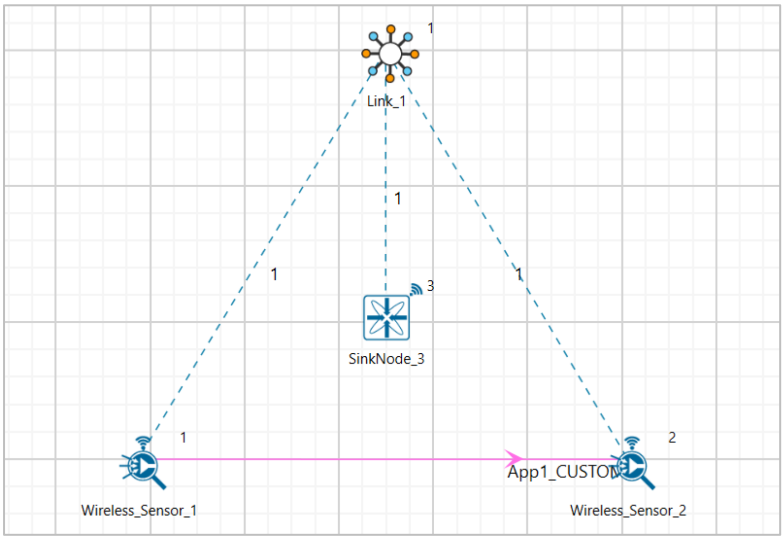


Figure 6-29: Network set up for studying the Super Frame Order 10 Sample

6.4.3 Procedure

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

Step 1: A network scenario is designed in NetSim GUI comprising of 2 Wireless Sensors and a WSN Sink in the **“Wireless Sensor Networks”** Network Library.

Step 2: Before we designed this network, in the **Fast Config Window** containing inputs for **Grid Settings and Placement Strategy**, the Grid Length and width were set to 500 and 250 meters respectively.

Step 3: Click on the WSN Sink, expand the property panel on the right, and set the Beacon mode to Enable. In the Data Link Layer of the Interface Zigbee, set the Beacon Order and Superframe Order to 10 as shown below.

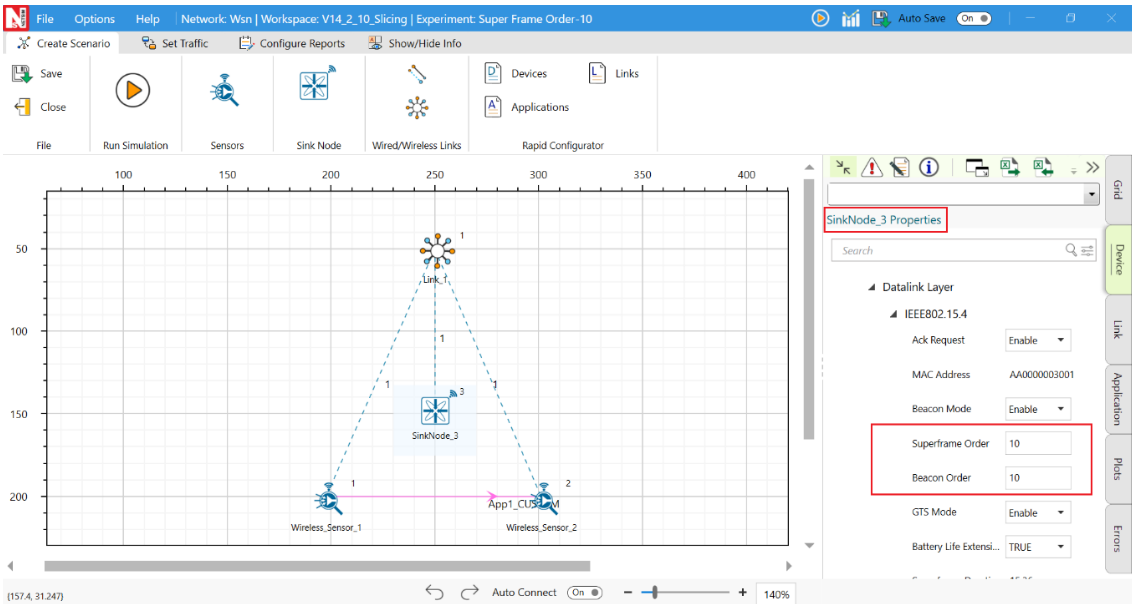


Figure 6-30: Setting Superframe and Beacon order in Sink node

Step 4: The **Ad hoc Link** is used to link the Sensors and the WSN Sink in an ad hoc basis. To set the channel characteristics, click on Ad hoc link and expand link property panel on right and set channel characteristics to No pathloss.

Step 5: : Configure a custom application between sensors by clicking on the 'Set Traffic' tab in the ribbon at the top. Click on the created application, set the packet size to 25 bytes, and set the inter-arrival time to 3000 µs.

The Packet Size and Inter Arrival Time parameters are set such that the Generation Rate approximately equals 67 Kbps. Generation Rate can be calculated using the formula:

$$Generation\ Rate\ (Mbps)\ =\ Packet\ Size\ (Bytes)\ *\ 8/Interarrival\ time\ (\mu s)$$

Step 6: Run the Simulation for 30 Seconds and note down the **Throughput** value.

Similarly, run the other samples by varying the Super Frame Order to 9, 8, 7, 6, 5, and 4 and note down the throughput values.

6.4.4 Output

The following are the throughputs obtained from the simulation for different Super Frame Orders Table 6-12.

Super Frame Order	Throughput (Kbps)	Theoretical Throughput (Kbps)
10	41.68	> 200% of T (T=21.07)
9	21.07	100 % T
8	10.5	50 % T
7	5.25	25 % T
6	2.63	12.5 % of T
5	1.30	6.25 % of T
4	0.62	3.12% of T

Table 6-12: Different Super Frame Orders vs. throughputs

To obtain throughput from simulation, payload transmitted values will be obtained from Link metrics and calculated using following formula:

$$Application\ Throughput(in\ Mbps) = \frac{Total\ payload\ delivered\ to\ destination\ (bytes)\ *\ 8}{Simulation\ Time(Millisecond)\ -App\ Start\ Time(Millisecond)}$$

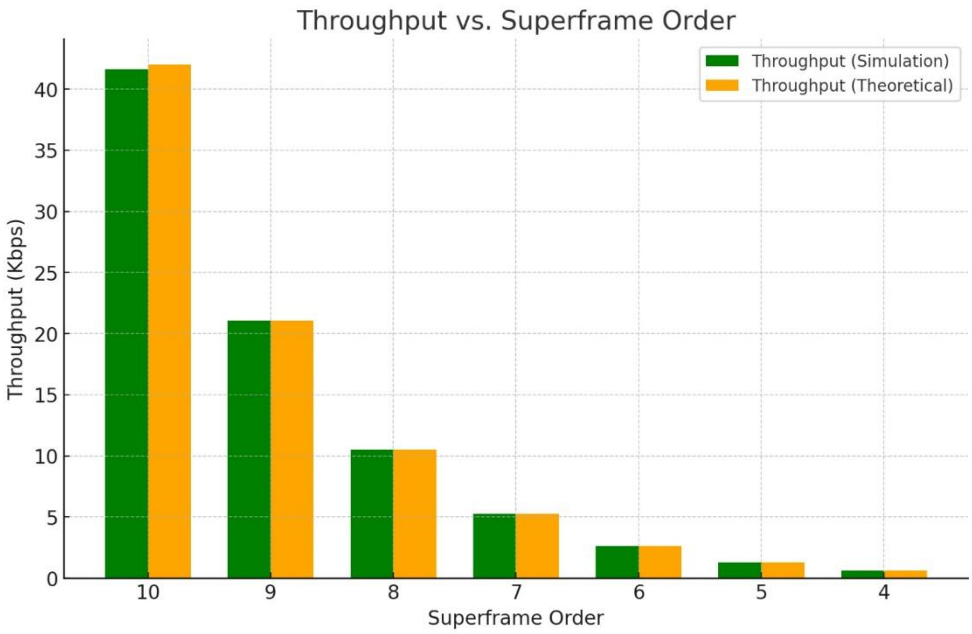


Figure 6-31: This plot presents a comparative analysis of Superframe order against both simulated and theoretical throughput values. Simulation results agree well with theoretical expectations.

Comparison Chart: All the above plots highly depend upon the placement of Sensor in the simulation environment. So, note that even if the placement is slightly different the same set of values will not be got but one would notice a similar trend.

6.4.5 Inference

From the comparison chart both the simulation and theoretical throughputs match except for the case with no inactive period. A sensor will be idle if the last packet in its queue is transmitted. If a packet is generated in inactive period, then the packet has to wait in the queue till the next Superframe so sensor has packets waiting in its queue and so it cannot be idle in the next Superframe, but if there is no inactive period then there might be no packets waiting in the queue and so sensor can be idle resulting in lesser throughput.

7 Radio Propagation

7.1 Pathloss, Shadowing and Fading (Level 1)

7.1.1 Part 1: Pathloss and Shadowing

Log-distance pathloss

Path loss is the reduction in power density of an electromagnetic wave as it propagates through space. Path loss is a positive quantity measured in dB and is defined as the difference (in dB) between the transmitted power and the received power and may or may not include the effect of antenna gains.

Both theoretical and measurement-based propagation models indicate that average received signal power decreases logarithmically with distance whether in outdoor or in indoor ratio channels. The average large-scale pathloss for an arbitrary transmitter-receiver separation is expressed as a function of distance by using the pathloss exponent η, which indicates the rate at which the pathloss increases with distance [4]. Therefore pathloss $PL(d) \propto \left(\frac{d}{d_0}\right)^{\eta}$.

The general formula (when written in dB scale) by which received power is calculated when using the log-distance model and including the effect of antenna gains is

$$P_r = P_t + G_t + G_r + 20 \cdot log_{10}\left(\frac{\lambda}{(4 \cdot \pi \cdot d_0)}\right) + \left(10 \cdot \eta \cdot log_{10}\left(\frac{d_0}{d}\right)\right)$$

Where,

P_r is the received power,

P_t is the transmit power,

G_t is the transmitter antenna gain, and G_r is the receiver antenna gain.

d is the distance between the transmitter and receiver.

d_0 is the reference distance, and the model is applicable only for $d > d_0$

λ is the wavelength and is equal to $\frac{c}{f}$ where c is the speed of light and f is the frequency in Hz

η is the path loss exponent, whose value is normally in the range of 2 to 5, Define PL_{d0}, the path loss at reference distance as $PL_{do} = 20 \cdot log_{10}\left(\frac{4 \cdot \pi \cdot d_0}{\lambda}\right)$. Now, when the antenna gains at the transmitter and receiver is unity, i.e., when $G_t = G_r = 0\ dB$, the general pathloss formula can be re-written as

$$P_r = P_t - \underbrace{(PL_{d0} + 10 \cdot \eta \cdot \log_{10}(d))}_{Pathloss}$$

which shows us that (in the dB scale) the received power is the transmitted power minus the pathloss.

Log Normal Shadowing

With pathloss models the predicted path loss between a transmitter and a receiver is constant for a given distance. However, different types of terrain and clutter may exist in the transmitter-receiver path. Therefore, the path losses may be vastly different than the *average* value predicted in (1), for two different locations having the same transmitter-receiver separation distance. Some paths undergo more loss while others are less obstructed and may have higher received signal strength. The variation of path loss with respect to the mean path loss values predicted by the propagation models, depending on the type of environment is called *shadowing*. Shadowing is also termed *shadow-fading* or *slow fading*.

Models for path loss and shadowing can be superimposed to capture power fall off versus distance along with the random attenuation about this average path loss from shadowing. Empirical studies have shown that this randomness is captured well by a second factor, S, of the form $10^{-\frac{\chi}{10}}$ with χ being a Gaussian random variable with mean 0 and variance σ^2. This is called the shadowing component of the attenuation, and, since $10 \cdot \log_{10}(S)$ has a Gaussian (or normal) distribution, it is called log-normal shadowing [5]. In the combined model the total loss in dB scale is given by

$$P_r = P_t - \underbrace{(PL_{d0} + 10 \cdot \eta \cdot \log_{10}(d))}_{Pathloss} - \underbrace{\chi}_{shadowing}$$

Where χ is a zero-mean Gaussian distributed random variable (in dB) with standard deviation σ (in dB). In NetSim, the default value for $\sigma\ is\ 5\ dB$, and the range of σ (in dB) is $5\ \leq\ \sigma\ \leq\ 12$.

7.1.1.1 Network Setup

Pathloss

Step 1: Open NetSim and click on Mobile Ad hoc Network.

Step 2: Create a scenario with 2 wireless nodes.

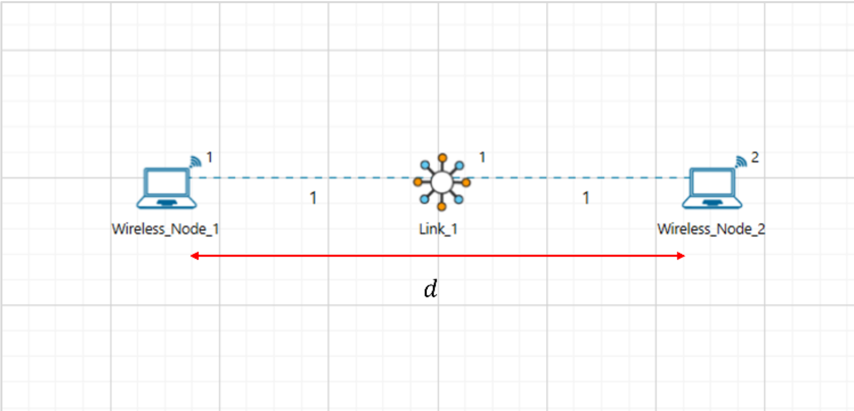


Figure 7-1: A Network consists of 2 Wireless Nodes. Wireless Node 1 and Wireless Node 2 separated by distance $\boldsymbol{d}$ in meters. The values of d are provided in Table 7-4.

Step 3: Go to the Properties of Wireless Node 1 → Position Layer → Mobility Layer → Mobility Model set to No Mobility.

Step 4: Go to the Properties of Wireless Node 1 → Network Layer → Enable the static route.

Step 5: Configure static routing in Wireless Node 1 as shown in below Figure 7-2

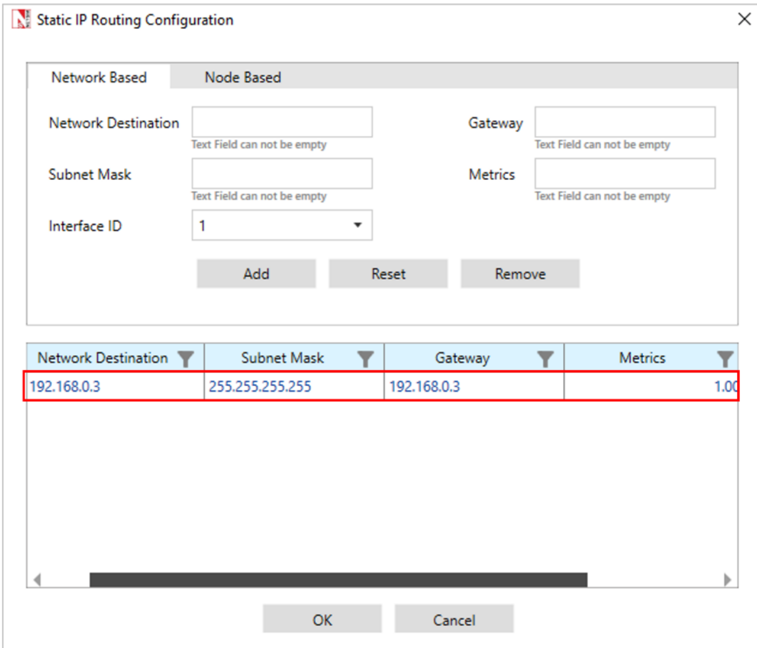


Figure 7-2: Configure Static Routing in Wireless Node 1 via GUI

Step 6: Configure the Wireless link properties as per Table 7-1.

Link_1 Properties	
Channel Characteristics	Pathloss
Pathloss Model	Log distance
Pathloss Exponent (η)	3

Table 7-1: Wireless Link Properties.

Step 7: To configure the Application Traffic, Go to Set Traffic Tab and set the properties as below in Table 7-2.

Application Properties	
Application Method	Unicast
Application Type	CBR
Source ID	1
Destination ID	2
Start Time (s)	5
Packet Size (Bytes)	500
Inter Arrival Time (ms)	20

Table 7-2: Application Properties

Step 8: Enable the IEEE 802.11 Radio Measurements Log in NetSim GUI from Configure Reports Tab → Logs Section as shown in Figure 7-3.

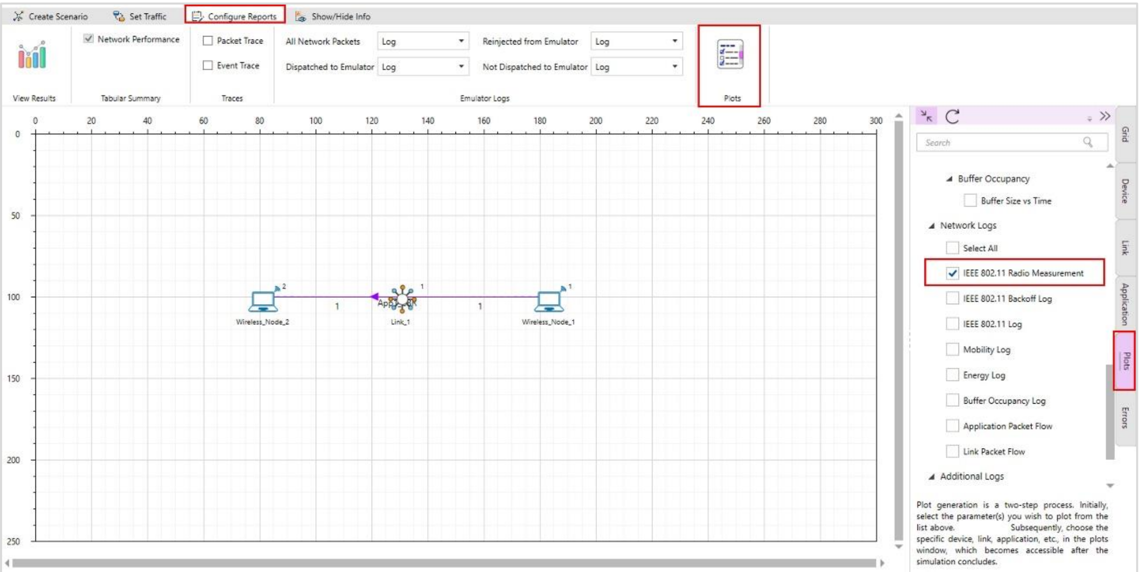


Figure 7-3: Enabling Radio Measurement Log from GUI

Step 9: Run the simulation for 10 seconds and for each simulation use different seed values.

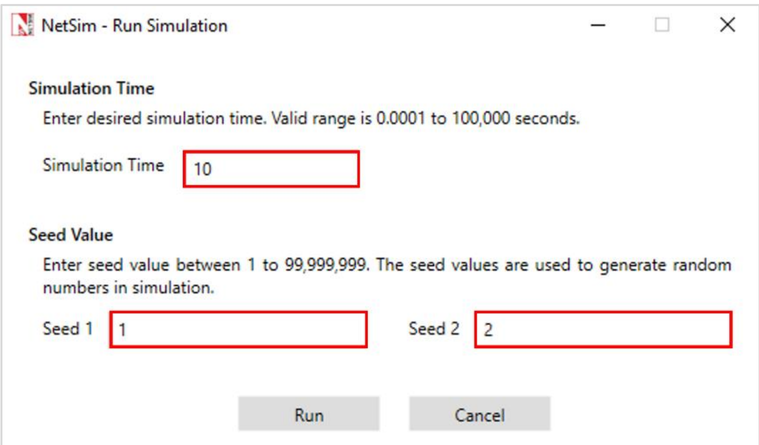


Figure 7-4: Run simulation window shown with the seed values.

Step 10: For each simulation vary the distance between the Transmitter that is Wireless Node 1 and Receiver that is Wireless Node 2 as 1, 2, 5, 10, 20, 50, 100, 200 (in meters).

Step 11: Note Down the Pathloss (dB), Receiver Power (dBm) from Radio Measurement log file present under log section in result dashboard.

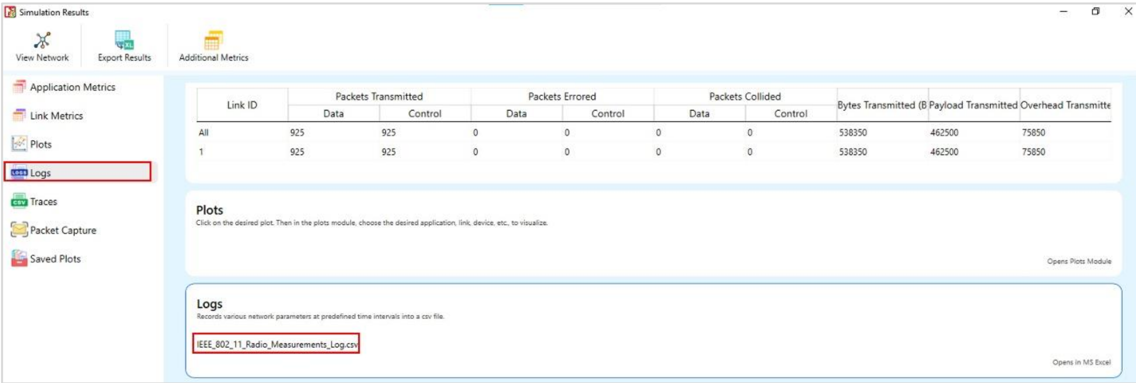


Figure 7-5: Radio Measurement Log under Logs section in Result Dashboard.

Pathloss and Shadowing

Consider Previous Scenario.

Step 1: Configure the Wireless link properties as per Table 7-3.

Link_1 Properties	
Channel Characteristics	Pathloss and Shadowing
Pathloss Model	Log distance
Pathloss Exponent (η)	3
Shadowing Model	Log normal
Standard Deviation (dB)	5

Table 7-3: Wireless Link Properties

Step 2: Run the simulation for 10 seconds and for each simulation use different seed values.

Step 3: For each simulation vary the distance between the Transmitter that is Wireless Node 1 and Receiver that is Wireless Node 2 as 1, 2, 5, 10, 20, 50, 100, 200 (in meters).

Step 4: Note Down the Pathloss (dB), Receiver Power (dBm), Shadowing Loss (dB) and Total Loss (dB) from Radio Measurement log file present under log section in result dashboard.

7.1.1.2 Results and Discussion

Pathloss

Distance (m)	RNG Seed 1	RNG Seed 2	TX Power (dBm)	Path Loss (dB)	RX Power (dBm)
1	1	2	20	40.09	-20.09

2	1	2	20	49.12	-29.12
5	1	2	20	61.06	-41.06
10	1	2	20	70.09	-50.09
20	1	2	20	79.12	-59.12
50	1	2	20	91.06	-71.06
100	1	2	20	100.09	-80.09
200	1	2	20	109.12	-89.12

Table 7-4: Results of NetSim simulations showing Pathloss and Received power vs. Distance with different seed values. Note that NetSim uses two input seeds for its random number generator.

Pathloss and Shadowing

Distance, d(m)	RNG Seed 1	RNG Seed 2	TX Power (dBm)	Path Loss (dB)	Shadowing Loss (dB)	Total Loss (dB)	RX Power (dBm)
1	1	2	20	40.09	-1.44	38.64	-18.64
2	2	3	20	49.12	-1.44	47.67	-27.67
5	3	4	20	61.06	-1.44	59.61	-39.61
10	4	5	20	70.09	-1.44	68.64	-48.64
20	5	6	20	79.12	-1.44	77.67	-57.67
50	6	7	20	91.06	-1.44	89.61	-69.61
100	7	8	20	100.09	-1.44	98.64	-78.64
200	8	9	20	109.12	-1.44	107.67	-87.67

Table 7-5: Results of NetSim simulations showing Pathloss, Shadowing loss and Received Power vs Distance with different seed values.

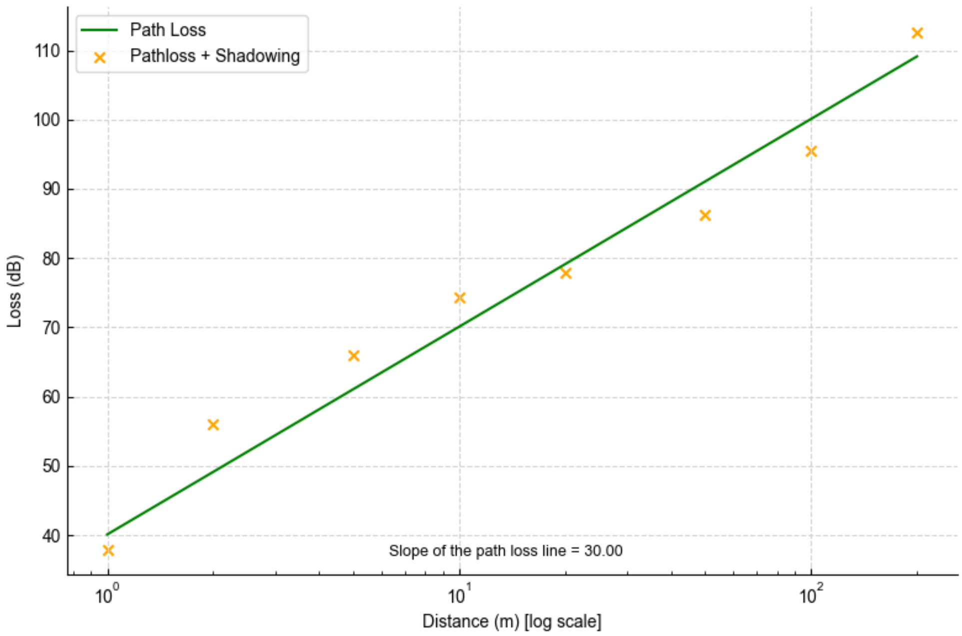


Figure 7-6: Simulation results of Loss (dB) vs. Distance (m) for Pathloss and for Pathloss with shadowing. The X-axis is a log scale. We can also easily calculate that the slope of the pathloss (green) line is 30 which is the value of $10 \times \eta$.

On comparing Table 7-4 and Table 7-5, we can observe identical pathloss values at equal distances. This shows that the pathloss is deterministic, showing no dependency on the seed used in the random number generator. Recall that $PathLoss = PL_{d0} + 10 \cdot \eta \cdot \log_{10}(d)$, implying that pathloss plots as a linear line against the logarithm of distance. This is exactly what we observe in the above figure. Furthermore, the slope of this pathloss line turns out to be 30 which is the value of $10 \times \eta$ configured in the simulation. Both of the above observations are consistent with the log-distance pathloss model. On the other hand, the shadowing loss behaves as a random variable.

Let us theoretically calculate the value of pathloss at a distance of 100m. We know that $\lambda = \frac{c}{f}$, where c, the speed of light is taken as $3 \cdot 10^8\, ms^{-1}$ and the frequency, f is 2.412 GHz.

$$PL(100m) = 20 \cdot log_{10}\left(\frac{4 \cdot \pi \cdot 1}{\left(\left(\frac{3 \cdot 10^8}{2.412 \cdot 10^9}\right)\right)}\right) + 10 \cdot \eta \cdot \log(100)$$

$$PL(100m) = 40.09 + 60 = 100.09\, dB$$

This agrees with the result shown in Table 7-5.

7.1.2 Part 2: Rayleigh Fading

Introduction

Small-scale fading or simply *fading,* is used to describe the rapid fluctuation of the amplitude of a radio signal over a short period of time. Fading is caused by interference between two or more versions of the transmitted signal which arrive at the receiver at slightly different times. These waves called *multipath waves,* combine at the receiver antenna to give a resultant signal which can vary widely in amplitude and phase depending on the distribution of the intensity and relative propagation time of the waves. [4]

Rayleigh fading is widely used to model rapid fluctuations in signal strength due to multipath propagation in wireless networks. In part 1, we saw the impact of two terms pathloss (PL) and shadowing (S) in signal attenuation. The third factor, Rayleigh fading, denoted as R^2 has a probability density function (PDF) given by

$$f(x) = \frac{x}{\sigma^2} e^{-\left(\frac{x^2}{2 \cdot \sigma^2}\right)}$$

The distribution of the amplitude attenuation is Rayleigh; hence this is also called Rayleigh fading.

7.1.2.1 Network Setup

Pathloss

Step 1: Open NetSim and click on Mobile ad hoc Network.

Step 2: Create a scenario with 2 wireless nodes at 100m from each other.

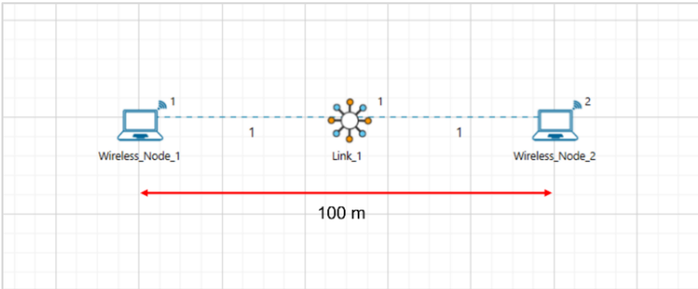


Figure 7-7: A Network consists of 2 Wireless Nodes. Wireless Node 1 and Wireless Node 2 separated by distance $100\ m$.

Step 3: Go to the Properties of Wireless Node 1 → Position Layer → Mobility Layer → Mobility Model set to No Mobility.

Step 4: Go to the Properties of Wireless Node 1 → Network Layer → Enable the static route.

Step 5: Configure static routing in Wireless Node 1 as shown in below figure.

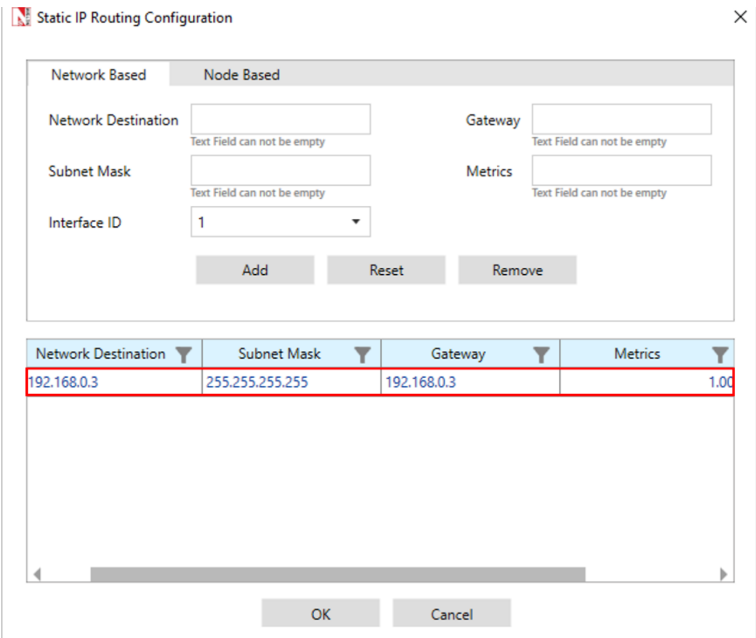


Figure 7-8: Configure Static Routing in GUI

Step 6: Configure the Wireless link properties as per Table 7-6.

Link 1 Properties	
Channel Characteristics	Pathloss
Pathloss Model	Log distance
Pathloss Exponent (η)	3

Table 7-6: Wireless Link Properties

Step 7: To configure the Application Traffic, Go to Set Traffic Tab and set the properties as below Table 7-7.

Application Properties	
Application Method	Unicast
Application Type	CBR
Source ID	1
Destination ID	2
Start Time (s)	0
Packet Size (Bytes)	1460
Inter Arrival Time (μs)	100000

Table 7-7: Application Properties

Step 8: Enable the IEEE 802.11 Radio Measurements Log in NetSim GUI from Configure Reports Tab → Logs Section.

Step 9: Run the simulation for 5 seconds.

Step 10: In the Radio Measurement log file present under log section of the result dashboard, filter packet type to CBR and then note down the Time (μs), Pathloss (dB) and Total Loss(dB).

Pathloss and Fading

Consider the previous scenario.

Step 1: Configure the Wireless link properties as per Table 7-8.

Link_1 Properties	
Channel Characteristics	Pathloss and Fading and Shadowing
Pathloss Model	Log distance
Pathloss Exponent (η)	3
Shadowing Model	None
Fading Model	Rayleigh
Scale Parameter (w)	1

Table 7-8: Wireless Link Properties

Step 2: Run the simulation for 5 seconds.

Step 3: In the Radio Measurement log file present under log section of the result dashboard, filter packet type to CBR and then note down the Time (μs), Pathloss (dB) and Total Loss(dB).

7.1.2.2 Results and Discussion

Time (s)	Pathloss		Pathloss and Fading		
	Path Loss (dB)	Total Loss (dB)	Path Loss (dB)	Fading Loss (dB)	Total Loss (dB)
0.012806	100.09	100.09	100.09	0	100.09
0.112826	100.09	100.09	100.09	-5.08	95.01
0.212546	100.09	100.09	100.09	-2.99	97.1
0.312686	100.09	100.09	100.09	-4.26	95.83
0.412526	100.09	100.09	100.09	-3.49	96.6
0.512506	100.09	100.09	100.09	1	101.09
0.612506	100.09	100.09	100.09	-7.87	92.22
0.712866	100.09	100.09	100.09	0.13	100.22
0.812486	100.09	100.09	100.09	2.36	102.45
0.912506	100.09	100.09	100.09	-11.82	88.27
1.013006	100.09	100.09	100.09	-2.62	97.47
1.112666	100.09	100.09	100.09	-5.57	94.52
1.213006	100.09	100.09	100.09	-7.88	92.21
1.312726	100.09	100.09	100.09	-12.64	87.45
1.412706	100.09	100.09	100.09	-4.49	95.6
1.512686	100.09	100.09	100.09	0.29	100.38
1.612586	100.09	100.09	100.09	5.61	105.7
1.712566	100.09	100.09	100.09	1.17	101.26
1.812866	100.09	100.09	100.09	-8.49	91.6
1.912486	100.09	100.09	100.09	1.84	101.93
2.012766	100.09	100.09	100.09	-9.02	91.07
2.112726	100.09	100.09	100.09	5.53	105.62
2.212766	100.09	100.09	100.09	-3.39	96.7
2.312486	100.09	100.09	100.09	-1.94	98.15
2.412706	100.09	100.09	100.09	1.23	101.32
2.513066	100.09	100.09	100.09	2.95	103.04
2.612886	100.09	100.09	100.09	-15.58	84.51
2.713066	100.09	100.09	100.09	-6.99	93.1
2.812766	100.09	100.09	100.09	5.44	105.53
2.912846	100.09	100.09	100.09	2.55	102.64
3.012486	100.09	100.09	100.09	1.48	101.57
3.112646	100.09	100.09	100.09	-3.41	96.68
3.213046	100.09	100.09	100.09	0.03	100.12
3.312686	100.09	100.09	100.09	-3.26	96.83
3.412486	100.09	100.09	100.09	-0.3	99.79
3.512926	100.09	100.09	100.09	1.4	101.49
3.612566	100.09	100.09	100.09	-5.85	94.24
3.712646	100.09	100.09	100.09	-2.73	97.36
3.813066	100.09	100.09	100.09	-3.53	96.56
3.912986	100.09	100.09	100.09	-14.11	85.98
4.012886	100.09	100.09	100.09	-3.29	96.8
4.112846	100.09	100.09	100.09	-7.14	92.95
4.212506	100.09	100.09	100.09	-2.67	97.42

4.312686	100.09	100.09	100.09	-5.31	94.78
4.412586	100.09	100.09	100.09	6.92	107.01
4.512926	100.09	100.09	100.09	0.45	100.54
4.612686	100.09	100.09	100.09	-8.77	91.32
4.712486	100.09	100.09	100.09	-6.04	94.05
4.812926	100.09	100.09	100.09	-8.63	91.46
4.912686	100.09	100.09	100.09	-0.15	99.94

Table 7-9: Results Table for Wireless Node 1 and Wireless Node 2 100m from each other with pathloss and pathloss and fading.

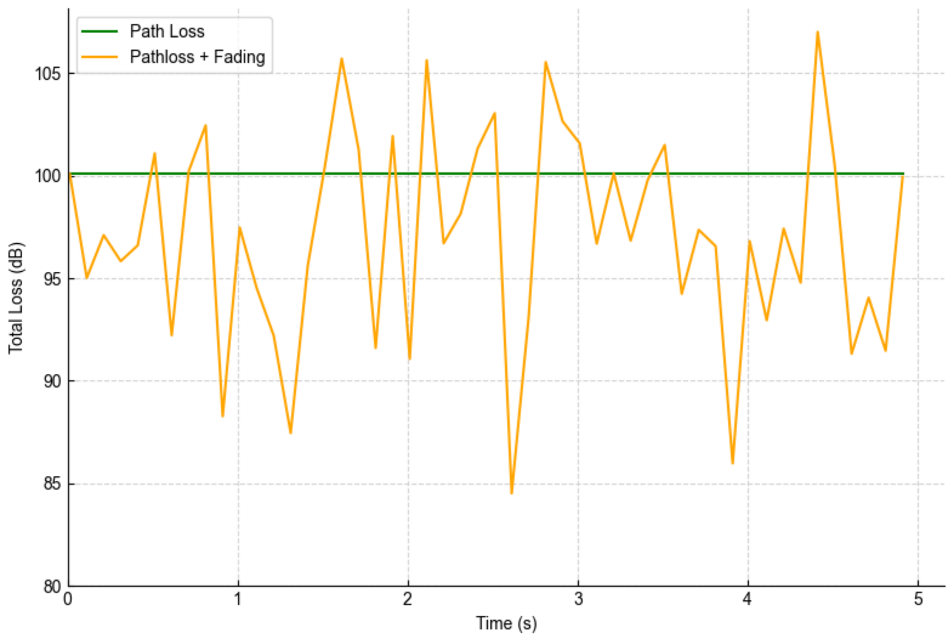


Figure 7-9: Loss in dB vs. Time when running Pathloss and Pathloss with Fading cases.

From the above figure, it is evident that pathloss (green line) remains constant over time. However, when fading is factored in, the total loss (yellow line) exhibits temporal variability. This total loss is composed of two elements: pathloss and fading loss. Among these, pathloss is time-invariant, whereas fading loss fluctuates over time as can be seen from Table 7-9.

7.1.3 Exercises

1. Change the pathloss exponent to any value between 2.0 to 5.0 and obtain the pathloss vs. distance plot similar to Figure 7-6. Explain the difference between your plot and the plot in the experiment.
2. Change the fading scale parameter to any value between 0.25 to 2.00 and combined plots of pathloss vs time and pathloss plus fading vs. time similar to the plot in Figure 7-9. Explain the difference between your plot and the plot in the experiment.

7.1.4 References

[1] T. S. Rappaport, "Wireless Communications," 2002.

[2] D. M. J. K. Anurag Kumar, "Wireless Networking," Morgan Kaufmann, 2008.

8 Mobile Ad hoc Networks

8.1 Connectivity of a randomly deployed 1-D ad hoc network (Level 2)

8.1.1 Objective

To analyze the probability of connectivity of a randomly deployed 1-dimensional ad hoc network.

In the first part of the experiment, we analyze connectivity for a simple 2-node network. In the advanced section, we extend this analysis to n nodes.[5]

8.1.2 Preliminaries

An important feature of wireless networks is that the node locations in a network are random, because of either mobility or deployment constraints. Thus, in exploring the fundamental performance limits of wireless networks, it is reasonable to model the node locations as random variables.

A *network graph* captures the communication capabilities among the nodes in the network and connectivity is an important network graph property. For a wireless network, the network graph also captures the communication constraints; for example, it can be used to specify the nodes that are in communication range of any node.

We assume that a transmission from node i, located at X_i, can be decoded at node j, located at X_j, if the Euclidean distance between X_i and X_j is less than r i.e., $|X_2 - X_1|, \leq r$ in 1-D , or $\sqrt{(Y_2 - Y_1)^2 + (X_2 - X_1)^2} \leq r$, in 2-D.

8.1.3 Mathematical analysis of a 2-node 1-D network

Consider a two-node, one-dimensional network with the location of each node uniformly distributed in $[0, z]$ and chosen independently of each other. Let the transmission range of both nodes be r. We now obtain the probability that the two nodes are connected. Without loss of generality, let X_1 be the location of the left node and X_2 that of the right node; that is, $X_1 \leq X_2$. The two-node network is connected if $X_2 - X_1 \leq r$. This is graphically shown in Figure 8-1. The set of values that (X_1, X_2) can take is denoted by

[5] This experiment is based on Section 9.2 of [1].

the area OAB. The set of (X_1, X_2) that would result in a connected network is given by the shaded area S in the below figure.

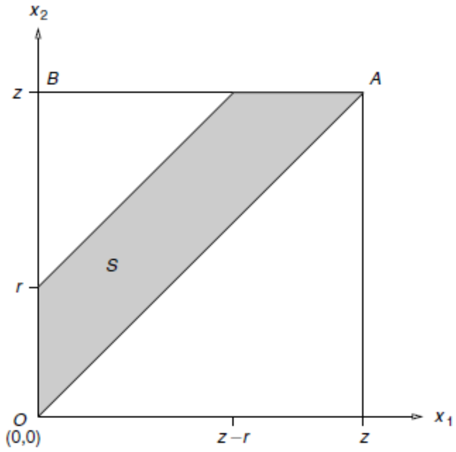


Figure 8-1: S represents the feasible region of a random, connected 2-node, 1-dimensional ad hoc network.

S is the region satisfying $X_1 < X_2$ (by definition of X_1 and X_2) and $X_2 - X_1 \leq r$ (the connectivity requirement). Since the nodes are distributed uniformly in $[0, z]$, the probability that the network is connected is the ratio of the area of S to the area of OAB. The area of S is $\frac{z^2}{2} - \frac{(z-r)^2}{2}$ and that of OAB is $\frac{z^2}{2}$. Thus, the probability that the network is connected is

$$P_c = 2 \cdot \left(\frac{r}{z}\right) - \left(\frac{r}{z}\right)^2$$

where, P_c is the network connectivity fraction or the probability of network connectivity, and

$\frac{r}{z}$ is the normalized transmission range.

8.1.4 Modelling the transmission range

NetSim supports are wide range of pathloss models. In this experiment, we need to use a pathloss model whereby the transmission range of a node is r. This can be modelled using the *Range based pathloss* model in NetSim. In this model, the propagation loss depends only on the distance (range) between transmitter and receiver. There is a single *Range* attribute that determines the path loss. Receivers located at or within 'Range' see a 0 dB pathloss. Hence received power equals transmit power. Receivers beyond 'Range' see a 1000 dB pathloss. Hence received power will be close to -1000 dBm which is essentially zero in linear units.

8.1.5 Procedure to simulate this scenario in NetSim

1. In NetSim home window click on MANET section. Under the Grid settings, set Grid Width as 1000m and Grid length as 500m and under device placement strategy select Manually via Click and Drop.

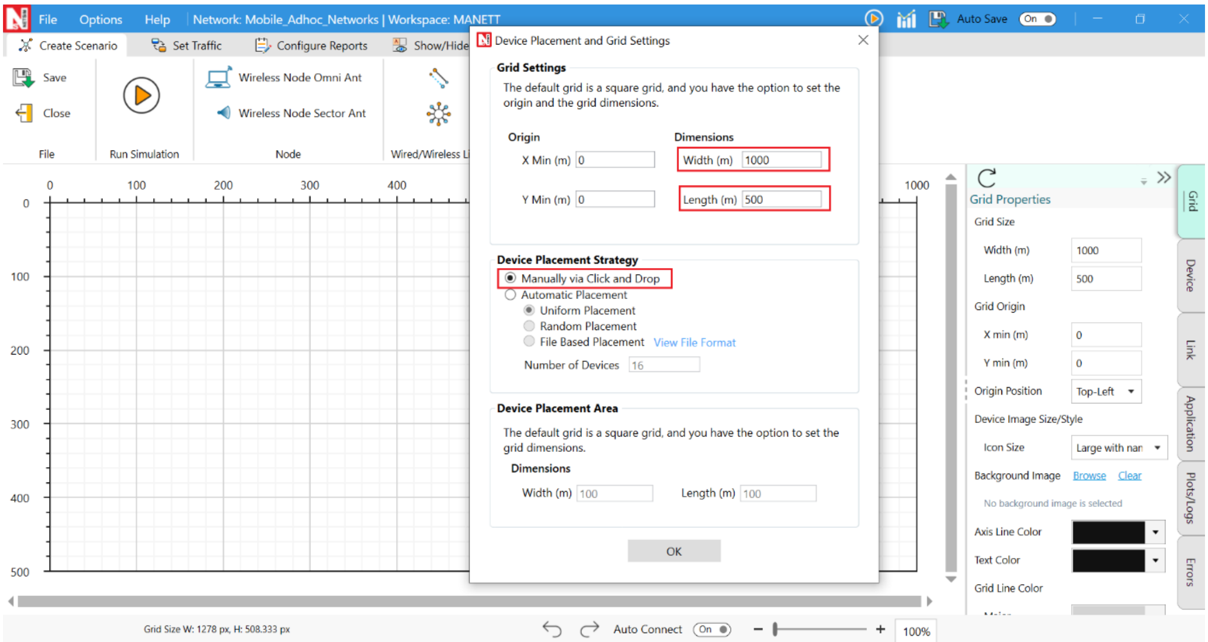


Figure 8-2: Select the 'Manually via click and drop' option in grid setting window.

2. Deploy two wireless nodes. Set the mobility of both devices to No mobility, and position both wireless nodes at Y coordinate 250. The X coordinate can be any value at this stage. Since the X coordinate is a variable for this experiment, the exact setting is explained subsequently.

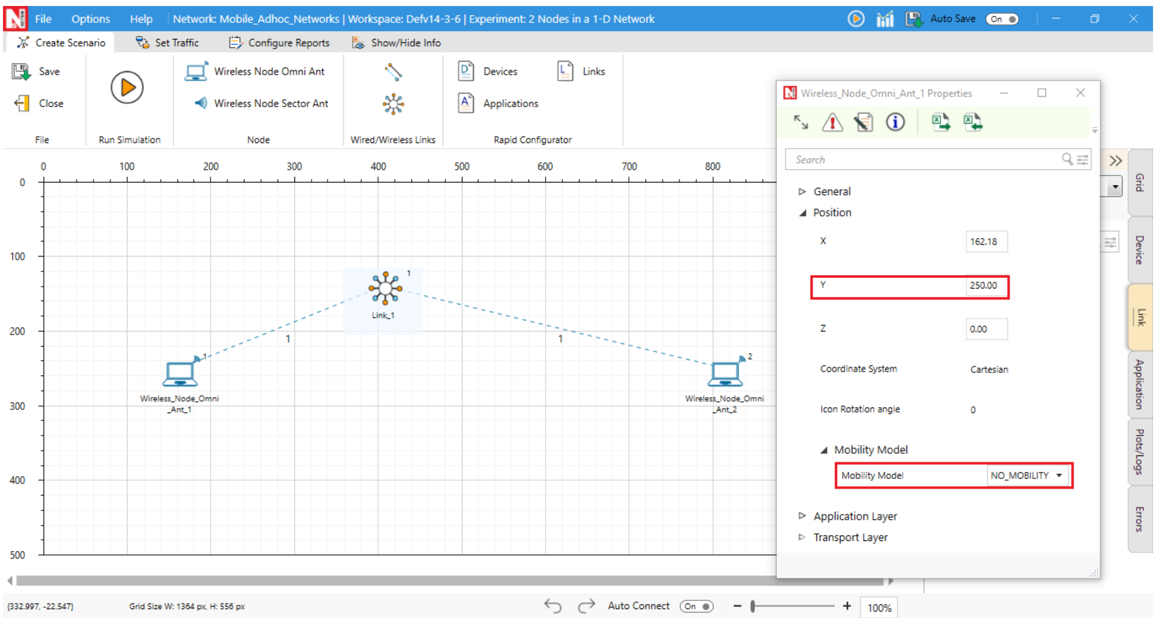


Figure 8-3: Set mobility model as No mobility

3. Configure a CBR application to communicate between the wireless nodes. Let the packet size be default, set the start time as 0 and Inter Arrival Time as 600,000.

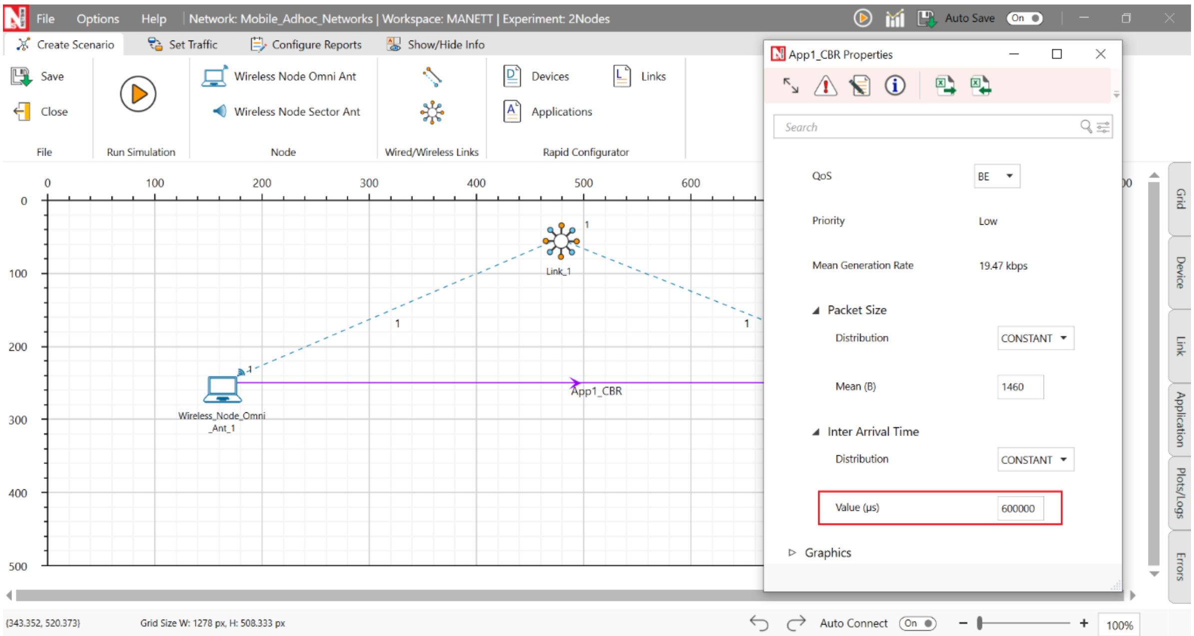


Figure 8-4: Set IAT in application settings

4. Set Channel Characteristics as pathloss and Path Loss Model as Range based. The range (m) can be any value currently. This is another variable in this experiment and the exact settings are explained subsequently.

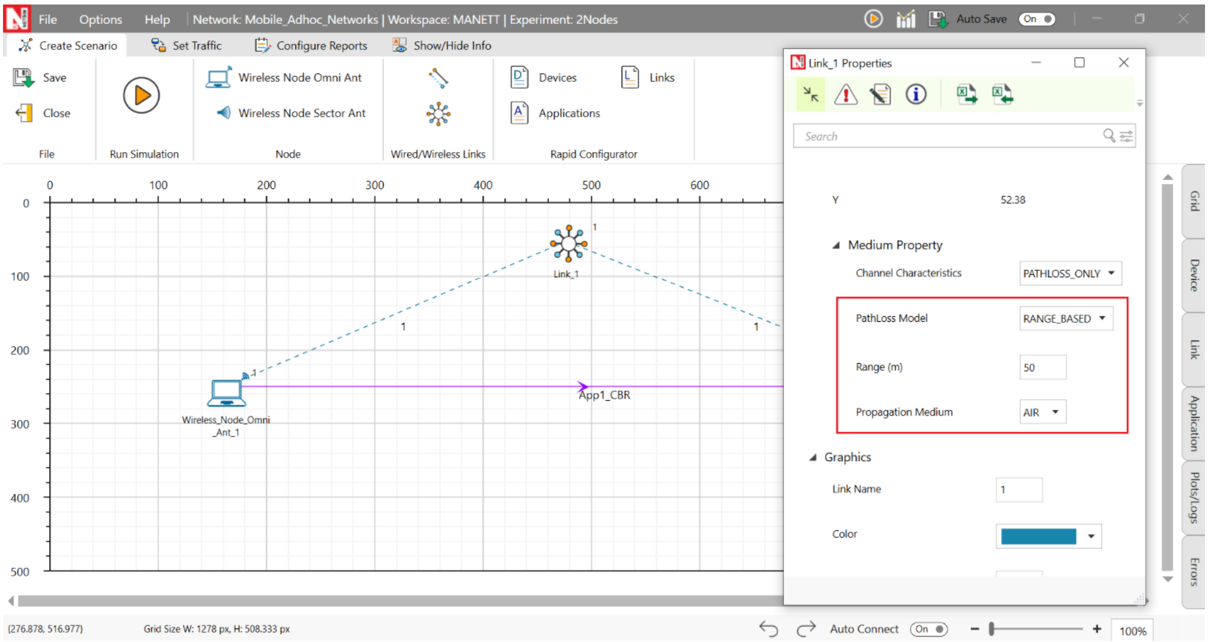


Figure 8-5: Channel characteristics properties

5. Add Static Route from Wireless Node 1 to Wireless Node 2. In Wireless Node 1 property, go to Network layer and make Static IP Route enable, then click on via GUI. The static routing setup neglects guiding control packets around it and doesn't

account for RTS/CTS thresholds. This approach might disrupt data flow and affect collision management during data transmission in the network.

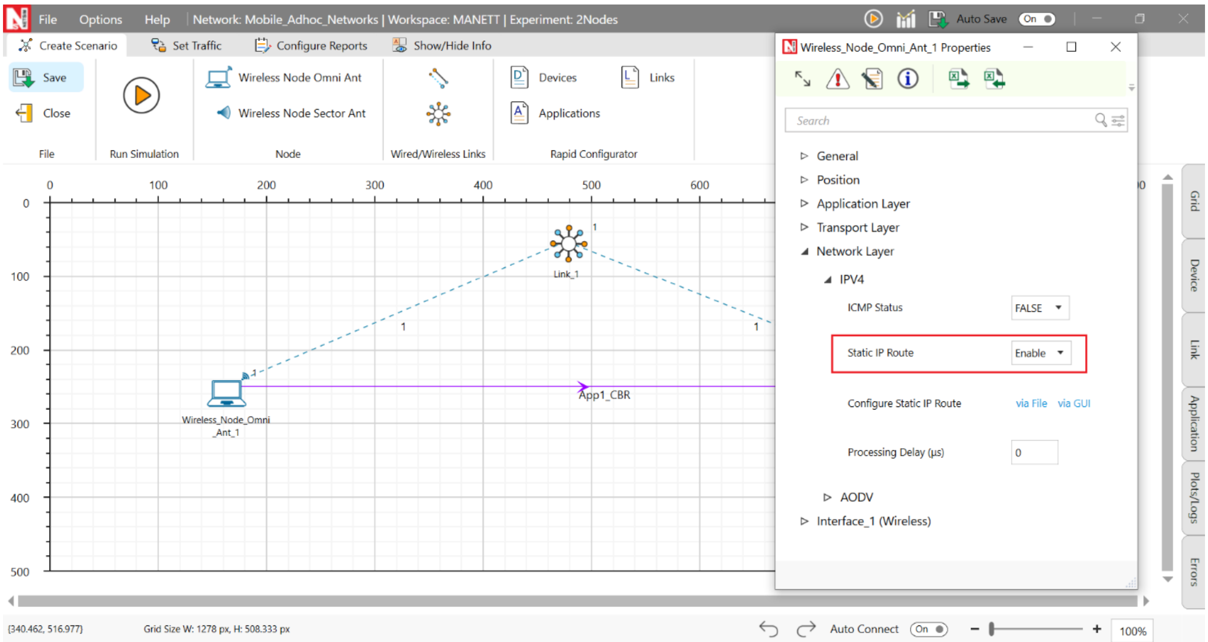


Figure 8-6: Enabling static route in wireless node 1.

- In the Static IP Routing Configuration Window, add destination IP address (enable Device IP in Show/Hide Info), gateway, subnet mask, metrics, interface id and click on Add.

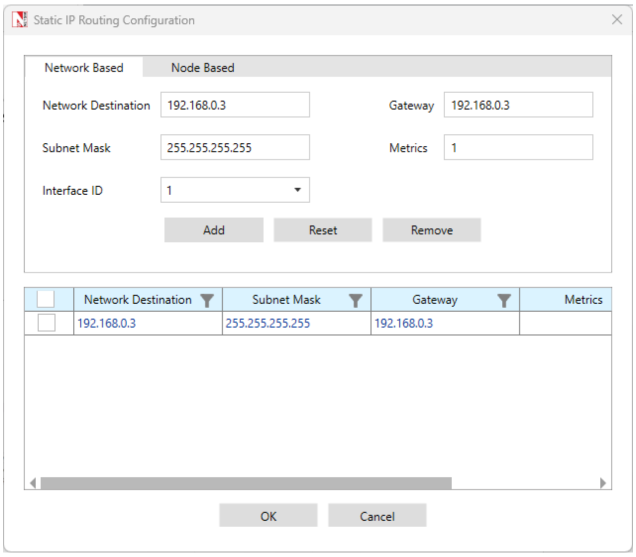


Figure 8-7: Static Route Configuration

6. Save this scenario and open the experiment in the file explorer and open Configuration.netsim in Visual Studio.

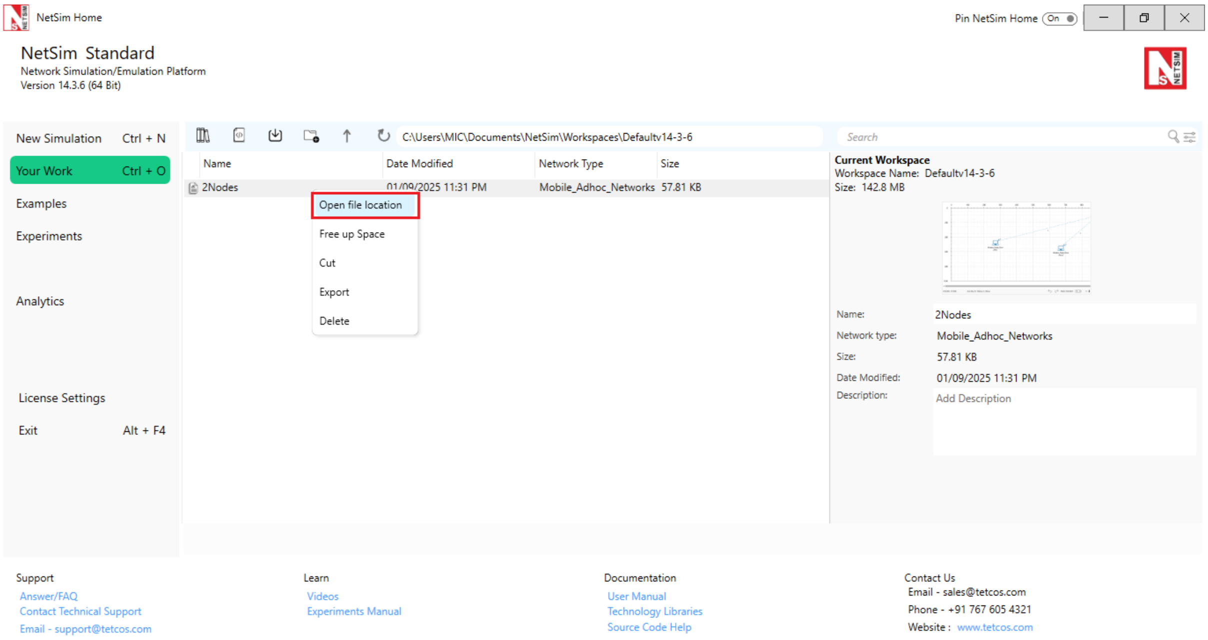


Figure 8-8: Opening saved file location from 'Your Work' window

7. Within the POS 3D tag of both wireless nodes, replace the current X coordinates with a variable {n}, where n = 0, 1, 2, 3, and so on, representing an input variable from the multi-parameter sweeper. Similar procedure is repeated for pathloss Range. Set the simulation time as 0.5.

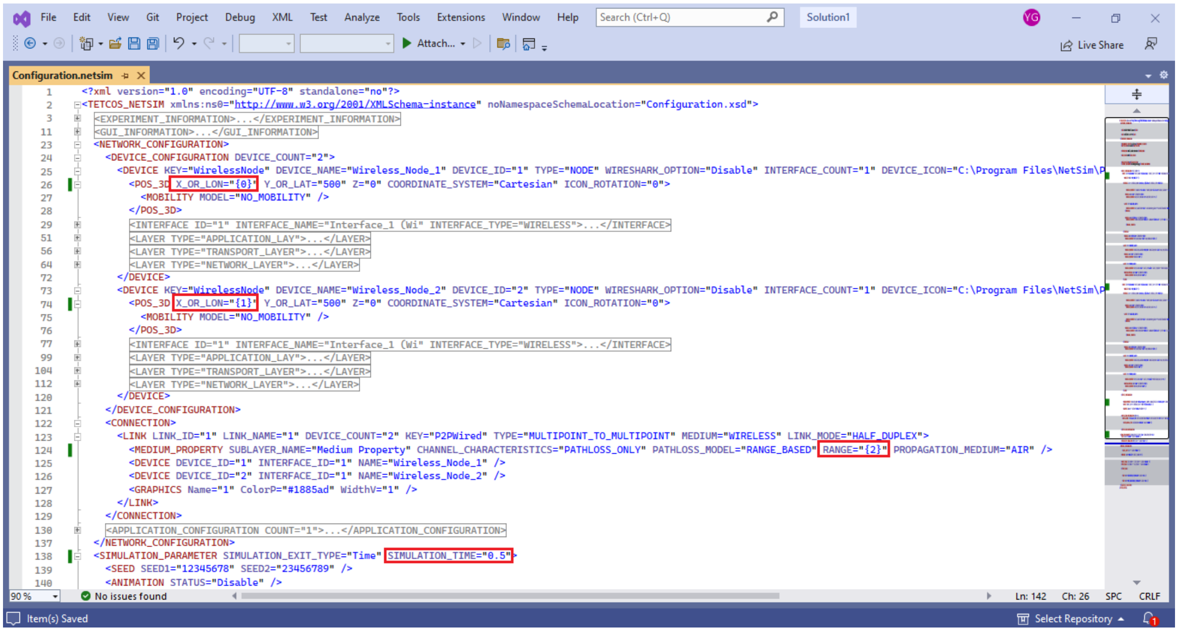


Figure 8-9: input variables for Multi-Parameter Sweeper

8. Save the configuration file and rename it as input.xml.
9. Download the multi-parameter sweeper from the given link https://github.com/NetSim-TETCOS/Connectivity-of-1D-ad-hoc-Networkv14.4/archive/refs/heads/main.zip
10. Paste input.xml and Config support folder into the 2Nodes folder.

11. Change the NetSim Path (line #2) to the current workspace bin_x64 path.

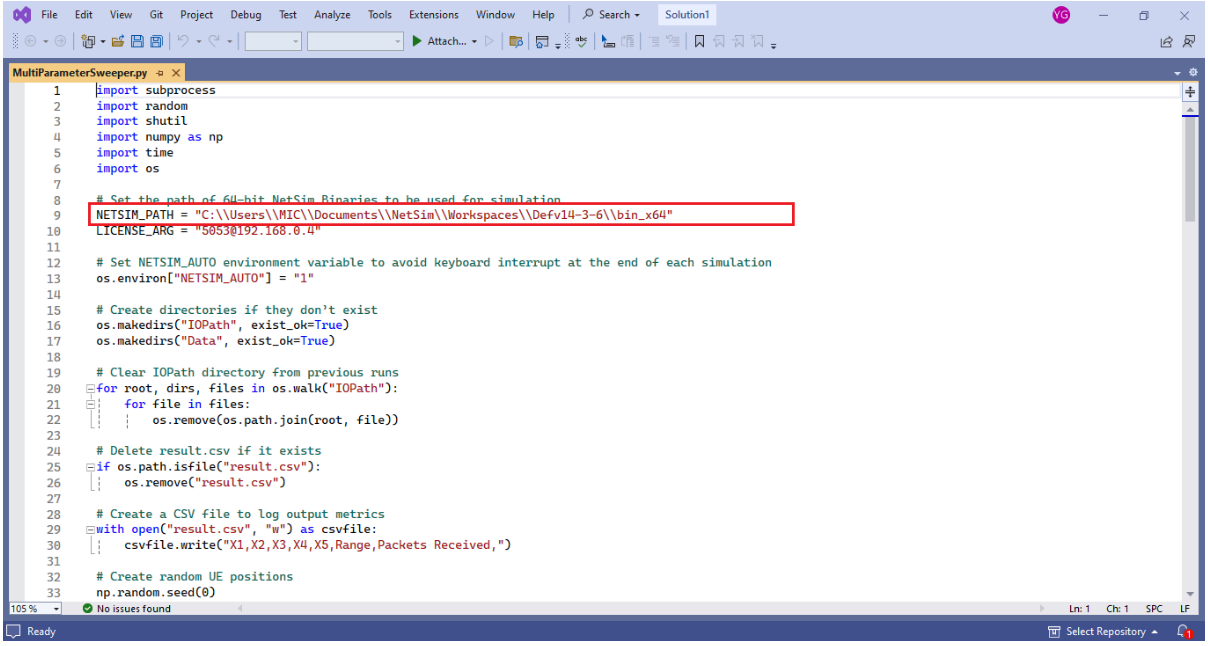
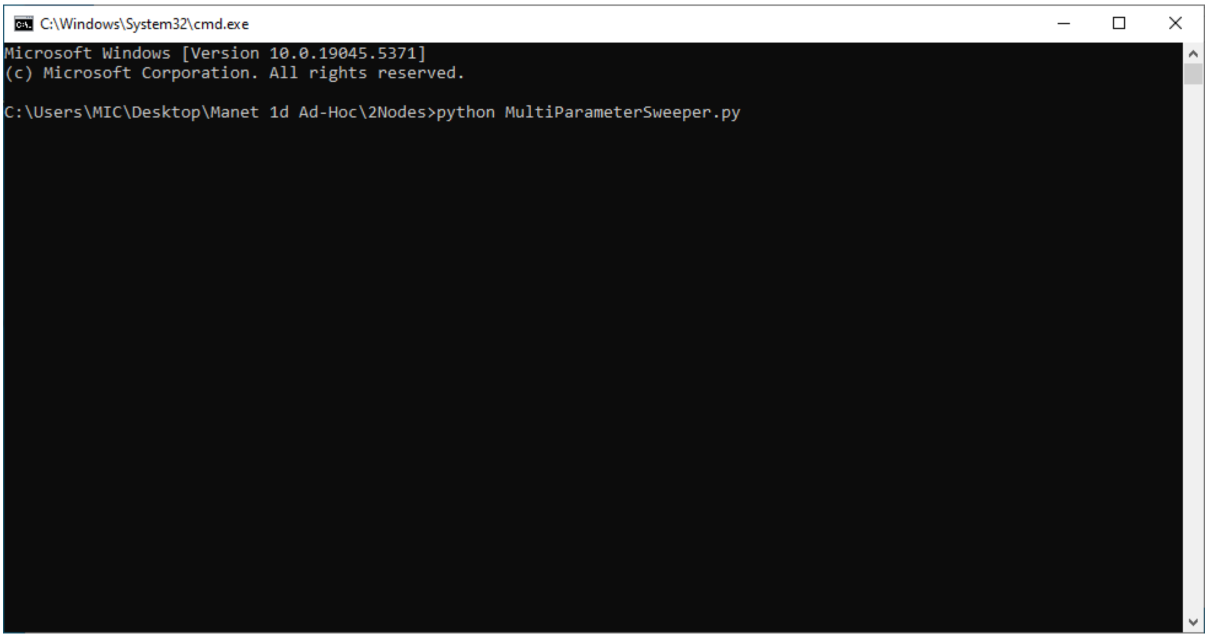


Figure 8-10: MultiParameterSweeper.py opened in the Visual Studio editor window.

12. Run MultiParameterSweeper.py using command prompt.



13. The multi-parameter sweeper runs 2000 simulations, varying X-coordinates between nodes and all transmission range values. It generates an output file named "result.csv" to store the collected data.

8.1.6 Procedure to obtain the number of time network is connected from results.

1. Open the results.csv file and in the toolbar's insert section, insert a table to the current section.

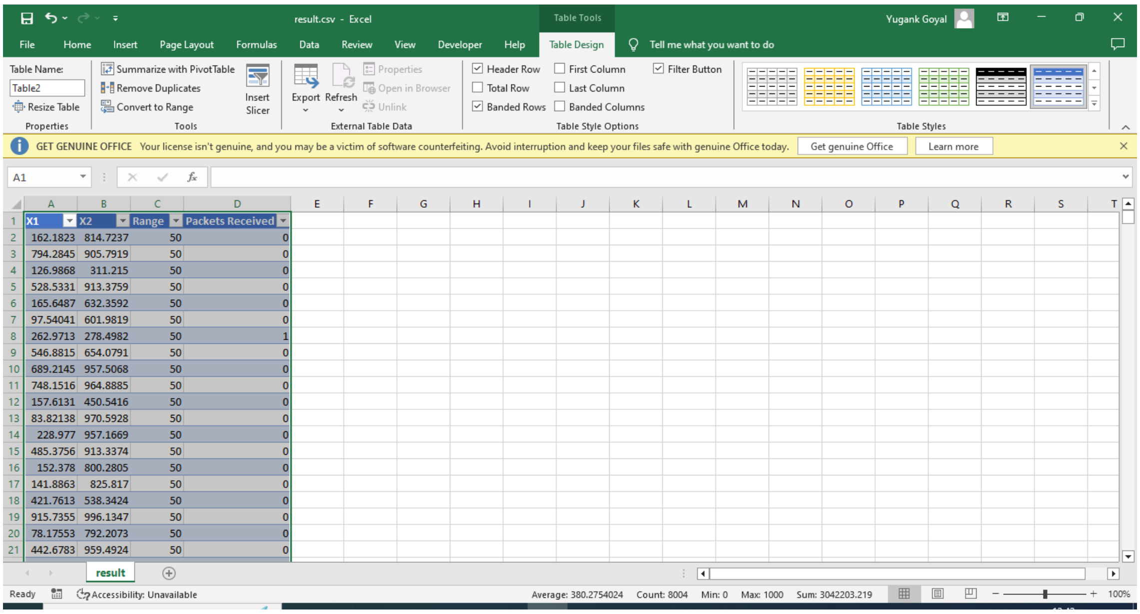


Figure 8-11: Opening Results.csv file

2. In the toolbar's insert section, insert a pivot table for the current table.

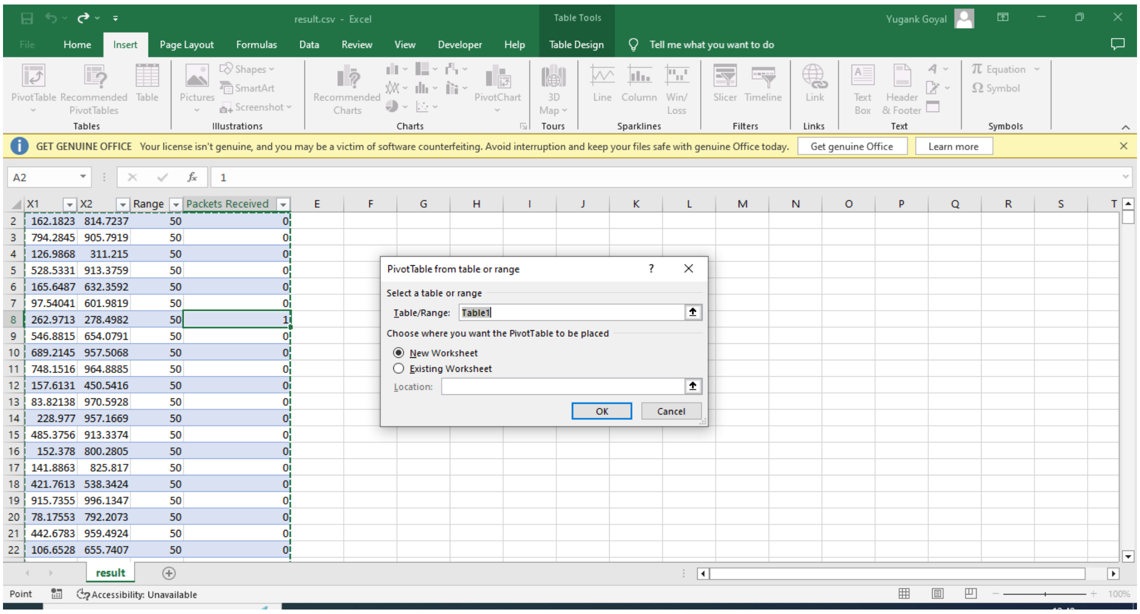


Figure 8-12: Inserting Pivot Table

3. In the pivot table add Range in Rows and Packets Received in Filters and Values. In the multi-parameter-sweeper, one packet is sent per simulation. A successful transmission is when the received packet is marked as 1, and unsuccessful if

marked as 0. After creating the pivot table, it reflects the total occurrences of successful transmissions per range by summing up the received packets.

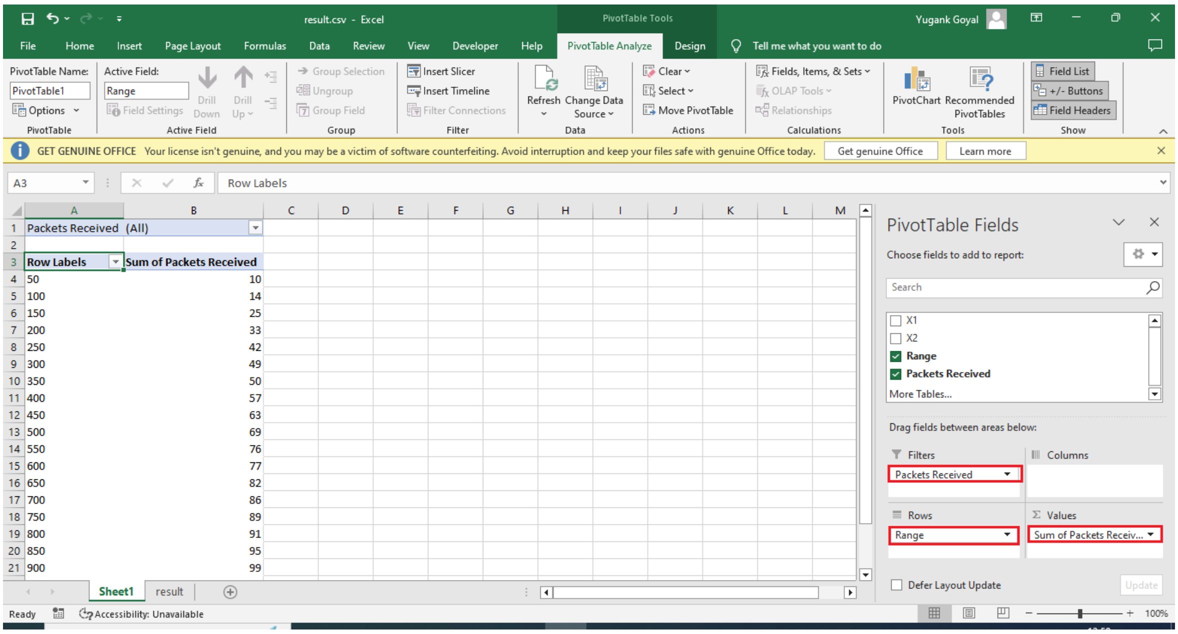


Figure 8-13: Pivot table settings

8.1.7 Simulation Results

Use the result.csv file to get the successful transmission probability table and its plot. From the principles of Monte Carlo simulations, the fraction of times the network is connected (from simulations) is nothing but the probability of network connectivity (from theory).

In the following table, the sum of packets obtained from the above pivot table is nothing but the count of times network is connected. The probability of network connectivity (analysis) is obtained using the formula $P_c = 2 \cdot \left(\frac{r}{z}\right) - \left(\frac{r}{z}\right)^2$

Transmission Range (r) [m]	Normalized transmission range (r/z) where $z = 1000$ [m]	Count of times network is connected (Total of 100 runs)	Network connectivity fraction (simulation)	Probability of network connectivity (analysis)
50	0.05	9	0.09	0.097
100	0.10	16	0.16	0.190
150	0.15	23	0.23	0.277
200	0.20	34	0.34	0.360
250	0.25	42	0.42	0.437
300	0.30	47	0.47	0.510
350	0.35	56	0.56	0.577
400	0.40	63	0.63	0.640
450	0.45	68	0.68	0.697
500	0.50	73	0.73	0.750
550	0.55	78	0.78	0.797

600	0.60	85	0.85	0.840
650	0.65	87	0.87	0.877
700	0.70	88	0.88	0.910
750	0.75	94	0.94	0.937
800	0.80	95	0.95	0.960
850	0.85	99	0.99	0.977
900	0.90	100	1.00	0.990
950	0.95	100	1.00	0.997
1000	1.00	100	1.00	1.000

Table 8-1: Results of NetSim simulation and analysis.

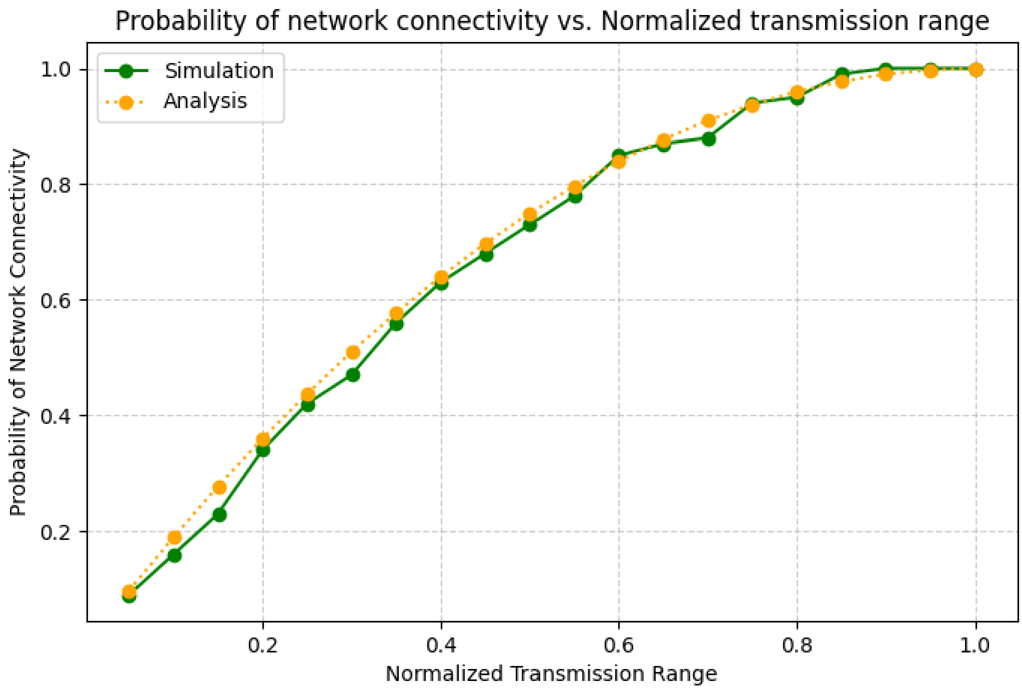


Figure 8-14: Plot comparing simulation results against analytical model for probability of network connectivity for different normalized transmission range values.

8.1.8 Advanced Topic: End-to-end connectivity of a network with n Nodes

Let there be n nodes in the network and let the location of node i be denoted by X_i. X_i are i.i.d. with uniform distribution in $[0, z]$. Thus, the random network is represented by a random vector $X = [X_1, X_2, \ldots, X_n]$. Let $p_c(n, z, r)$ be the probability that **X** represents a connected network when each node has a transmission range of *r*. Let $\hat{X} = [\hat{X}_1, \hat{X}_2, \ldots \ldots \hat{X}_n]$ be the node locations ordered according to their positions on $[0, z]$; that is, $\hat{X}_1 < \hat{X}_2 < \hat{X}_3 < \ldots < \hat{X}_n$. Define $\hat{X}_0 = 0$. The condition $\hat{X}_{i+1} - \hat{X}_i < r$ for $i = 1, \ldots, (n-1)$ needs to be satisfied for **X** to represent a connected network.

The derivation of the closed form analytical equation for $p_c(n, z, r)$ is provided in pages 299 – 301 of [6]. It finally turns out that the probability that the network is connected it

$$p_c(n,z,r) = \frac{U_c(n,z,r)}{U(n,z)} = \sum_{k=0}^{n-1} \binom{n-1}{k} (-1)^k \frac{(z-kr)^n}{z^n} u(z-kr)$$

where $u(z)$ is the unit step function i.e., $u(z) = 0$ when $z \leq 0$ and $u(z) = 1$ when $z > 0$.

8.1.9 Procedure to simulate the n node 1-D scenarios in NetSim

We conduct simulation experiments in NetSim for $n = 5, 10,$ and 20, and the procedure for 5-nodes is explained below. Follow similar steps for $n = 10$, and 20.

1. In NetSim home window click on MANET section. Under the Grid settings, set Grid Length as 1000 and under device placement strategy select Manually via Click and Drop.

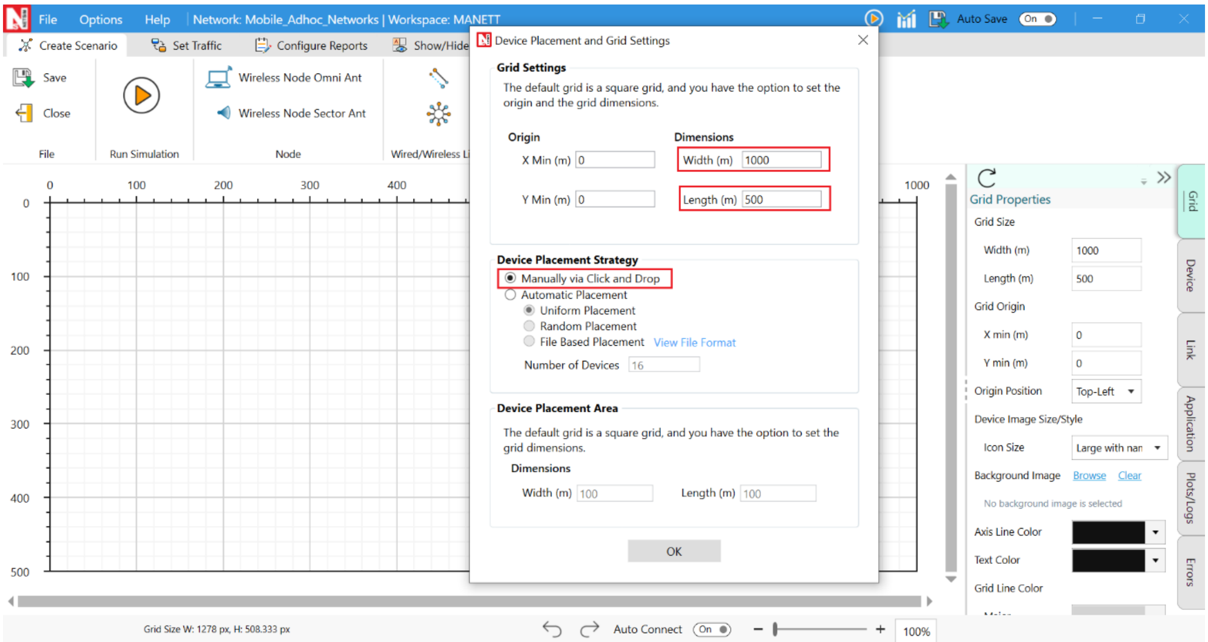


Figure 8-15: Select the ‘Manually via click and drop’ option in grid setting window.

2. Deploy five wireless nodes. Set the mobility of all five devices to NO MOBILITY, and position of all five wireless nodes at Y coordinate 250.

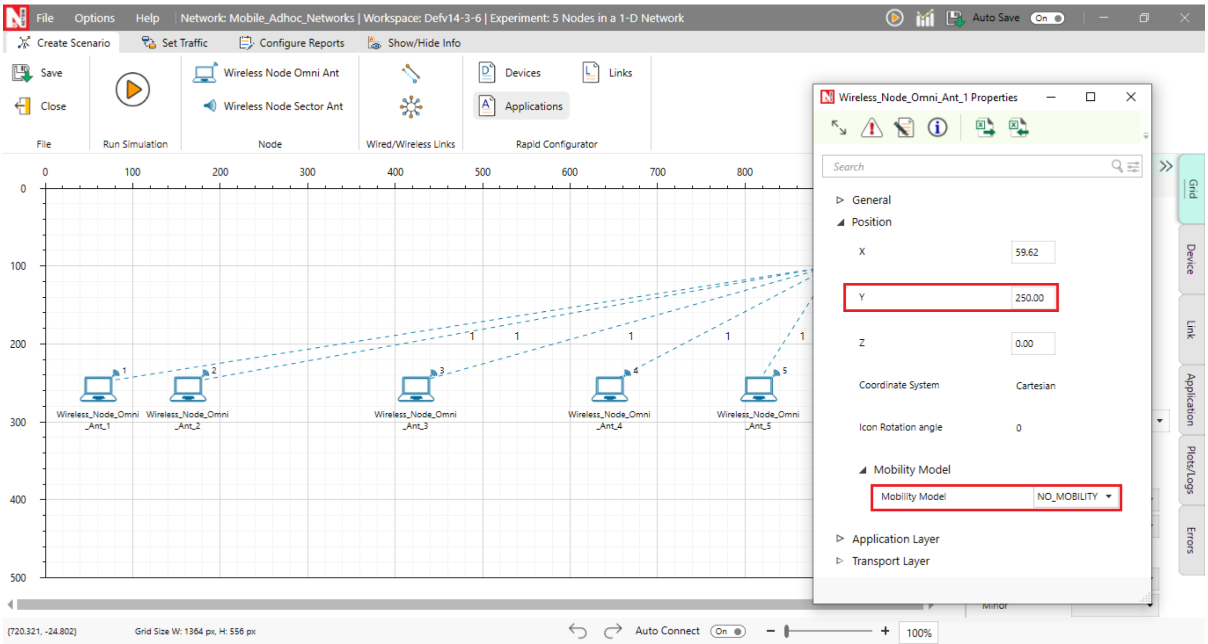


Figure 8-16: Device Placement and Mobility Settings

3. Configure a CBR application to communicate between the wireless nodes from Wireless node 1 to wireless node N, i.e node 1 to node 5. Let the packet size be default, set the start time as 0 and Inter Arrival Time as 600,000.

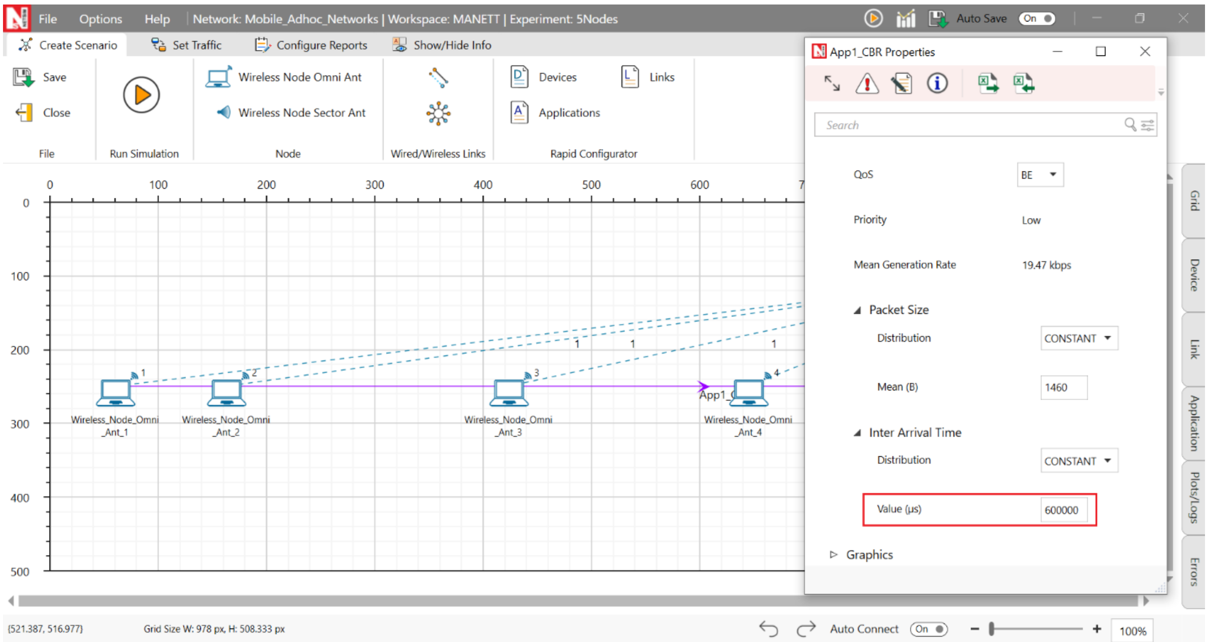


Figure 8-17: Set IAT in Application Settings

4. Set Channel characteristics as Pathloss and Pathloss model as Range based.

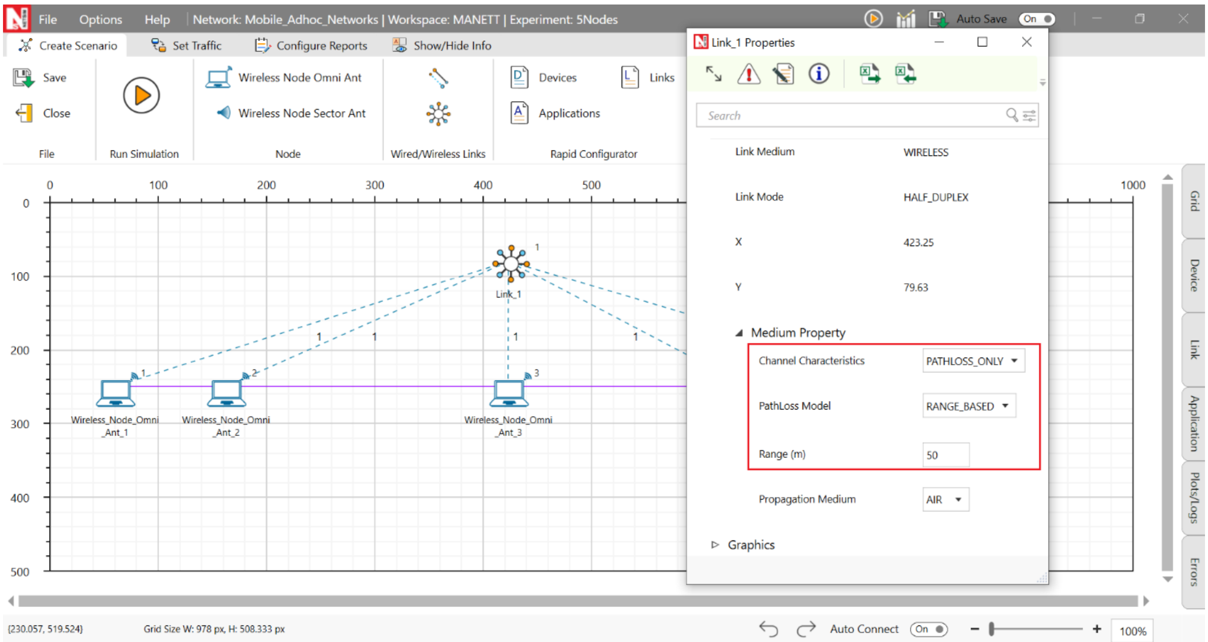


Figure 8-18: Set channel characteristics properties.

5. Add Static route from wireless node 1 to Wireless Node 5. The static routing setup neglects guiding control packets around it and doesn't account for RTS/CTS thresholds. This approach might disrupt data flow and affect collision management during data transmission in the network. In Wireless Node 1 properties, go to Network layer and make Static IP route enable, then click on via GUI.

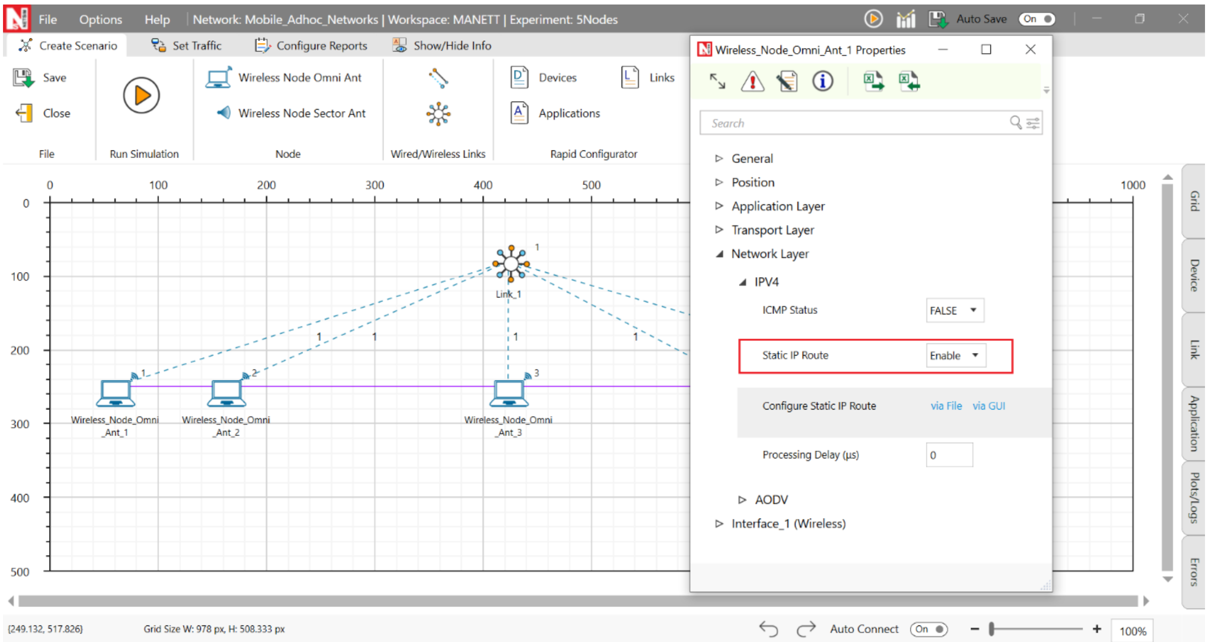


Figure 8-19: Network Layer Settings

In the Static IP Routing Configuration Window, add destination IP address (it should be to the next Wireless Node i.e., 1>2, 2>3, … and so on), gateway, subnet mask, metrics, interface id. Click on Add to add the static route.

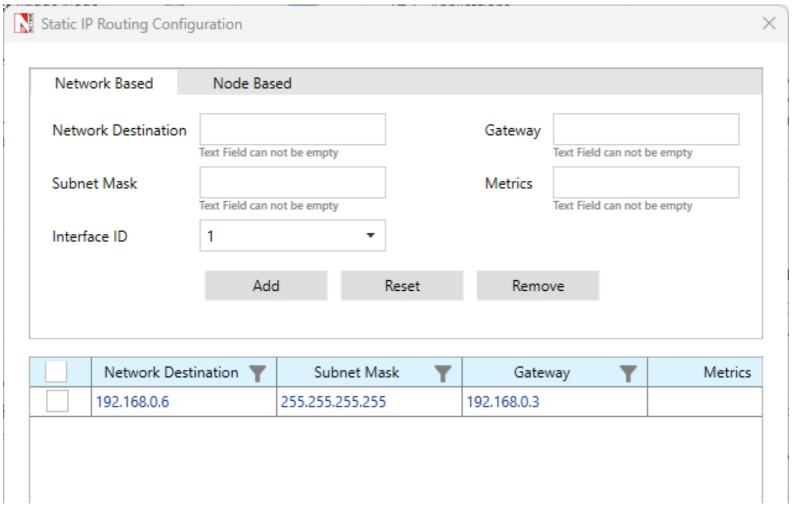


Figure 8-20: Static IP Routing Configuration Window

Wireless Node	Network Destination	Subnet Mask	Gateway	Metrics	Interface
Wireless Node 1	192.168.0.6	255.255.255.255	192.168.0.3	1	1
Wireless Node 2	192.168.0.6	255.255.255.255	192.168.0.4	1	1
Wireless Node 3	192.168.0.6	255.255.255.255	192.168.0.5	1	1
Wireless Node 4	192.168.0.6	255.255.255.255	192.168.0.6	1	1

Table 8-2: Static route configurations for wireless nodes.

6. Save this scenario and open the experiment in the file explorer and open Configuration.netsim in Visual Studio.

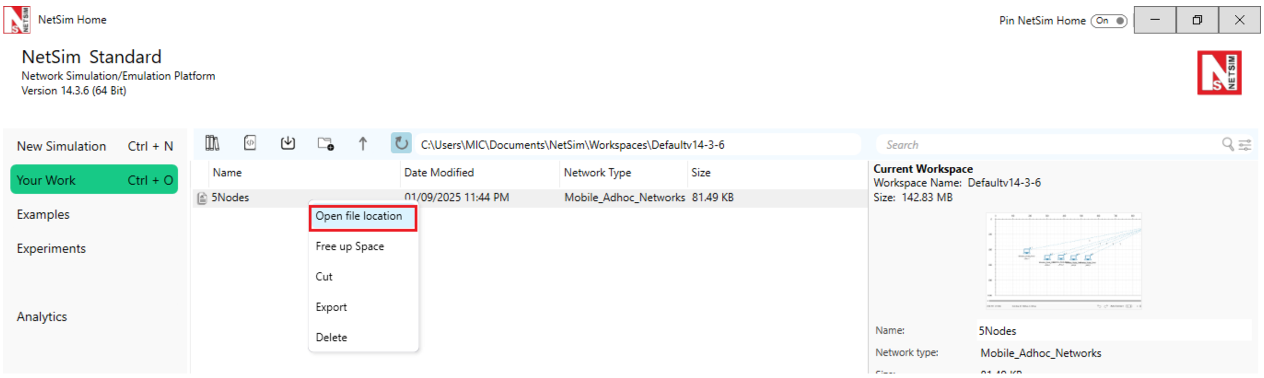


Figure 8-21: Your Work window

7. Within the POS 3D tag of all the wireless nodes, replace the current X coordinates with a variable {n}, where n = 0, 1, 2, 3, and so on, representing an input variable from the multi-parameter sweeper. Similar procedure is repeated for Pathloss range. Set the simulation time as 0.5.

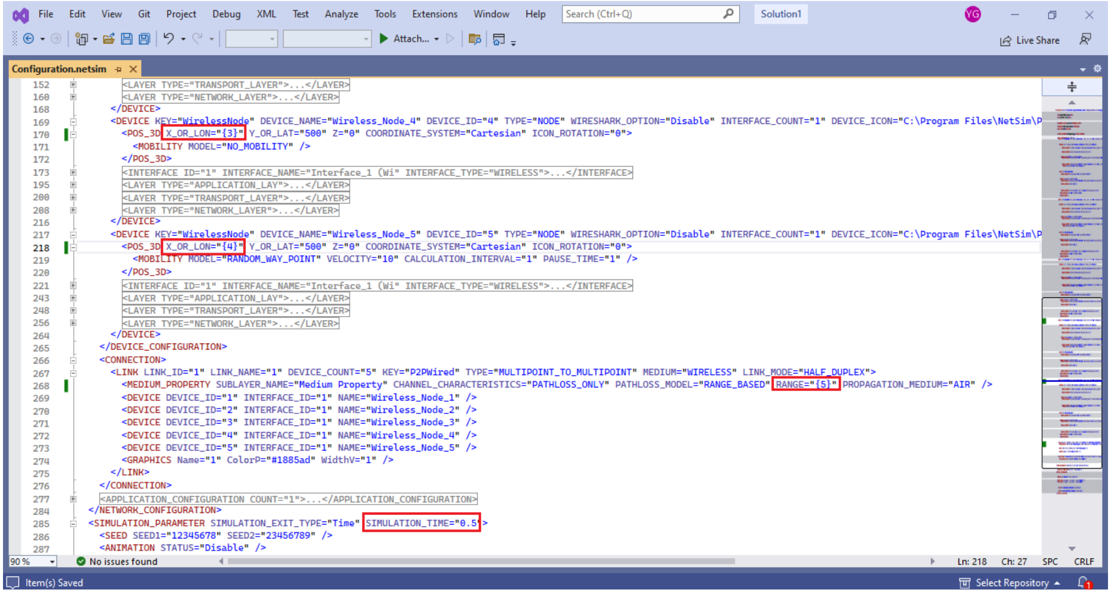


Figure 8-22: Configuration file changes for Multi-Parameter Sweeper

8. Save the configuration file and rename it as input.xml.
9. Download the multi-parameter sweeper from the given link https://github.com/NetSim-TETCOS/Connectivity-of-1D-ad-hoc-Networkv14.4/archive/refs/heads/main.zip
10. Paste input.xml, and Config support folder into 5Nodes folder.
11. change the NETSIM_PATH (line #2) to the current workspace bin_x64 path.

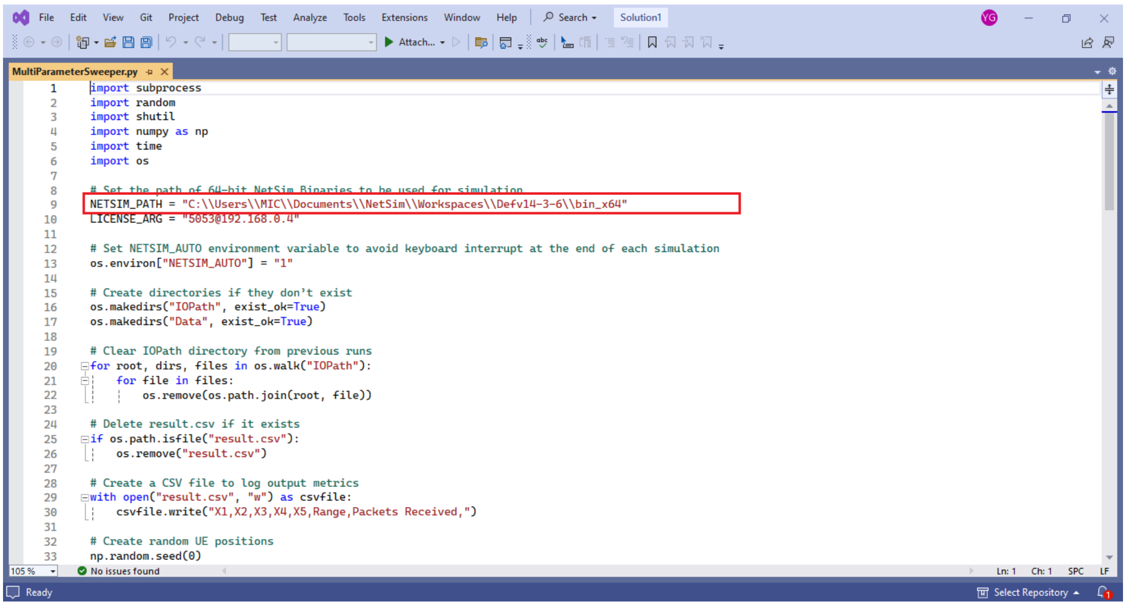


Figure 8-23: MultiParameterSweeper.py opened in the Visual Studio editor window

12. Run MultiParameterSweeper.py using command prompt.

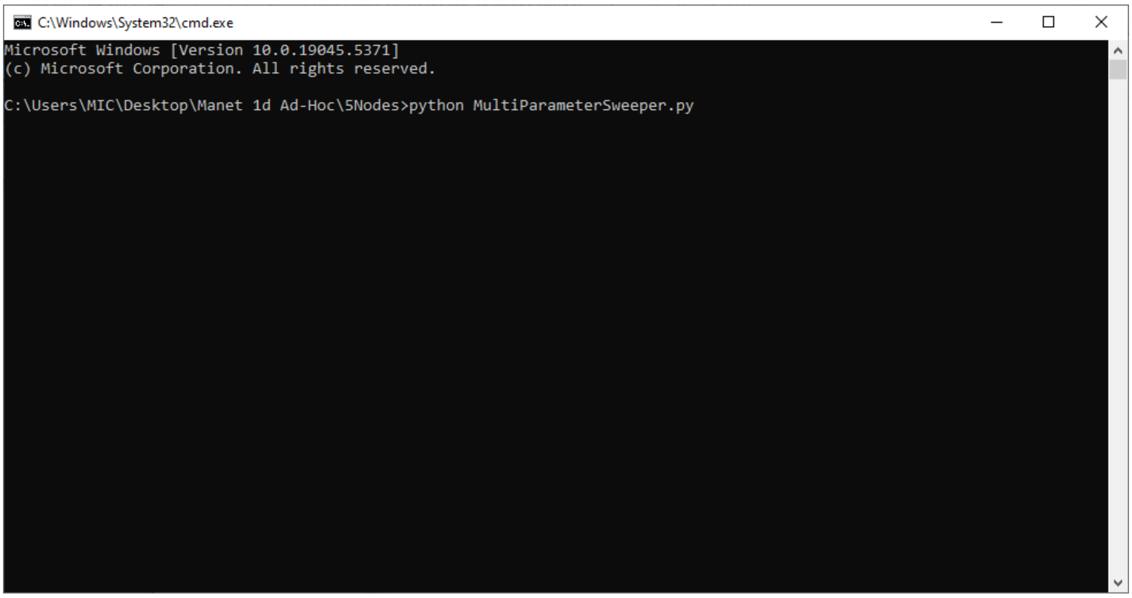


Figure 8-24: Command prompt window in MultiParameterSweeper.py

13. The multi-parameter sweeper runs a total of 2000 simulations, varying X-coordinates between nodes and all transmission range values. It generates an output file named "result.csv" to store the collected data. (It took us approximately 2 hours to complete all 2000 simulations; we used a machine with a i5 processor and with 8 GB RAM).
14. To obtain the number of times the network is connected from the results, similar Excel procedures as with 2Nodes can be followed.

8.1.10 Python code for obtaining p_c from the analytical expression.

We recall that the theoretical formula for probability of network connectivity for n nodes is

$$p_c(n,z,r) = \frac{U_c(n,z,r)}{U(n,z)} = \sum_{k=0}^{n-1} \binom{n-1}{k} (-1)^k \frac{(z-kr)^n}{z^n} u(z-kr)$$

where $u(z)$ is the unit step function i.e., $u(z) = 0$ when $z \leq 0$ and $u(z) = 1$ when $z > 0$.

Python Code for obtaining probability of network connectivity.

```
import numpy as np
import math
from scipy.special import comb

def unit_func(z):
    return 1 if z > 0 else 0

def analytical_probability(n):
    analytical_prob = []
```

```
    for j, r_z in enumerate(np.arange(0.05, 1.05, 0.05), start=1):
        pro = 0
        for k in range(n):
            pro += comb(n-1, k) * ((-1) ** k) * ((1 - k * r_z) ** n) * unit_func(1 - k * r_z)
        analytical_prob.append(round(pro, 3))  # Rounding to 3 decimals

    print(analytical_prob)
    return analytical_prob
# Example usage
n = 10 # Replace with your desired value
analytical_probability(n)
```

8.1.11 Results

The values in the table below for the columns marked "Analysis" are calculated using the python code provided above, with n (the number of nodes) as an input to the python function.

Normalized transmission range (r/z)	Network Connectivity Fraction (N=2)		Network Connectivity Fraction (N=5)		Network Connectivity Fraction (N=10)		Network Connectivity Fraction (N=20)	
	Sim.	Analysis	Sim.	Analysis	Sim.	Analysis	Sim.	Analysis
0.05	0.09	0.097	0.00	0.001	0.00	0.000	0.00	0.000
0.10	0.16	0.190	0.00	0.010	0.00	0.002	0.03	0.019
0.15	0.23	0.277	0.06	0.043	0.06	0.045	0.39	0.393
0.20	0.34	0.360	0.15	0.115	0.21	0.243	0.77	0.787
0.25	0.42	0.437	0.32	0.234	0.53	0.528	0.91	0.939
0.30	0.47	0.510	0.43	0.389	0.72	0.750	0.99	0.984
0.35	0.56	0.577	0.6	0.550	0.9	0.879	1	0.996
0.40	0.63	0.640	0.77	0.691	0.94	0.946	1.00	0.999
0.45	0.68	0.697	0.82	0.799	0.97	0.977	1.00	0.999
0.50	0.73	0.750	0.92	0.875	1.00	0.991	1.00	1.00
0.55	0.78	0.797	0.95	0.926	1.00	0.997	1.00	1.00
0.60	0.85	0.840	0.95	0.959	1.00	0.999	1.00	1.00
0.65	0.87	0.877	0.98	0.979	1.00	1.00	1.00	1.00
0.70	0.88	0.910	1.00	0.990	1.00	1.00	1.00	1.00
0.75	0.94	0.937	1.00	0.996	1.00	1.00	1.00	1.00
0.80	0.95	0.960	1.00	0.999	1.00	1.00	1.00	1.00
0.85	0.99	0.977	1.00	1.00	1.00	1.00	1.00	1.00
0.90	1.00	0.990	1.00	1.00	1.00	1.00	1.00	1.00
0.95	1.00	0.997	1.00	1.00	1.00	1.00	1.00	1.00
1.00	1.00	1.000	1.00	1.00	1.00	1.00	1.00	1.00

Table 8-3: Results of NetSim simulation and analysis for N= 2, 5, 10 and 20 nodes. The network connectivity fraction is the ratio of number of times the network is connected to the total number of simulations runs.

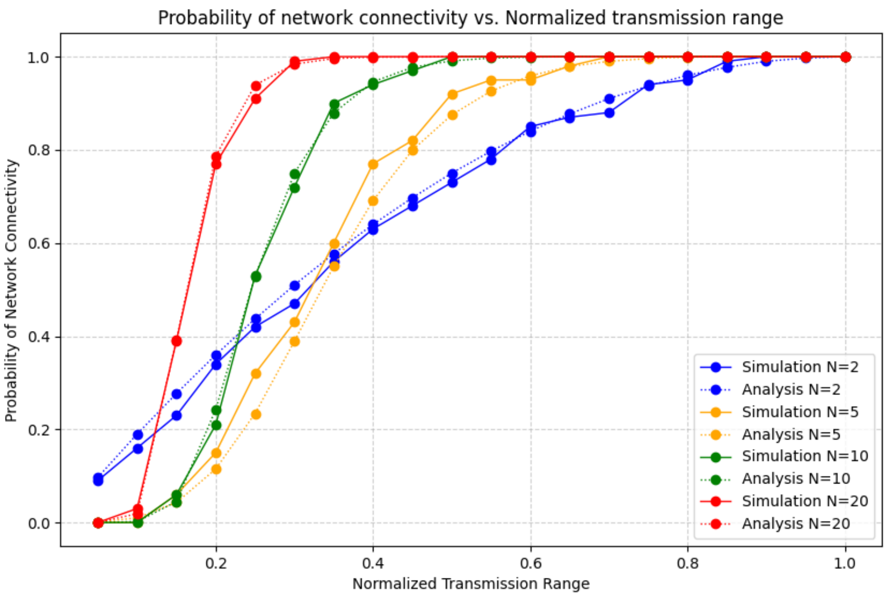


Figure 8-25: Plot comparing simulation results against analytical model for probability of network connectivity for different transmission ranges.

8.1.12 Exercises

1. Configure NetSim simulations for node counts such as 3, 4, 6, 7, 8 etc. Then tabulate the output results containing Network connectivity for various Normalized Transmission Ranges using NetSim simulation, and from the closed form expression provided. Plot the results and compare simulation results vs. theory.

8.1.13 References

[1] A. Kumar, D. Manjunath and J. Kuri, Wireless Networking, Morgan Kaufmann, 2008.

9 4G LTE

9.1 LTE Handover (Level 1)

Handover is a critical process in LTE networks that ensures seamless connectivity and service continuity for mobile users as they move across different cell areas. The goal of handover is to provide the best possible connection by dynamically switching the User Equipment (UE) to the eNodeB (eNB) that offers the best signal and optimal quality of service. This experiment focuses the concepts, algorithms, and procedures involved in LTE handover.

9.1.1 Handover modeling in NetSim

The handover logic in NetSim LTE is based on the Strongest Adjacent Cell Handover Algorithm [1]. The algorithm enables each UE to connect to that eNB which provides the highest signal quality. Therefore, a handover occurs the moment a better eNB i.e., when adjacent cell has offset stronger signal quality is detected. The exact handover algorithm including details of the offset (also known as handover-margin) is explained in the 5G manual.

9.1.2 Use of SNR instead of RSRP

NetSim is a packet-level simulator for simulating the performance of end-to-end applications over various packet transport technologies. NetSim can scale to simulating networks with 100s of UEs, eNBs, routers, switches, etc. In order to achieve a scalable simulation, that can execute in reasonable time on desktop-level computers, many details of the physical layer techniques have been abstracted.

In NetSim LTE, there are no pilots/reference/synchronization signals. The channel matrix H is assumed to be known perfectly and instantaneously at the transmitter and receiver, respectively. Hence there is no RSRP, and all signal power related calculations are done in the SSB channel. Therefore, the hand-over is based on the SNR measured at the source (serving) eNB and the target (neighbouring) eNB. Since the noise power would be the same at s-eNB and t-eNB, in effect the handover is based on received signal level on the SSB.

9.1.3 Control packet exchange

1. During initial attachment, a UE associates with the eNB from which it sees the highest SNR. The eNB to which a UE is attached is called the serving eNB.
2. The UE periodically sends the UE MEASUREMENT REPORT to the Serving eNB, which contains the SNRs from all the nearby eNBs. The Serving eNB makes the

decision on whether to hand off the UE to a T-eNB (Target-eNB) based on the handover condition

$$SNR_{t-gNB} - SNR_{s-gNB} > HOM$$

In NetSim by default the HOM is set to 3 dB.

3. If the condition is met, the S-eNB issues a HANDOVER REQUEST message to the T-eNB passing necessary information to prepare the handover at the target side.
4. The T-eNB sends back the HANDOVER REQUEST ACKNOWLEDGE message including a transparent container to be sent to the UE as an RRC message to perform the handover.
5. The S-eNB generates the RRC (Radio resource control used for signaling transfer) message to perform the handover, i.e., RRC CONNECTION RECONFIGURATION message including the mobility Control Information.
6. The S-eNB starts forwarding the downlink data packets to the T-eNB for all the data bearers which are being established in the T-eNB during the HANDOVER REQ message processing.
7. The T-eNB now requests the S-eNB to release the resources. With this, the handover procedure is complete.

9.1.4 Scenario

Open NetSim and Select **Experiments** > **LTE** > **Handover in 4G** then click on the tile in the middle panel to load the example as shown in below screenshot Figure 9-1.

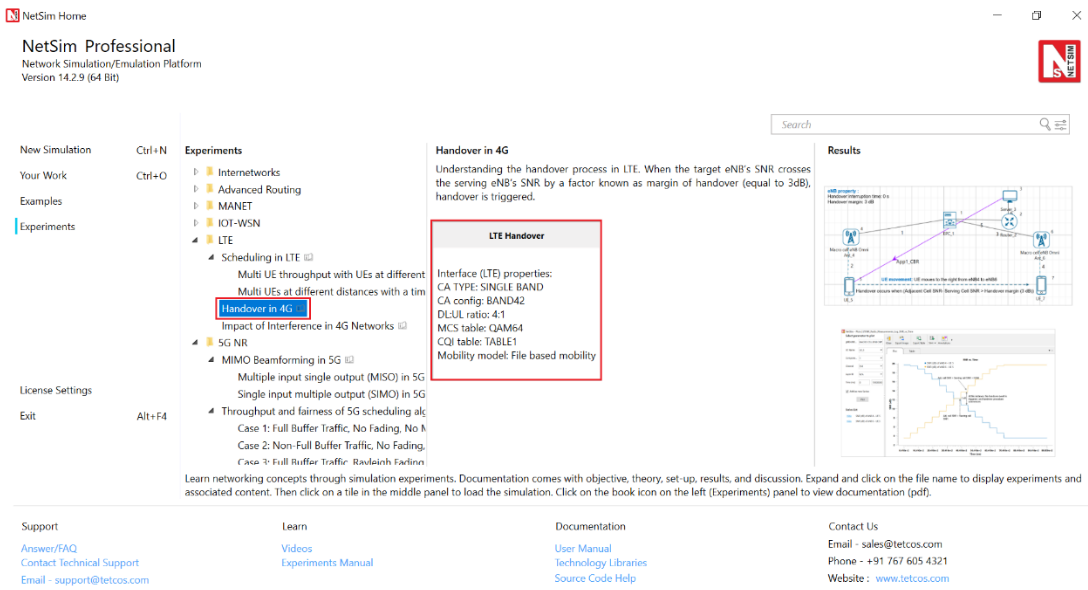


Figure 9-1: List of scenarios for the example of LTE Handover

The following network diagram illustrates what the NetSim UI displays when you open the experiment configuration file as shown Figure 9-2.

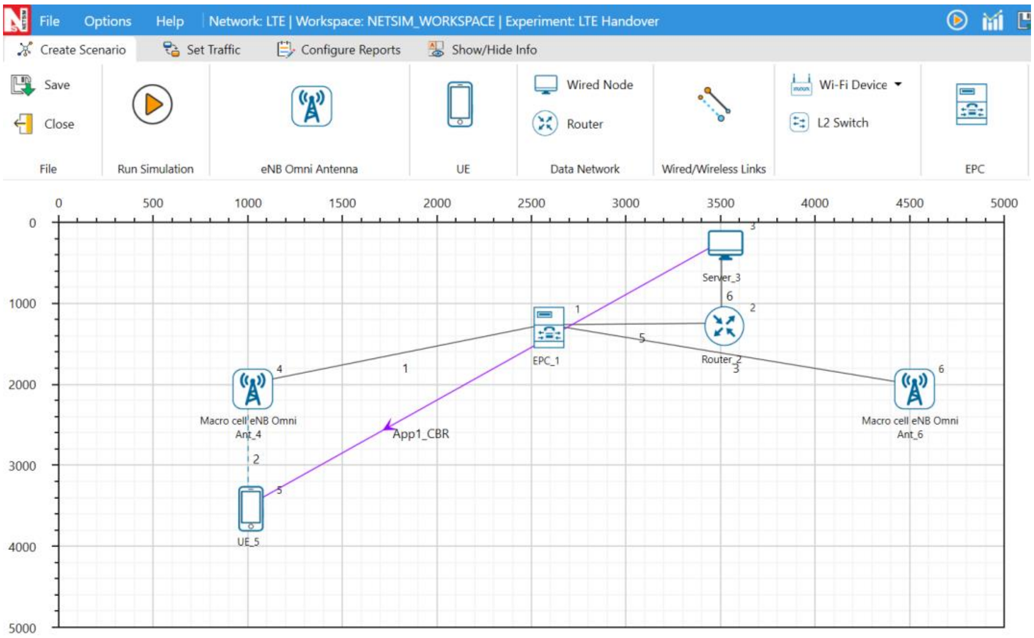


Figure 9-2: Network set up for studying the LTE-Handover

9.1.5 Network Settings

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

Step 1: Environment Grid length: 6000m x 3000m.

Step 2: A network scenario is designed in NetSim GUI comprising of 2 ENBs, 1 EPC, 1UE, 1 Router and 1 server in the “LTE/LTE-A” Network Library.

Step 3: The device positions are set as per the table given below.

	ENB 4	ENB 6	UE 5
X Co-ordinate	1000	4500	1000
Y Co-ordinate	1000	1000	2500

Table 9-1: Device Position

Step 4: The eNB properties are set as follows. To configure them, click on the eNB, and in the right panel, expand the window to view all layer properties and set the properties listed below.

Interface (LTE) Properties	
CA TYPE	SINGLE BAND
CA Configuration	BAND 42
Component Carrier 1	
DL UL Ratio	4:1
Numerology	0
Channel Bandwidth (MHz)	10
PDSCH Configuration	
MCS Table	QAM64
CSI Report Configuration	
CQI Table	TABLE1
Channel Model	
Outdoor Scenario	URBAN MACRO
LOS NLOS Selection	USER DEFINED

LOS Probability	1
Shadow Fading Model	None
Fast Fading Model	No Fading

Table 9-2: eNB > Interface (LTE) Properties Setting

Similarly, it is set for eNB 6.

Step 5: In the UE 5 device position properties, set Mobility Model as File Based Mobility.

File Based Mobility

In File Based Mobility, users can write their own custom mobility models and define the movement of the mobile users. Create a mobility.csv file for UE's involved in mobility with each step equal to 0.5 sec with distance 250 m. where the UE moves 2500m away from the eNB 4 towards eNB 6.

The NetSim Mobility File (mobility.csv) format is as follows:

#Time(s)	Device ID	X	Y	Z
0	5	1000	2500	0
0.5	5	1250	2500	0
1	5	1500	2500	0
1.5	5	1750	2500	0
2	5	2000	2500	0
2.5	5	2250	2500	0
3	5	2500	2500	0
3.5	5	2750	2500	0
4	5	3000	2500	0
4.5	5	3250	2500	0
5	5	3500	2500	0
5.5	5	3750	2500	0
6	5	4000	2500	0
6.5	5	4250	2500	0
7	5	4500	2500	0

Table 9-3: Mobility.csv file

The mobility path for UE 5 can be observed using the mobility viewer as shown below. For more detailed on mobility viewer refer the user manual section 8.6.

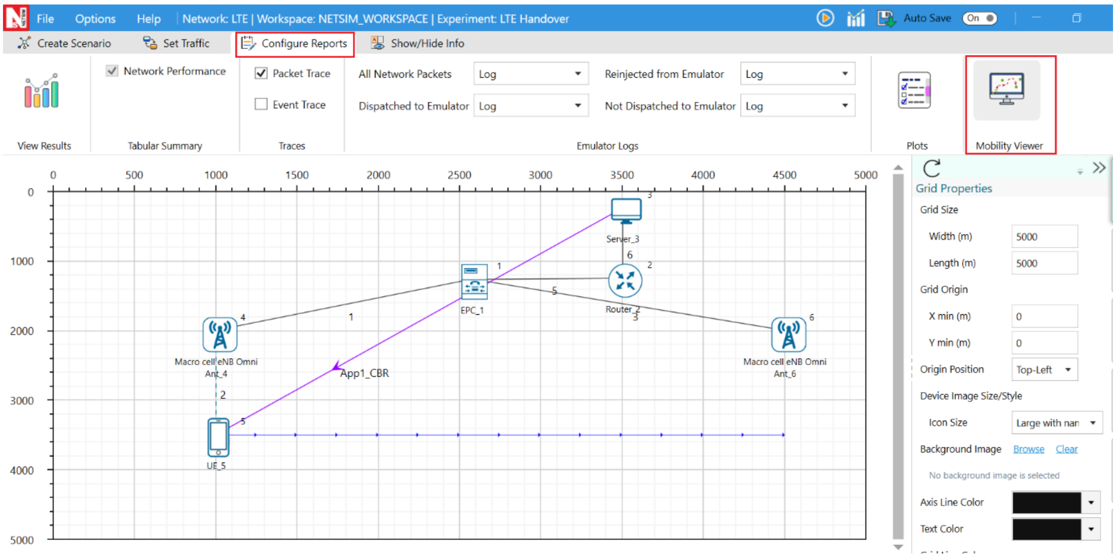


Figure 9-3: Mobility viewer showing the UE 5 mobility path

Step 6: Configure an application between the server 3 (wired node 3) and UE 5 by selecting a CBR application from the **Set Traffic tab**. The QoS for the CBR application is set to UGS. To configure the QOS, click on the created application and expand the application property in the right, and set the QOS as mentioned.

Step 7: Enable Packet Trace, SNR vs Time plot and Logs in NetSim GUI as shown in below figure. At the end of the simulation, a large .csv file contains all the packet information and is available for the users to perform packet level analysis.

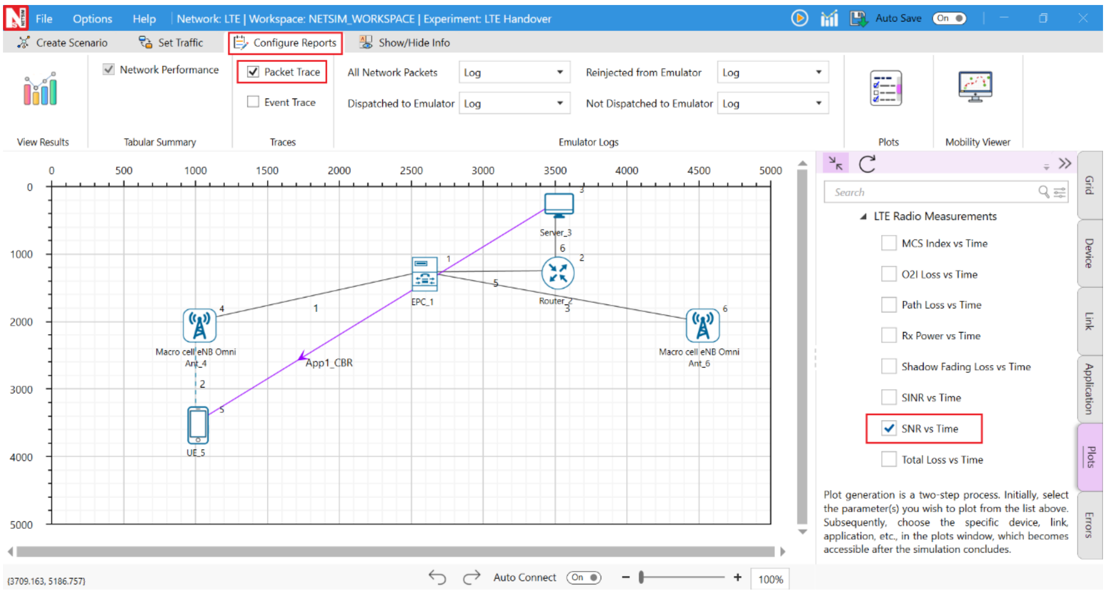


Figure 9-4: Enabling plots, packet trace and log file.

Step 8: Run the Simulation for 10 Seconds.

9.1.6 Results and Discussion

9.1.6.1 Handover Signaling

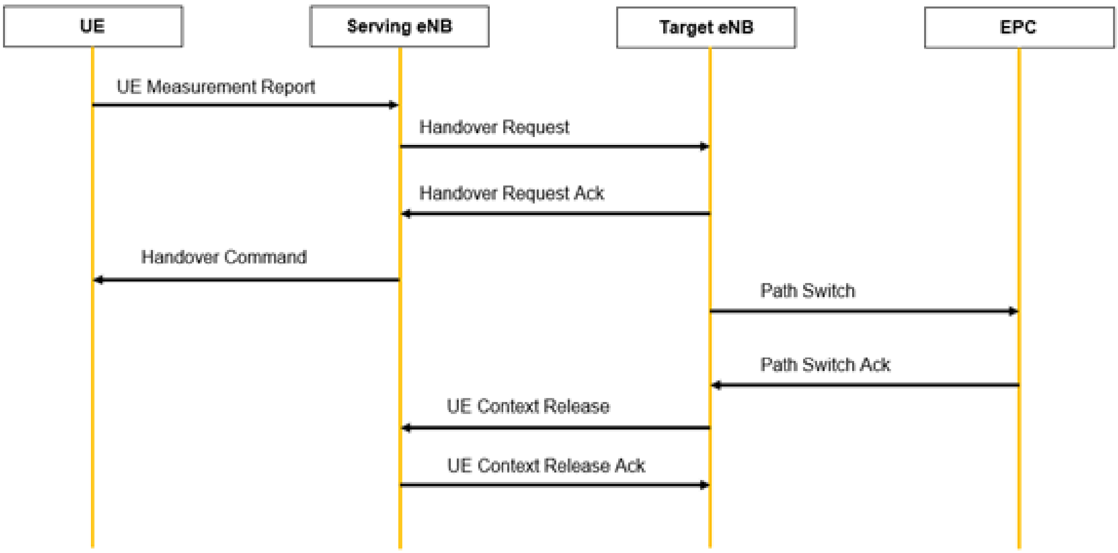


Figure 9-5: Control packet flow in the LTE handover process

Note:

- The Handover Request will be sent from the serving eNB to the target eNB, and the Handover Request Ack will be sent from the target eNB to the serving eNB through the EPC.
- The Context Release will be sent from the serving eNB to the target eNB, and the Context Release Ack will be sent from the target eNB to the serving eNB through the EPC.

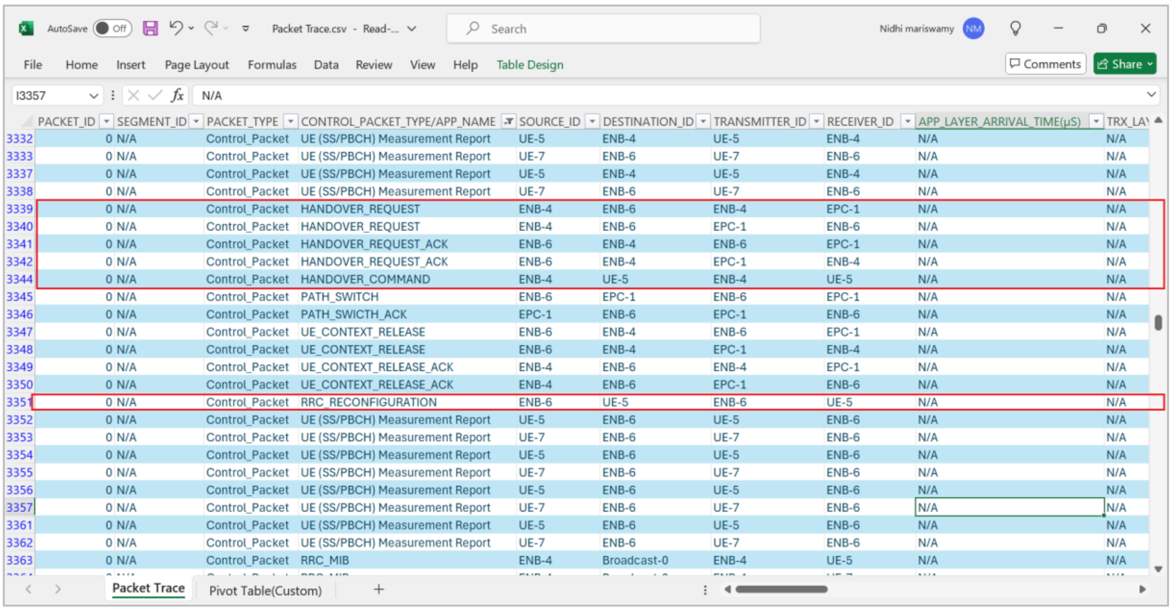
The packet flow depicted above can be observed from the packet trace.

1. UE will send the UE (SS/PBCH) MEASUREMENT REPORT every 5 ms to the connected eNB.
2. The initial UE- eNB connection, eNB will send the RRC MIB packets to the UE every 40 ms and RRC SIB1 every 80 ms.
3. After the transmission of the RRC MIB and RRC SIB1 packets, the eNB will send RRC_SI packet to the UE.
4. After reception of RRC SI packet, UE will send RRC Setup Request to the eNB.
5. On receiving the RRC Setup Request packet, the eNB will acknowledge the request by transmitting RRC Setup packet to the UE.
6. The UE will send back the RRC Setup Complete packet on the receipt of RRC Setup message.

	PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(µS)	TRX_LAYE
10	0	N/A	Control_Packet	RRC_MIB	ENB-4	Broadcast-0	ENB-4	UE-5	N/A	N/A
11	0	N/A	Control_Packet	RRC_MIB	ENB-6	Broadcast-0	ENB-6	UE-5	N/A	N/A
12	4	0	CBR	App1_CBR	NODE-3	UE-5	NODE-3	ROUTER-2	60000	
13	4	0	CBR	App1_CBR	NODE-3	UE-5	ROUTER-2	EPC-1	60000	
14	5	0	CBR	App1_CBR	NODE-3	UE-5	NODE-3	ROUTER-2	80000	
15	5	0	CBR	App1_CBR	NODE-3	UE-5	ROUTER-2	EPC-1	80000	
16	0	N/A	Control_Packet	RRC_SIB1	ENB-4	Broadcast-0	ENB-4	UE-5	N/A	N/A
17	0	N/A	Control_Packet	RRC_MIB	ENB-4	Broadcast-0	ENB-4	UE-5	N/A	N/A
18	0	N/A	Control_Packet	RRC_SIB1	ENB-6	Broadcast-0	ENB-6	UE-5	N/A	N/A
19	0	N/A	Control_Packet	RRC_MIB	ENB-6	Broadcast-0	ENB-6	UE-5	N/A	N/A
20	0	N/A	Control_Packet	RRC_SI	ENB-4	UE-5	ENB-4	UE-5	N/A	N/A
21	0	N/A	Control_Packet	RRC_SETUP_REQUEST	UE-5	ENB-4	UE-5	ENB-4	N/A	N/A
22	0	N/A	Control_Packet	RRC_SETUP	ENB-4	UE-5	ENB-4	UE-5	N/A	N/A
23	0	N/A	Control_Packet	RRC_SETUP_COMPLETE	UE-5	ENB-4	UE-5	ENB-4	N/A	N/A
24	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-4	UE-5	ENB-4	N/A	N/A
25	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-4	UE-5	ENB-4	N/A	N/A
26	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-4	UE-5	ENB-4	N/A	N/A
27	6	0	CBR	App1_CBR	NODE-3	UE-5	NODE-3	ROUTER-2	100000	
28	6	0	CBR	App1_CBR	NODE-3	UE-5	ROUTER-2	EPC-1	100000	
29	6	0	CBR	App1_CBR	NODE-3	UE-5	EPC-1	ENB-4	100000	
30	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-4	UE-5	ENB-4	N/A	N/A
31	6	0	CBR	App1_CBR	NODE-3	UE-5	ENB-4	UE-5	100000	
32	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-4	UE-5	ENB-4	N/A	N/A
33	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-4	UE-5	ENB-4	N/A	N/A
34	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-4	UE-5	ENB-4	N/A	N/A

Figure 9-6: Packet trace file showing the RRC Initial Association.

7. As Per the configured file-based mobility, UE 5 moves towards eNB 6.
8. After 4.5s eNB 4 sends the HANDOVER REQUEST to eNB 6 through EPC 1.
9. eNB 6 sends back HANDOVER REQUEST ACK to eNB 4 through EPC 1.
10. After receiving HANDOVER REQUEST ACK from eNB 6, eNB 4 sends the HANDOVER COMMAND to UE 5
11. After the HANDOVER COMMAND packet is transferred to the UE, the target eNB will send the PATH SWITCH packet to the EPC 1.
12. When the EPC 1 receives the PATH SWITCH packet, it sends PATH SWICTH ACK packet to the eNB 6.
13. The target eNB sends CONTEXT RELEASE to source eNB, and the source eNB sends back CONTEXT RELEASE ACK to target eNB. The context release request and ack packets are sent between the source and target eNB via EPC 1.
14. RRC Reconfiguration will take place between target eNB and UE 5.
15. The UE 5 will start sending the UE (SS/PBCH) MEASUREMENT REPORT to eNB 6



	PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_TIME(μS)	TRX_LA
3332	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-4	UE-5	ENB-4	N/A	N/A
3333	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-7	ENB-6	UE-7	ENB-6	N/A	N/A
3337	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-4	UE-5	ENB-4	N/A	N/A
3338	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-7	ENB-6	UE-7	ENB-6	N/A	N/A
3339	0	N/A	Control_Packet	HANDOVER_REQUEST	ENB-4	ENB-6	ENB-4	EPC-1	N/A	N/A
3340	0	N/A	Control_Packet	HANDOVER_REQUEST	ENB-4	ENB-6	EPC-1	ENB-6	N/A	N/A
3341	0	N/A	Control_Packet	HANDOVER_REQUEST_ACK	ENB-6	ENB-4	ENB-6	EPC-1	N/A	N/A
3342	0	N/A	Control_Packet	HANDOVER_REQUEST_ACK	ENB-6	ENB-4	EPC-1	ENB-4	N/A	N/A
3344	0	N/A	Control_Packet	HANDOVER_COMMAND	ENB-4	UE-5	ENB-4	UE-5	N/A	N/A
3345	0	N/A	Control_Packet	PATH_SWITCH	ENB-6	EPC-1	ENB-6	EPC-1	N/A	N/A
3346	0	N/A	Control_Packet	PATH_SWICTH_ACK	EPC-1	ENB-6	EPC-1	ENB-6	N/A	N/A
3347	0	N/A	Control_Packet	UE_CONTEXT_RELEASE	ENB-6	ENB-4	ENB-6	EPC-1	N/A	N/A
3348	0	N/A	Control_Packet	UE_CONTEXT_RELEASE	ENB-6	ENB-4	EPC-1	ENB-4	N/A	N/A
3349	0	N/A	Control_Packet	UE_CONTEXT_RELEASE_ACK	ENB-4	ENB-6	ENB-4	EPC-1	N/A	N/A
3350	0	N/A	Control_Packet	UE_CONTEXT_RELEASE_ACK	ENB-4	ENB-6	EPC-1	ENB-6	N/A	N/A
3351	0	N/A	Control_Packet	RRC_RECONFIGURATION	ENB-6	UE-5	ENB-6	UE-5	N/A	N/A
3352	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-6	UE-5	ENB-6	N/A	N/A
3353	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-7	ENB-6	UE-7	ENB-6	N/A	N/A
3354	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-6	UE-5	ENB-6	N/A	N/A
3355	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-7	ENB-6	UE-7	ENB-6	N/A	N/A
3356	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-6	UE-5	ENB-6	N/A	N/A
3357	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-7	ENB-6	UE-7	ENB-6	N/A	N/A
3361	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-5	ENB-6	UE-5	ENB-6	N/A	N/A
3362	0	N/A	Control_Packet	UE (SS/PBCH) Measurement Report	UE-7	ENB-6	UE-7	ENB-6	N/A	N/A
3363	0	N/A	Control_Packet	RRC_MIB	ENB-4	Broadcast-0	ENB-4	UE-5	N/A	N/A

Figure 9-7: Packet trace file showing control messages involved during the handover process.

9.1.6.2 Plot of SNR vs. Time

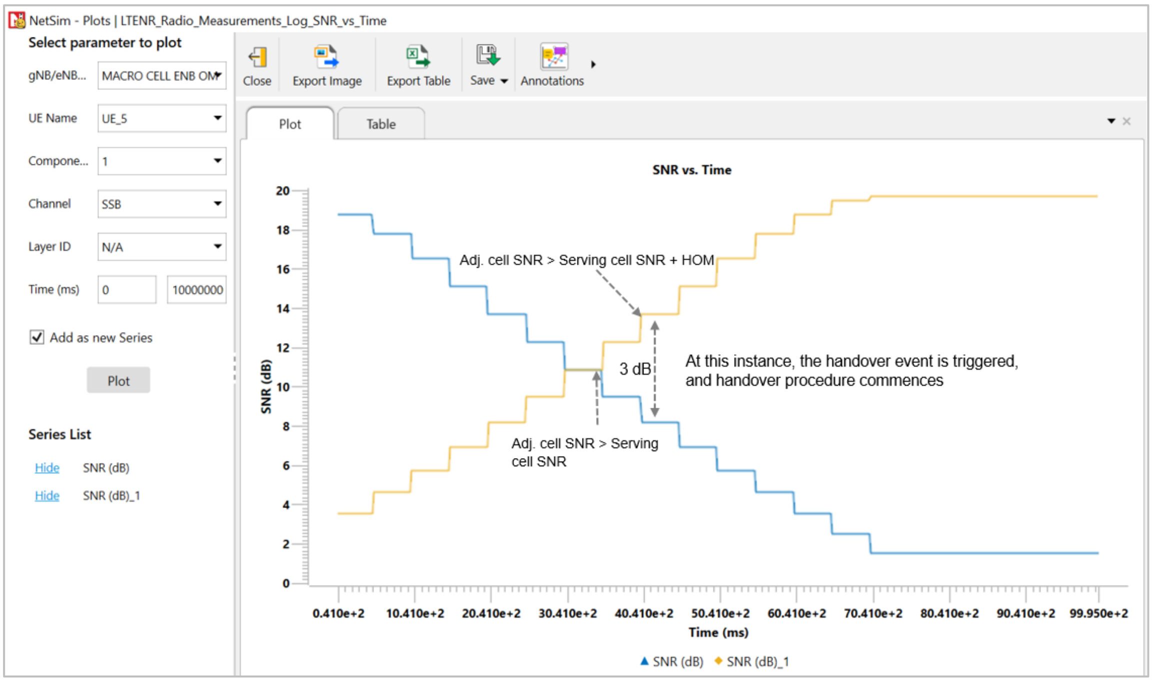


Figure 9-8: Plot of the DL SNR over time seen by the UE from the serving cell (eNB 4) and the target cell (eNB 6). The handover process does not commence with Adj. cell SNR is greater than Serving cell SNR but only commences with Adj. cell SNR is greater than Serving cell SNR by the Handover margin (3 dB in this case). The SNR is measured in the SSB channel.

This chart can be obtained in NetSim by enabling the option to plot SNR vs. time prior to the simulation. First, plot the SNR curve for eNB 4 and UE 5 keeping the channel as PDSCH and Layer ID as 1. Then select "Add as new series" and select the gNB/eNB as eNB 6 and UE name as UE 5. Click on plot, and you would then obtain the above "stacked" plot

- Time 4.5s when the SNR from eNB 4 is 9.47dB and the SNR from eNB 6 is 15.12dB. This represents the point where Adj cell RSRP is greater than serving cell RSRP by Hand-over margin (HOM) of 3dB.

9.1.7 References

[1] K. Dimou, “Handover within 3GPP LTE: Design Principles and Performance,”Ericsson Research

9.2 Impact of Interference in 4G Networks (Level 3)

9.2.1 Objective

In this experiment, we will simulate and study the impact of downlink interference on the signal-to-interference ratio (SINR) in NetSim v14.2. We will study the following aspects.

A. We consider a handover procedure in a cellular system and analyse the following cases:

- The handover of a UE without any interference, with pathloss exponent, $\eta = 2.5$,
- The handover of a UE without any interference, with pathloss exponent $\eta = 4$,
- The handover of a UE with interference, with pathloss exponent $\eta = 2.5$, and
- The handover of a UE with interference, with pathloss exponent $\eta = 4$.

B. To understand the effect of path loss exponents with and without interferences on the point of handovers in cellular systems.

In this experiment, we consider the following handover scenario: A UE starts from BS_1 and moves in a straight line to BS_2. While the UE is attached to BS_1 it experiences interference from BS_2. Once it gets handed over to BS_2 the UE experiences interference from BS_1. There is always thus only one interferer, and we analyse the SINR as the UE moves a straight line from BS_1 to BS_2.

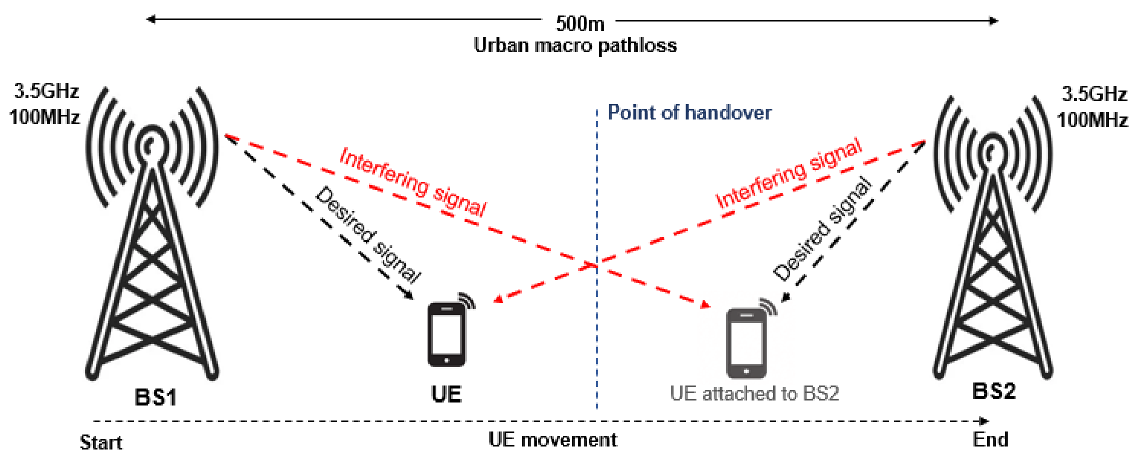


Figure 9-9: UE1 is initially attached to BS1. The signal from BS1 is the desired signal while the signal from BS2 is the interfering signal. Post-handover, the desired signal is transmitted from BS2 while signal from BS1 is the interfering signal. We assume omni-directional antennas at both BSs and consider cases with $\eta = 2.5, 4$.

9.2.2 Introduction[6]

Due to the scarcity of the wireless spectrum, it is not possible in 4G networks to separate concurrent transmissions completely in frequency. Some transmissions will necessarily occur at the same time in the same frequency band, separated only in space, and the signals operating on the same time-frequency resources from many undesired or interfering transmitters are added to the desired transmitter's signal at a receiver. The main determinants of the interference are,

- The network geometry, i.e., the location of the receivers and the transmitters
- Base stations' (or eNBs') transmit power, and
- The path loss model (signal attenuation with distance).

The performance and coverage of a 4G network critically depends on the signal-to-interference-and-noise ratios (SINRs) level at the receivers. This is defined as

$$SINR = \frac{P_r}{N_0 W + I}$$

Where P_r is the received power of the desired signal, W is the bandwidth, $N_0 W$ is the thermal noise and I is the received power of interfering signals. In 4G, the modulation and coding scheme (MCS) is computed from the SINR. The higher the SINR, the higher the MCS, and hence the higher the date rate. Interference is therefore an important performance-limiting factor in wireless networks and hence it is crucial to characterize the effect of interference.

9.2.3 Network Simulation Setup

Open NetSim and click on **Experiments > LTE > Impact of Interference in 4G Networks** then click on the tile in the middle panel to load the example as shown in below.

[6] Some of the content in this section is from [11.2.1]

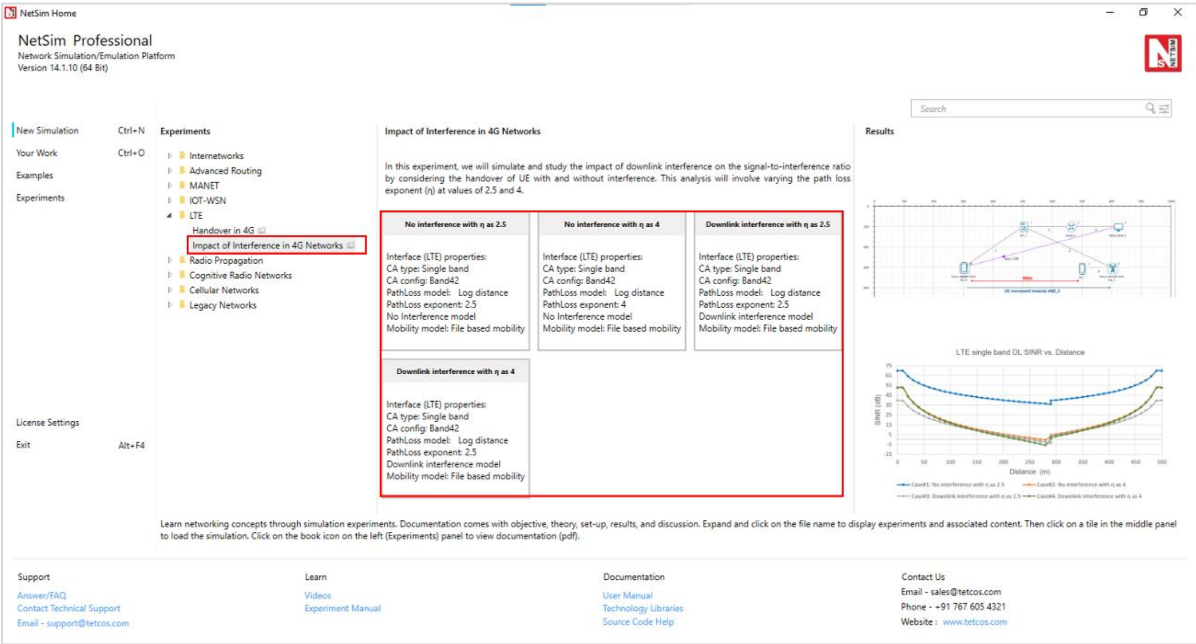


Figure 9-10: List of scenarios for the example of Impact of Interference in 4G Networks.

ISD = 500m, BAND42, 20 MHz, Urban Macro

In our network scenario, the inter-site distance (ISD) between BS_1 and BS_2 is 500m. Both base stations (eNB) operate in the 3.4 GHz band, with a bandwidth of 20 MHz. The environment is assumed to be urban with signal attenuation as per the 3G PP Urban Macro pathloss model. Shadow-fading and fast fading are turned off to avoid sources of randomness.

Case 1: No interference in both base stations with $\eta = 2.5$

Network Scenario

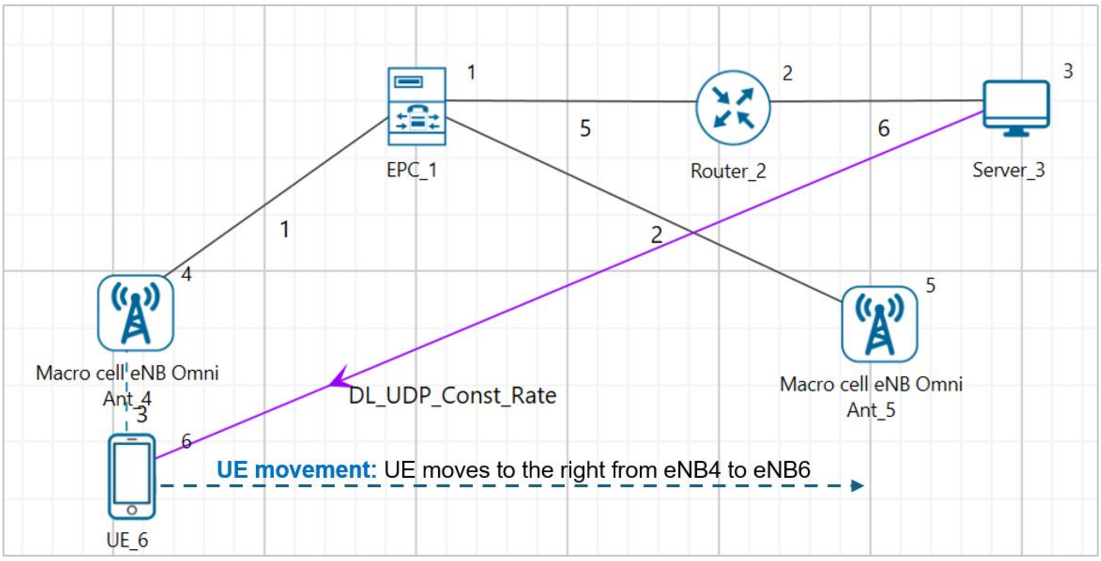


Figure 9-11: NetSim Scenario during mobility.

Simulation Parameters

Set grid length as 1600 x 800 from grid setting property panel on the right. This needs to be done before any device is placed on the grid. Place the second eNB 500 meters away from the first eNB

Devices	X	Y
eNB 4	300	600
eNB 5	800	600
UE 6	300	600

Table 9-4: Device Co-ordinates.

- Click on eNB and expand property panel on right and change the properties as mentioned in the below steps. Similarly set the same properties in another eNB.

eNB> Interface LTE	
Physical Layer Properties	
eNB Height (m)	10
Tx Power (dBm)	40dBm
CA Type	Single Band
CA Configuration	BAND42
Component Carrier 1	
DL: UL Ratio	4:1
F Low (MHz)	3400
F High (MHz)	3600
Numerology	0
Channel Bandwidth (MHz)	20MHZ
Antenna	
Tx Antenna Count	1
Rx Antenna Count	1
PDSCH and PUSCH Configuration	
MCS Table	QAM64
CSI Report Configuration	
CQI Table	Table1
Channel Model	
Pathloss model	Log distance
Pathloss exponent	2.5
Shadowing model	None
Fast Fading Model	No Fading
Interference Model	
Downlink interference model	No interference

Table 9-5: Values set for different parameters in simulation.

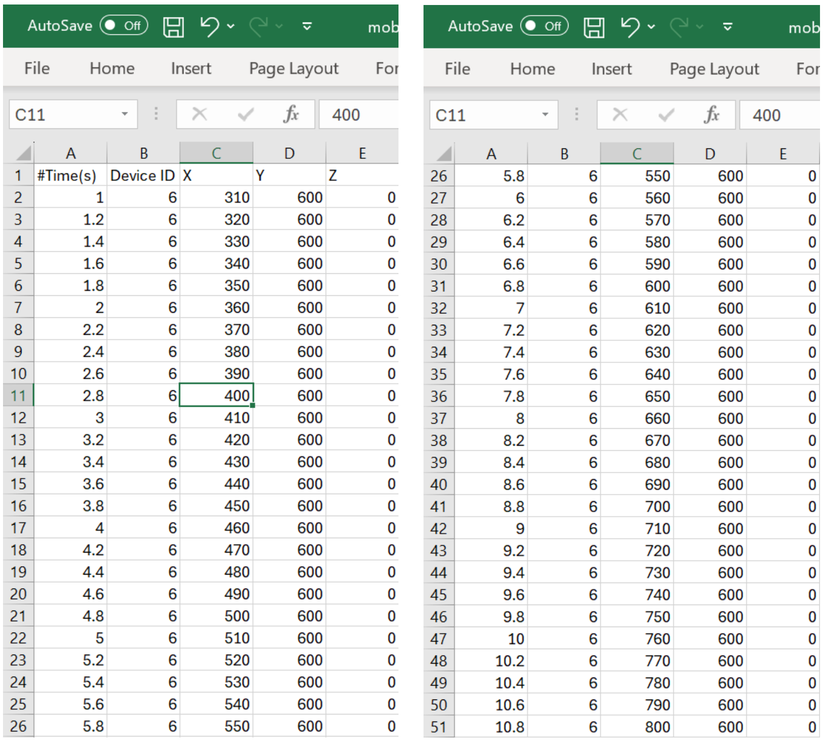
- Go to UE properties. In the RAN Interface set physical layer properties in both UEs as shown below.

UE > Interface LTE>Physical Layer	
UE Height	1.50
Tx Power	23 dBm

Antenna	
Tx Antenna Count	1
Rx Antenna Count	1

Table 9-6: Properties set for UE

- In the General layer, set UE X and Y coordinates as the eNB1's X and Y coordinates. That is, the initial position of UE must be the position of eNB1.
- Set the mobility model as file-based mobility and configure the mobility that UE needs to be travel straight towards to the eNB2. So, configure the mobility file according to the distance that UE needs to travel. For example, in the above scenario, UE needs to travel 500m from eNB1 to eNB2 So, it is travelling straight towards to another eNB since it's Y coordinate is fixed. Hence, give input in the excel sheet as in Figure 9-12



	A	B	C	D	E
1	#Time(s)	Device ID	X	Y	Z
2	1	6	310	600	0
3	1.2	6	320	600	0
4	1.4	6	330	600	0
5	1.6	6	340	600	0
6	1.8	6	350	600	0
7	2	6	360	600	0
8	2.2	6	370	600	0
9	2.4	6	380	600	0
10	2.6	6	390	600	0
11	2.8	6	400	600	0
12	3	6	410	600	0
13	3.2	6	420	600	0
14	3.4	6	430	600	0
15	3.6	6	440	600	0
16	3.8	6	450	600	0
17	4	6	460	600	0
18	4.2	6	470	600	0
19	4.4	6	480	600	0
20	4.6	6	490	600	0
21	4.8	6	500	600	0
22	5	6	510	600	0
23	5.2	6	520	600	0
24	5.4	6	530	600	0
25	5.6	6	540	600	0
26	5.8	6	550	600	0

	A	B	C	D	E
26	5.8	6	550	600	0
27	6	6	560	600	0
28	6.2	6	570	600	0
29	6.4	6	580	600	0
30	6.6	6	590	600	0
31	6.8	6	600	600	0
32	7	6	610	600	0
33	7.2	6	620	600	0
34	7.4	6	630	600	0
35	7.6	6	640	600	0
36	7.8	6	650	600	0
37	8	6	660	600	0
38	8.2	6	670	600	0
39	8.4	6	680	600	0
40	8.6	6	690	600	0
41	8.8	6	700	600	0
42	9	6	710	600	0
43	9.2	6	720	600	0
44	9.4	6	730	600	0
45	9.6	6	740	600	0
46	9.8	6	750	600	0
47	10	6	760	600	0
48	10.2	6	770	600	0
49	10.4	6	780	600	0
50	10.6	6	790	600	0
51	10.8	6	800	600	0

Figure 9-12: UE Mobility file for 500m.

- Before clicking on Run, enable "LTENR Radio Measurements Log" as shown in Figure 9-13

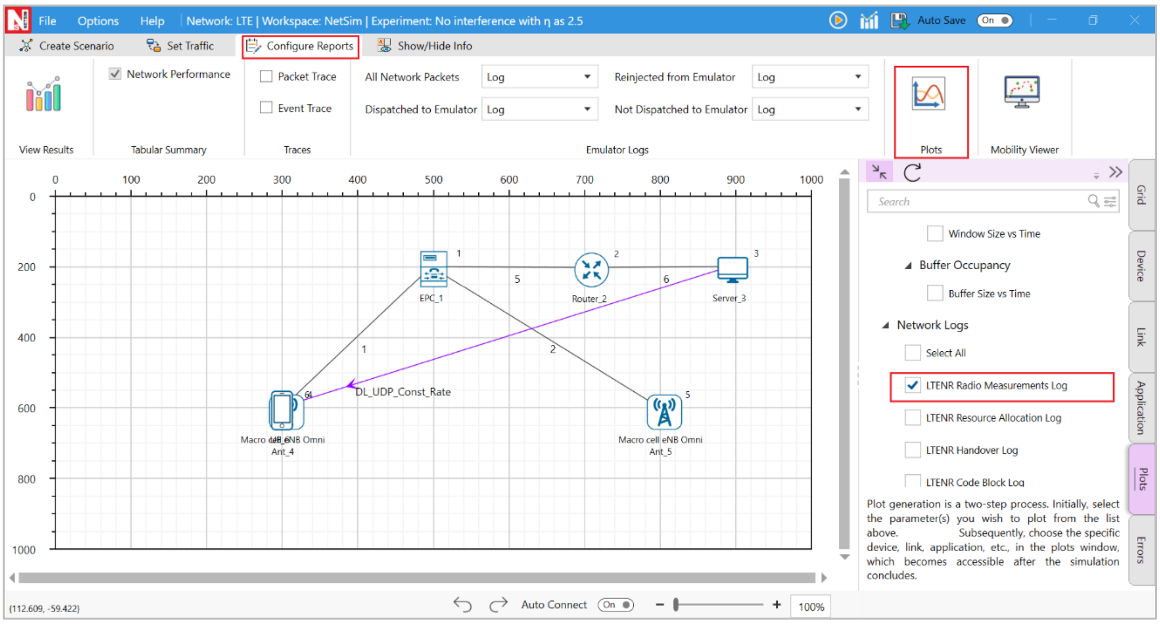


Figure 9-13: Enabling the Log Options from GUI.

- Now, Run the Simulation for 12 s.
- After simulation, open LTENR Radio measurement log present under the Log files section in the Results Dashboard as in Figure 9-14.

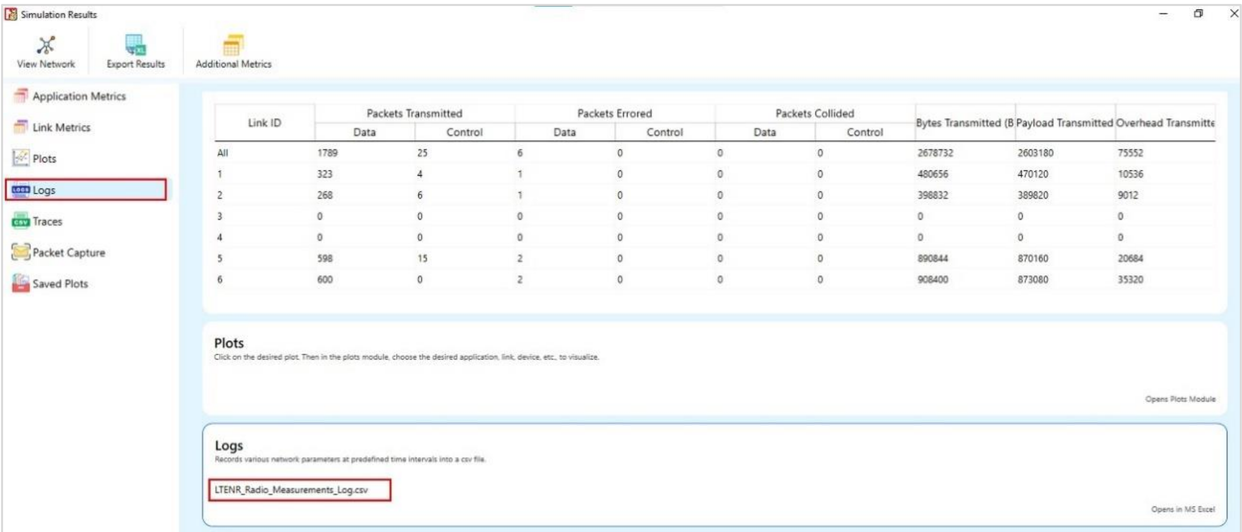


Figure 9-14: ISD 500m simulation result Dashboard.

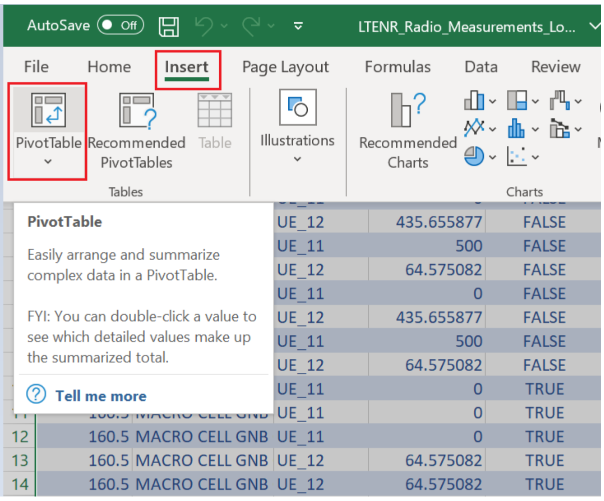


Figure 9-15: Inserting Pivot table.

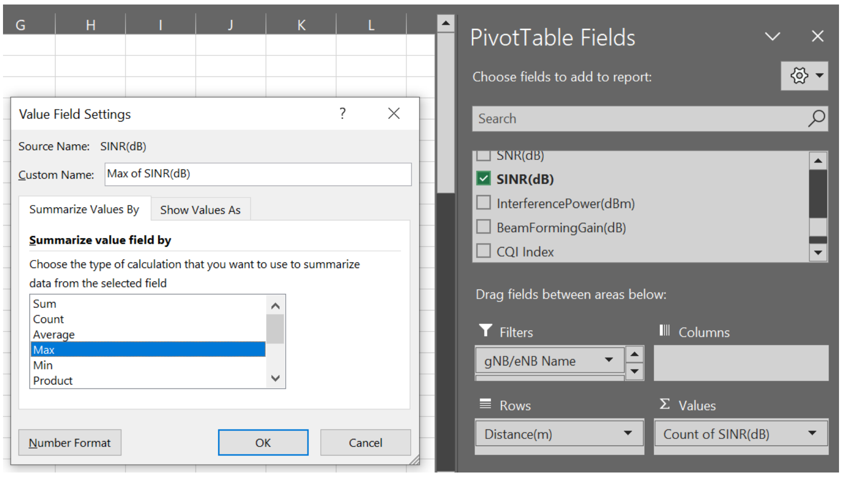


Figure 9-16: Creating Pivot table.

- Create a pivot table for this log file by clicking the pivot option present at the top of the ribbon under insert section as shown below.
- In the pivot table drop 'gNB/eNB Name', 'UE Name', 'Channel' fields under filters area, drop 'Distance' in Row area and drop 'SINR' in Value area. Set the SINR values to max by clicking on the arrow icon present at the end of the field ->value field setting->max as shown below as shown in Figure 9-16
- Now filter eNB as eNB4, UE name as UE 6, Channel as PDSCH as shown in Figure 9-16
- Copy the values from 0 to 290 along with Row Labels and Max SINR dB header and paste it into another sheet. Similarly filter gNB/eNB name to eNB 5 and copy the row label value along with SINR and paste it into next to the previously created new sheet.
- NetSim calculates the distance of a UE from its attached eNB. In the plot that we eventually wish to obtain the X axis has distance from the initial attached eNB which is eNB4. In our experiment, UE6 is initially attached to eNB4 and post-handover it

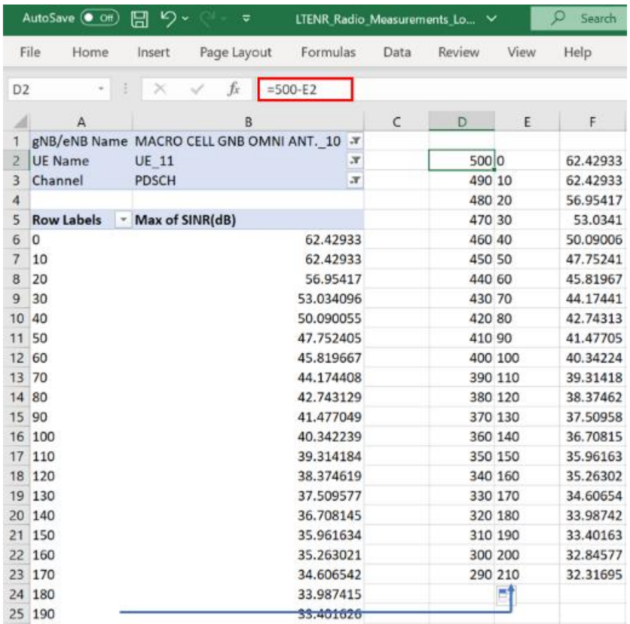
gets attached to eNB5. Since eNB4 to eNB5 distance is 500m, post-handover the distance of UE6 from the initial eNB4 is $500 - d_{gNB(5)}^{UE(6)}$i.e., $d_{gNB(4)}^{UE(6)} = 500 - d_{gNB(5)}^{UE(6)}$.

a. Copy the Row table and distance to the empty cells after filtering that gNB/eNB Name to eNB5 and UE Name to UE6 as shown in Figure 9-17

b. In the adjacent cell calculate the UE 6 distance from eNB 4 as shown below as shown in Figure 9-18. Observe that it is initially $d_{gNB(4)}^{UE(6)}$ and post-handover it is $500 - d_{gNB(5)}^{UE(6)}$.

c. Distance between UE6 and eNB4, $d_{gNB(4)}^{UE(6)}$, along with the SINR value and copy them into new cells.

d. Filter it from the ascending order/ smallest to largest, copy these values to the paste it below the previously created new sheet as shown in Figure 9-19



	A	B	C	D	E	F
1	gNB/eNB Name	MACRO CELL GNB OMNI ANT._10				
2	UE Name	UE_11		500	0	62.42933
3	Channel	PDSCH		490	10	62.42933
4				480	20	56.95417
5	Row Labels	Max of SINR(dB)		470	30	53.0341
6	0	62.42933		460	40	50.09006
7	10	62.42933		450	50	47.75241
8	20	56.95417		440	60	45.81967
9	30	53.034096		430	70	44.17441
10	40	50.090055		420	80	42.74313
11	50	47.752405		410	90	41.47705
12	60	45.819667		400	100	40.34224
13	70	44.174408		390	110	39.31418
14	80	42.743129		380	120	38.37462
15	90	41.477049		370	130	37.50958
16	100	40.342239		360	140	36.70815
17	110	39.314184		350	150	35.96163
18	120	38.374619		340	160	35.26302
19	130	37.509577		330	170	34.60654
20	140	36.708145		320	180	33.98742
21	150	35.961634		310	190	33.40163
22	160	35.263021		300	200	32.84577
23	170	34.606542		290	210	32.31695
24	180	33.987415				
25	190	33.401626				

Figure 9-17: Copying the Distance and SINR values into new cells.

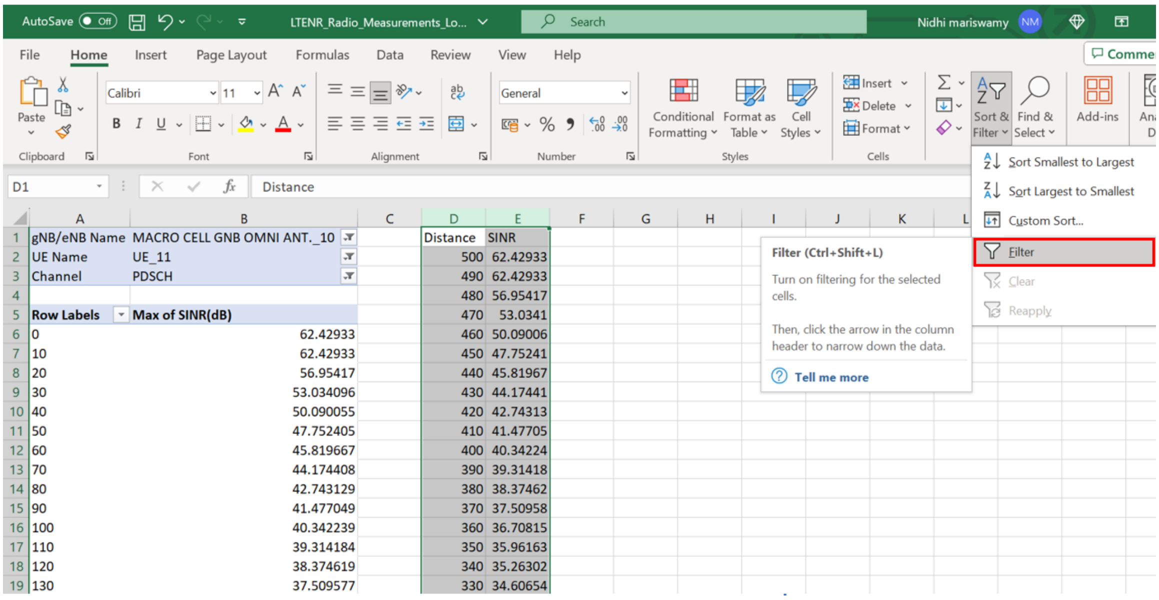


Figure 9-18: Inserting filter.

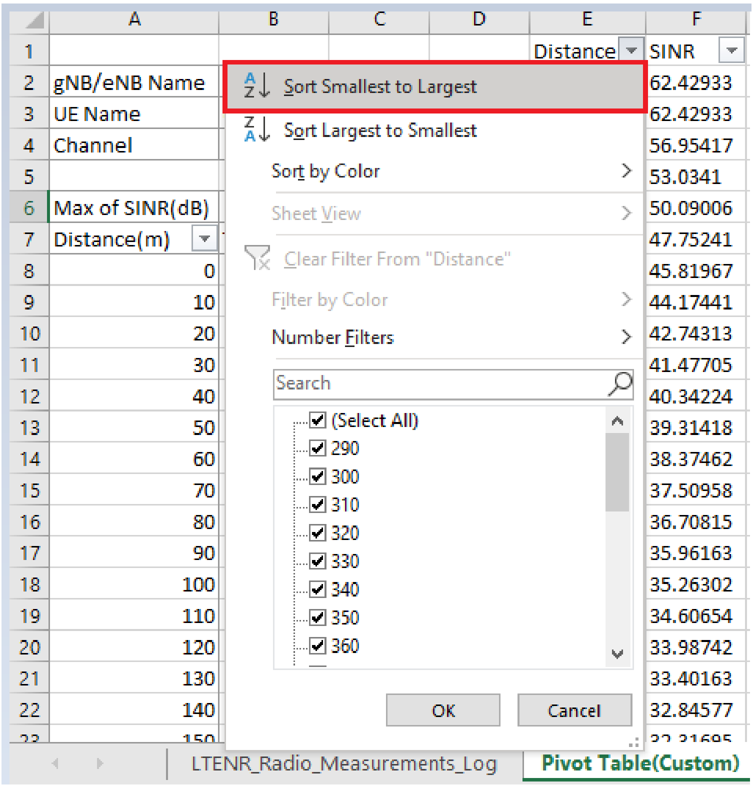


Figure 9-19: Sorting from smallest to largest.

Results

Distance	Max of SNR(dB)
0	69.54
10	69.54
20	64.07
30	60.15
40	57.20
50	54.87

60	52.93
70	51.29
80	49.86
90	48.59
100	47.46
110	46.43
120	45.49
130	44.62
140	43.82
150	43.07
160	42.38
170	41.72
180	41.10
190	40.51
200	39.96
210	39.43
220	38.93
230	38.44
240	37.98
250	37.54
260	37.11
270	36.70
280	36.31
290	35.93

Table 9-7: Downlink SINR values for eNB 4, with ISD = 500m

Distance	Max of SNR(dB)
290	39.43
300	39.96
310	40.51
320	41.10
330	41.72
340	42.38
350	43.07
360	43.82
370	44.62
380	45.49
390	46.43
400	47.46
410	48.59
420	49.86
430	51.29
440	52.93
450	54.87
460	57.20
470	60.15
480	64.07

490	69.54
500	69.54

Table 9-8: Downlink SINR results eNB_5, with ISD = 500m

Case 2: No interference in both base stations with $\eta = 4$

eNB4 > Interface LTE	
Channel Model	
Pathloss model	Log distance
Pathloss exponent	4
Interference Model	
Downlink Interference	No Interference
eNB5 > Interface LTE	
Channel Model	
Pathloss model	Log distance
Pathloss exponent	4
Interference Model	
Downlink interference	No Interference

Table 9-9: Properties set for Case 02.

Set the above property values and simulate the scenario for 12 sec. Tabulate the results obtained from LTENR Radio measurement log in the simulation metrics window.

Case 3: UE to Both Base stations is in $\eta = 2.5$.

eNB 4 > Interface LTE	
Channel Model	
Pathloss model	LOG DISTANCE
Pathloss exponent	2.5
Interference Model	
Downlink Interference	Exact geometric model
eNB 5 > Interface LTE	
Channel Model	
Pathloss model	Log distance
Pathloss exponent	2.5
Interference Model	
Downlink interference	Exact geometric model

Table 9-10: Properties set for Case 03.

Set the above property values and simulate the scenario for 12 sec. Tabulate the results obtained from LTENR Radio measurement log in the simulation metrics window.

Case 4: UE to Both Base stations with $\eta = 4$.

eNB 4 > Interface LTE	
Channel Model	
Pathloss Model	Log distance

Pathloss Exponent	4
Interference Model	
Downlink Interference	Exact geometric model
eNB 5 > Interface LTE	
Channel Model	
Pathloss model	Log distance
Pathloss exponent	4
Interference model	
Downlink Interference	Exact geometric model

Table 9-11: Properties set for Case 04.

Set the above property values and simulate the scenario for 12 sec. Tabulate the results obtained from LTENR Radio measurement log in simulation metrics window.

Results of UE with $\eta = 2.5, and\ 4$

Distance	Case #2 DL SINR (dB)	Case #3 DL SINR (dB)	Case #4 DL SINR (dB)
0	52.77	39.52	52.40
10	52.77	39.30	52.37
20	44.01	33.60	43.57
30	37.74	29.45	37.27
40	33.03	26.28	32.52
50	29.29	23.70	28.73
60	26.20	21.52	25.59
70	23.56	19.63	22.90
80	21.27	17.94	20.55
90	19.25	16.42	18.46
100	17.43	15.01	16.57
110	15.79	13.71	14.84
120	14.28	12.49	13.25
130	12.90	11.33	11.76
140	11.62	10.24	10.37
150	10.42	9.18	9.04
160	9.30	8.17	7.78
170	8.25	7.19	6.57
180	7.26	6.24	5.40
190	6.33	5.31	4.27
200	5.44	4.40	3.15
210	4.59	3.50	2.06
220	3.78	2.61	0.97
230	3.01	1.74	-0.11
240	2.27	0.87	-1.20
250	1.57	0.00	-2.30
260	0.89	-0.87	-3.41
270	0.23	-1.74	-4.54
280	-0.40	-2.62	-5.70

280	3.78	-2.62	0.97
290	4.59	-3.50	2.06
290	4.59	3.50	2.06
300	5.44	4.40	3.15
310	6.33	5.31	4.27
320	7.26	6.24	5.40
330	8.25	7.19	6.57
340	9.30	8.17	7.78
350	10.42	9.18	9.04
360	11.62	10.24	10.37
370	12.90	11.33	11.76
380	14.28	12.49	13.25
390	15.79	13.71	14.84
400	17.43	15.01	16.57
410	19.25	16.42	18.46
420	21.27	17.94	20.55
430	23.56	19.63	22.90
440	26.20	21.52	25.59
450	29.29	23.70	28.73
460	33.03	26.28	32.52
470	37.74	29.45	37.27
480	44.01	33.60	43.57
490	52.77	39.30	52.37
500	52.77	39.52	52.40

Table 9-12: Results for SINR vs. distance for ISD-500m downlink.

The red marks indicate the points of handover.

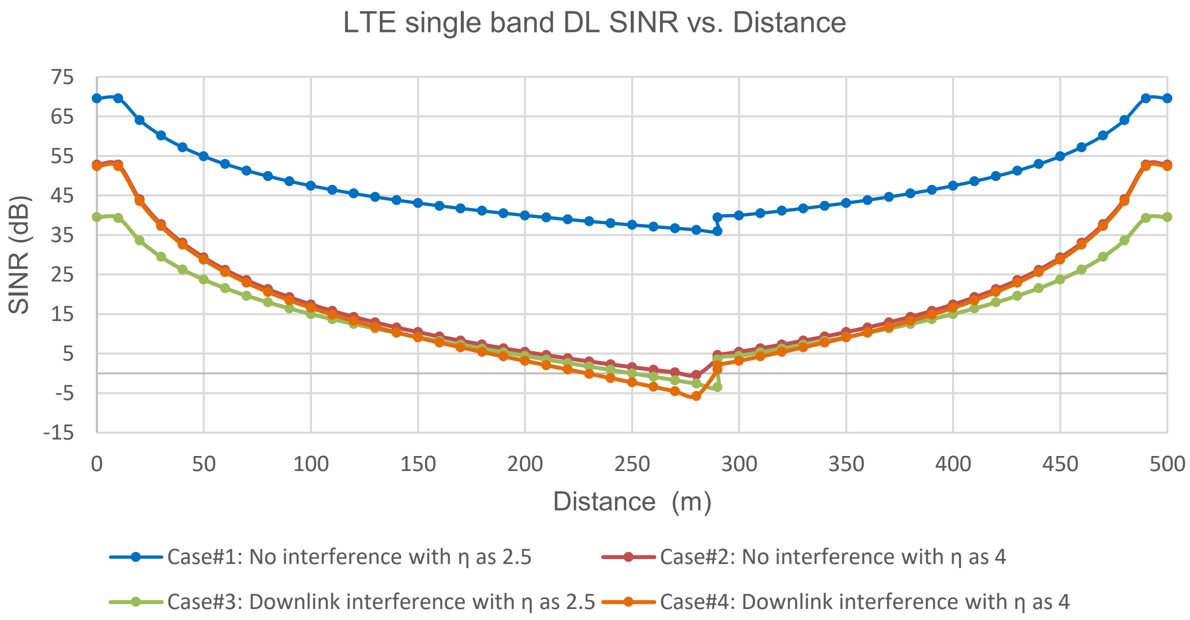


Figure 9-20: Downlink SINR vs. distance plot for BAND42 for different network configurations.

9.2.4 Discussions

Initially (in case 1), the UE is attached to BS_1. In the scenario, the UE moves in a straight line towards BS_2 and at 290 m it is handed over to BS_2. Till 290 m the "desired" signal is from BS_1 while the "interfering" signal is from BS_2. Post-handover there is a reversal; the desired signal is from BS_2 while the interfering signal is from $BS_{1.}$

Signals from BS_1 and from BS_2 to the UE undergo pathloss. If the transmit powers at both BSs are P_t then the $SINR$ works out to be

$$SINR = \frac{P_t \times PL(d)}{N_0 \times W + P_t \times PL(d_{ISD} - d)}$$

Where $PL(d)$ is the pathloss loss (per the 3GPP pathloss models) at a distance of d. Since the distance between the two BSs is equal to d_{ISD}, the inter site distance, the UE is at a distance of $(d_{ISD} - d)$ from the interfering BS, and hence the $PL(d_{ISD} - d)$ term in the denominator. When there is a line-of-sight (LOS) condition with a eNB, the path loss is lower, modelled here by setting the path loss exponent as 2.5. When there is a non-line-of-sight (NLOS) condition with a eNB, the path loss is higher, modelled here by setting the path loss exponent as 4.

With this background, let us look Figure 9-20

- In all cases, we see a constant SINR till 10m because the pathloss equations defined in the standard take effect only from 10m.
- SINR. vs distance is plotted for four cases.
  - Case #1: UE is in LOS with both BSs, interference is turned off
  - Case #2: UE is in NLOS with both BSs, interference is turned off
  - Case #3: UE is in LOS with both BSs, with interference turned on
  - Case #4: UE is in NLOS with both BSs, with interference turned on

  Note: Here, we use the terms LOS for $\eta = 2.5$, and NLOS for $\eta = 4$, for simplicity.

- In case 1 and case 2, the term I in $SINR = \frac{P_r}{N_0 W + I}$ is set to zero. Practically, this means that the two BSs operate in non-overlapping frequency bands. Therefore, $SINR = SNR = \frac{P_r}{N_0 W}$. We see the SNR dropping as the UE moves away from BS_1. At 280/290m, it gets handed over to BS_2, and we see the SNR increasing as the UE moves closer to BS_2. Why is there a "jump" at the handover point? This is because the standards specify that handover should occur only when target-eNB's SINR is offset (3 dB) higher than serving-eNB's SINR. This condition is satisfied at

280m. You are encouraged to think about the question: Why does the standard specify such an offset?

- Next, we observe that the NLOS curve (purple) is lower than the LOS curve (blue). This is because NLOS pathloss is higher than the LOS pathloss.
- In Cases 3, and 4, the observations are similar to the above, but they are lower than their respective counterparts in cases 1 and 2, due to additional interferences which degrade the SINR further. Further, we notice that the two curves of cases 2 and 4 are very close to each other. Why?
  - Open the log files for these cases and observe the following: The pathloss is more pronounced in these cases, and the interference also decays faster than the case (3) case. Hence, the effect of interference is not very pronounced, leading to almost similar performance.
  - **Optional Exercise**: Check whether the gap between the two curves increases by increasing eNB transmit powers (i.e., increasing the interference power).

9.2.5 Understanding the points of handoff

For the sake of exposition, we investigate the point of hand-off for case #1 and case #2 where the interferences are assumed to be absent from the base-stations. This therefore represents a noise limited regime, or a scenario where the two BSs use non-overlapping frequency bands. In such a scenario, as the UE moves from BS_1 towards BS_2, the SNR from BS_1 decreases, while the SNR from BS_2 increases. The point where the SNR from BS_2 is 3 dB higher than that from BS_1 determines the point of handoff. But we observe that between the cases with path loss exponents of 2.5 and 4, the points of handovers are different! Why? Read the discussion below.

9.2.6 Further Discussion

For the sake of generality, we discuss the effect of path loss exponent on handover in a general setting independent of the values obtained in the experiment. Consider the scenario:

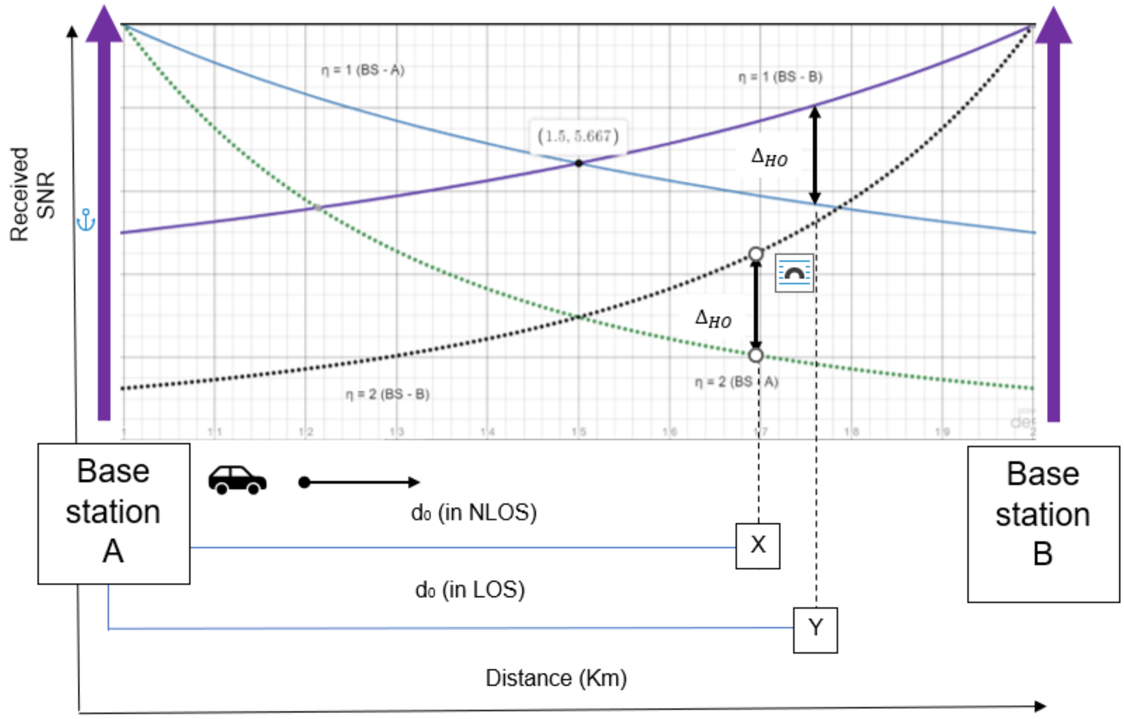


Figure 9-21: Illustration of different handover points in LOS/NLOS cases.

Case A: Noise limited scenario

Let a UE move from base station A towards base station B with inter site distance = 1 km as shown in Figure 9-21. Consider for the moment that we are in a noise-limited regime, where the interferences are assumed to be absent from the base-stations. In such a scenario, as the UE moves from BS A towards BS B, the SNR seen by the UE from BS A decreases, while the SNR seen from BS B increases. For example, let the path loss exponent be set to 1 (η=1), then solid blue curve represents the received SNR as a function of distance seen from BS A, while solid purple curve represents the received SNR as a function of distance seen from BS B. Further, assume that Δ_{HO} is the “handoff threshold” or “handover margin”, i.e., the required SNR difference between the base stations in order to perform a handoff from BS A to BS B. Let the distance at this point be d_0. This point is indicated by point Y in the above plot.

However, when the path loss exponent increases to 2, i.e., η=2, the received SNR from both the BSs changes and the corresponding curves are shown in the above figure using dashed lines. Clearly, due to the differing slopes in the SNR curves at different values of η, **the point at which the handoff occurs is different** (in fact it occurs earlier than the former case) and this point is indicated by X.

Conclusion: The point of handoff is different for environments with different pathloss exponents for a given handoff threshold.

Remarks: Observe from the figure that, as we increase the path loss from 1 to 2, the point of handoff occurs at d = 1.7 Km and d = 1.76 Km, respectively. This difference is more pronounced when the pathloss difference increases.

Case B: Interference limited scenario.

In this case, a similar plot (like above) can be used for handoff analysis, except that y-axis will now have the SINR instead of SNR. It is an exercise for the reader to understand and explain the impact of interference on the handover points in the LOS/NLOS cases.

9.2.7 References

[1] M. Haenggi and R. K. Ganti, "Interference in Large Wireless Networks," 2009.

9.3 Understanding the Impact of MAC Scheduling algorithms on throughput, in a multi-UE scenario (Level 2)

9.3.1 Introduction

Base stations (eNBs) generally deal with multiple mobile stations UEs, some of which require larger bandwidths than others and some of which have better connections (signal quality) than others. In ideal circumstances the base station has plenty of resources (e.g., bandwidth) and each UE gets the resources it needs. However, usually resources are limited, and the base station needs some way of fairly allocating the resources between the UEs.

Consider the downlink of a single eNB 4G cellular system. Several UEs are receiving data from ongoing transfers, for example, TCP controlled file downloads. Assuming that the bottleneck on the transfer path for these connections is this eNB to UE wireless access, the downlink per-UE queues in the eNB will be nonempty. At the beginning of each downlink slot (TTI) the eNB scheduler has to decide which of the UEs' waiting data to transmit in that slot.

At each eNB the MAC scheduler decides the PRB allocation, per carrier, per TTI (slot), in the PDSCH (DL) and in the PUSCH (UL). Control packets such as the buffer status report (BSR) and UL assignment, are assumed to be sent out of band. The resources for transmission of these control packets are part of Overhead as defined in 3.9.21 5G manual.

9.3.1.1 Round Robin Scheduler

It divides the available PRBs among the active flows, i.e., those logical channels which have a non-empty RLC queue. The MCS for each user is calculated according to the received CQIs.

9.3.1.2 Proportional Fair Scheduler

For data transfers, an important performance measure is long term throughput in bits/second, say, $T_i, 1 \leq i \leq n$, where n is the number of UEs. One approach to designing a scheduler is to evaluate the goodness of the throughput vector $(T_1, \ldots, T_n)$ by a network utility, which is the sum of individual user utilities. The utility (or, usefulness) of a throughput T, to a user, increases with increasing throughput, but for large throughputs, increasing throughput further gives diminishing increase in usefulness. This property is modeled as a nondecreasing concave function of throughput. A common measure of utility is the log function, i.e., for the throughput vector $(T_1, \ldots, T_n)$, the utility of throughput T_i to user i is measured as $\ln T_i$. The network utility is, then, given as

$$\sum_{i=1}^{n} \ln T_i$$

A Proportional Fair (PF) scheduler works by scheduling users in slots so that the utility of their long-term throughputs is maximized. In the 4G setting, the scheduling decisions at the beginning of a TTI are based on the physical rates that each UE can get in each Resource Block (RB). If we are given statistical models of these rates, then a nonlinear optimization problem can be formulated and solved to obtain the schedule. This is not a practical approach, however, and a learning algorithm is desired, which, based on slot-by-slot CSI measurements, takes scheduling decisions, which lead to PF optimal throughputs.

The Proportional Fair Scheduler is such a learning scheduler, that uses the throughputs that users are expected to get in the next slot, and the average throughputs they have each obtained up to this slot, to decide which UEs to schedule in the next slot. The practical PF scheme, described below, is based on information such as a presently available data rate for each user in each RB in the next slot (obtained by CSI measurements), and an average data rate over an immediately prior predetermined interval for each user.

9.3.1.3 Implementation

Since NetSim uses a flat fading model, in each slot, each UE achieves the same MCS in every RB in that slot. In other words, different UEs achieve, possibly, different MCSs, but a single UE has the same MCS across all RBs in a slot. Under this assumption, it is optimal to schedule the same UE in every RB in that slot. Since the channel condition can stochastically vary from slot to slot, the MCSs that the UEs achieve will vary from slot to slot. Under this assumption, the following algorithm is Proportional Fair optimal.

Let i, j denote generic users and let t be the slot index. A resource block index k is required given the flat fading assumption. Let $M_i(t)$ be the MCS seen by user i at time (slot) t. The channel CQI (derived from the data channel SINR) is used by the adaptive modulation and coding (AMC) module to determine the MCS. We denote by $S(M, B)$ the TB size in bits for a given MCS, M, and a given number of physical resource blocks (PRBs), B. The achievable rate $R_i(t)$ in bit/s for user i in slot t is defined as

$$R_i(t) = \frac{S(M_i\,(t), 1)}{\tau}$$

where τ is the TTI, i.e., 1 slot duration. At the start of each slot t, the user index $i^*(t)$ - selected by the scheduler - to which required PRBs (per that user's demand) is assigned at time t is determined as

$$i^*(t) = \underset{j=1,\ldots,N}{argmax}\left(\frac{R_j(t)}{T_j(t)}\right)$$

This selection is carried out by the scheduler till all PRBs in slot t are allocated. In the above expression, $T_j(t)$ is the past throughput performance perceived by the user j, and is defined as

$$T_j(t) = \left(1 - \frac{1}{\alpha}\right) T_j(t-1) + \frac{1}{\alpha}\, \hat{T}_j(t)$$

Where α is the time constant (in units of slots) of the exponential moving average. NetSim uses $\alpha = 50$, and $\hat{T}_j(t)$ is the actual throughput achieved by the user i in the subframe t. If $\hat{B}_j(t)$ is the number of PRBs allocated to user j, we finally get

$$\hat{T}_j(t) = \frac{S(M_j(t), \hat{B}_j(t))}{\tau}$$

The value of α can be changed by the user by editing the NetSim's source code; it cannot be changed via the GUI. The PF scheduler thus selects a user having the maximum among values obtained by dividing a present possible data rate by an average data rate during a predetermined interval at every scheduling time point.

9.3.1.4 Max Throughput Scheduler

The Max Throughput (MT) scheduler aims to maximize the overall throughput of the Base station (eNB). It allocates each PRBs to the user that can achieve the maximum achievable rate in the current TTI. The highest achievable rate is calculated by wideband MCS, that is derived from the CQI which in-turn is computed from the SINR. The scheduler allocates the required PRBs to this UE in the current TTI (slot). The calculation of achievable rate is similar to what is explained in PF scheduler.

We denote $S(M, B)$ as the TB size in bits for a given MCS, M, and a given number of physical resource blocks (PRBs), B. The achievable rate $R_i(t)$ in bit/s for user i at slot t is defined as

$$R_i(t) = \frac{S(M_i\,(t), 1)}{\tau}$$

where τ is the TTI i.e., 1 slot duration. At the start of each slot t, the user index $i^*(t)$ - selected by the scheduler - to which required PRBs (per that user's demand) is assigned at time t is determined as

$$i^*(t) = \underset{j=1,\dots,N}{argmax}\left(R_j(t)\right)$$

While MT can maximize cell throughput, it cannot provide fairness to the UEs that experience poor channel condition.

When there are several UEs having the same achievable rate, NetSim implements RR scheduling amongst these UEs that have the same achievable rate.

9.3.2 Network simulation setup

Open NetSim and click on **Experiments> LTE > Scheduling in LTE > Multi UE throughput with UEs at different distances and channel is not time varying** then click on the tile in the middle panel to load the example as shown in below.

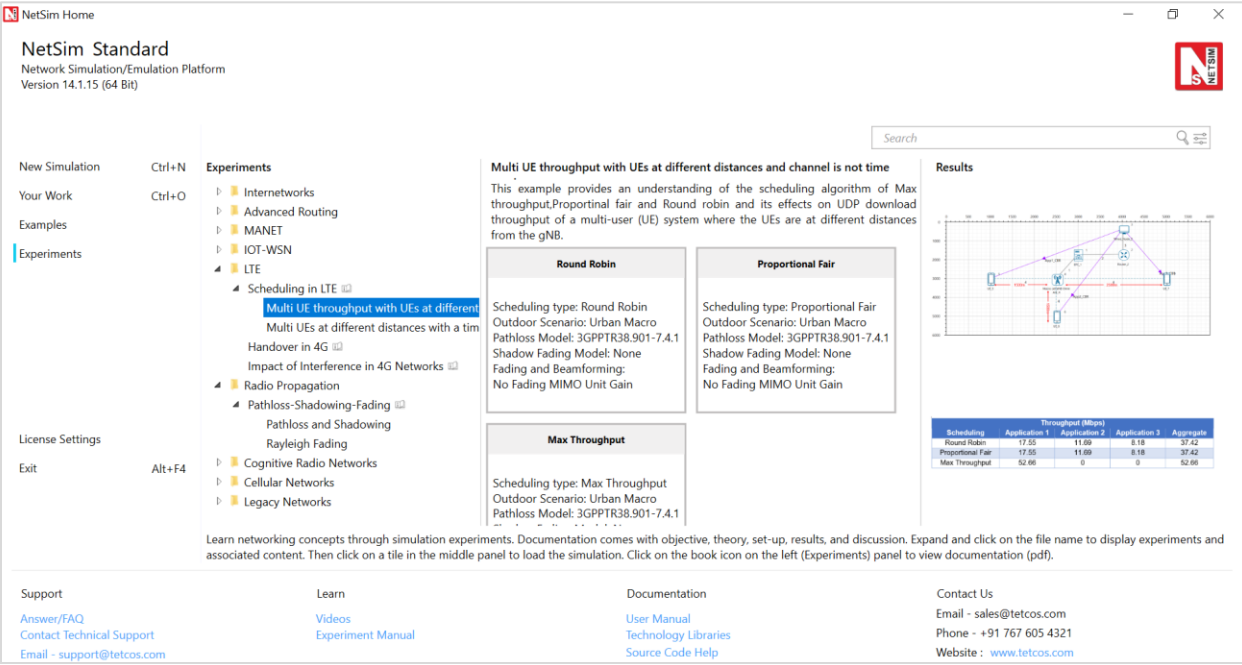


Figure 9-22: List of samples under scheduling in LTE for Multi UE throughput with UEs at different distances and channel is not time varying.

9.3.3 Part I: Multi UE throughput with UEs at different distances and channel is not time varying

In this example we understand how the scheduling algorithm affects the UDP download throughput of a multi-user (UE) system where the UEs are at different distances from the eNB.

The following network diagram illustrates what the NetSim UI displays when you open this example configuration file.

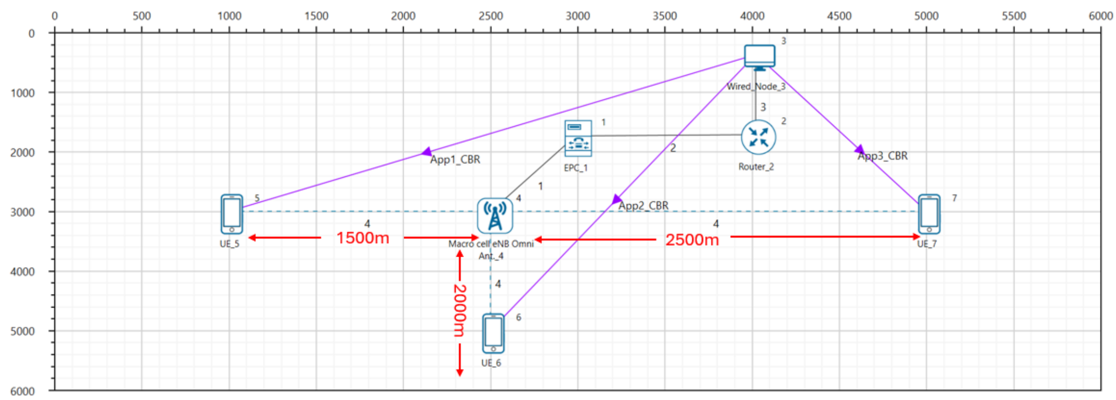


Figure 9-23: Network set up for studying the Scheduling example.

Configuring the scheduling algorithm, and parameter settings in example config files

1. Set grid length as 12000m× 6000m from grid setting property panel on the right. This needs to be done before any device is placed on the grid.

2. Set distance as follows.
   - eNB 4 to UE 5 = 1500m
   - eNB 4 to UE 6 = 2000m, and
   - eNB 4 to UE 7 = 2500m

3. Click on eNB, expand the properties panel on right and go to Interface (LTE), set the following properties as shown below table**.** In the first sample the scheduling type is set to Round Robin, in the second to Proportional fair, and in the third to Max throughput.

Properties	
Data Link Layer Properties	
Scheduling Type	Varies: Proportional Fair, Max throughput, Round Robin
Physical Layer Properties	
CA Type	Single band
CA Configuration	BAND33
CC1	
Numerology	0
Channel Bandwidth	20 MHz
Outdoor Scenario	Urban macro
Channel Model Properties	
LOS NLOS Selection	User defined
LOS Probability	1
Pathloss model	3GPPTR38.901-7.4.1
Shadow Fading Model	None
Fast Fading Model	No Fading

Table 9-13: eNB >Interface (LTE) >Data Link layer and Physical Layer properties

4. Set Tx Antenna Count as 1 and Rx Antenna Count as 1 in eNB properties.
5. Set Tx Antenna Count as 1 and Rx Antenna Count as 1 in all the UEs.

6. Configure the application from the **Set Traffic** tab in the ribbon at the top. Expand the application properties panel on the right and set the following properties as shown below table.

Application Properties			
	Application 1	**Application 2**	**Application 3**
Application Type	CBR	CBR	CBR
Source ID	3	3	3
Destination ID	5	6	7
QoS	UGS	UGS	UGS
Transport Protocol	UDP	UDP	UDP
Packet Size	1460Bytes	1460Bytes	1460Bytes
Inter-arrival time	116.8µs	116.8µs	116.8µs
Start Time	0s	0s	0s

Table 9-14: Application properties

7. Run Simulation for 1.5s and note down throughput value in the results window in each sample. Recall that each sample has a different scheduling algorithm configured.

9.3.3.1 Results and discussions

The results with all the three UEs simultaneously downloading data is as given below.

Throughput (Mbps)				
Scheduling	Application 1	Application 2	Application 3	Aggregate
Round Robin	17.77	11.85	8.29	37.92
Proportional Fair	17.77	11.86	8.28	37.92
Max Throughput	53.33	0	0	53.33

Table 9-15: UDP download throughputs for different scheduling algorithms when all three 3 UEs simultaneously downloading data

Next, consider a scenario with only one of the UEs seeing DL traffic (we don't provide inbuilt configuration file for this, and since it is a simple exercise for a user) First, run for the UE at 1500m, then for UE at 2000m and finally for UE at 2500m. This gives the maximum achievable throughput per node since the eNB resources (bandwidth) is not shared between 3 UEs and is fully dedicated to just one UE. The results are below

Distance from eNB (m)	Application ID	Throughput (Mbps)	Remarks
1500	1	53.33	UE 1 alone has full buffer DL traffic
2000	2	35.55	UE 2 alone has full buffer DL traffic
2500	3	24.88	UE 3 alone has full buffer DL traffic

Table 9-16: UE throughputs if they were run standalone (without the other UEs downloading data)

The PHY rate is decided by the received SNR. Therefore, a UE closer to the eNB will get a higher date rate than a UE further away. In this example the distances from the eNB are such that UE12 Distance > UE11 Distance > UE10 Distance.

In Round Robin PRBs are allocated equally among all three nodes. However, throughputs are in the order UE10 Distance > UE11 Distance > UE12 Distance because of their distances from the eNB. The individual throughputs seen by each of the UEs is exactly $\frac{1}{3}$ of the throughput as shown in Table 9-16.The PF scheduler results will match that of the RR scheduler since the channel is not time varying. In Max throughput scheduling the PRBs are allocated such that the system gets the maximum download throughput. The nearest UE will get all the resources and its throughput will be 3 times the throughput of the UE which got the max throughout in RR.

9.3.4 Part II: Multi UEs at different distances with a time varying channel

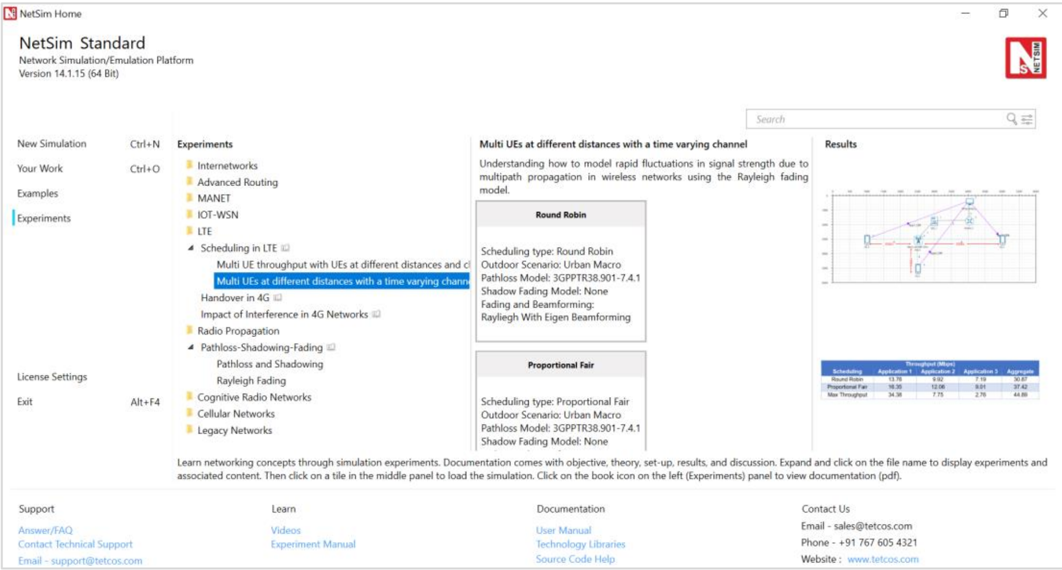


Figure 9-24: List of samples under scheduling in LTE for Multi UEs at different distances with a time varying channel.

Configuring the scheduling algorithm, and parameter settings will remain the same for the case below

eNB properties are as follows.

1. Click on eNB, expand the properties panel on right and go to Interface (LTE), set the following properties as shown below. In the first sample the scheduling type is set to Round Robin, in the second to Proportional fair, and in the third to Max throughput.

Properties	
Data Link Layer Properties	
Scheduling Type	Varies: Proportional Fair, Max throughput, Round Robin

Physical Layer Properties	
CA Type	Single band
CA Configuration	BAND33
CC1	
Numerology	0
Channel Bandwidth	20 MHz
Channel model properties	
Outdoor Scenario	Urban macro
LOS NLOS Selection	User defined
LOS Probability	1
Pathloss model	3GPPTR38.901-7.4.1
Shadow Fading Model	None
Fast Fading Model	Rayleigh
Channel Rank/MIMO Layers	Max Rank
MIMO Beamforming Model	Eigne BF

Table 9-17: eNB >Interface (LTE) >Data Link and Physical layer properties

1. Run Simulation for 1.5s and note down throughput value in the results window in each sample.

9.3.4.1 Results and discussions

The results with all the three UEs simultaneously downloading data are as given below

Throughput (Mbps)				
Scheduling	**Application 1**	**Application 2**	**Application 3**	**Aggregate**
Round Robin	13.51	10.62	7.02	31.15
Proportional Fair	16.99	13.41	9.23	39.63
Max Throughput	32.31	9.82	2.49	44.62

Table 9-18: UDP download throughputs for different scheduling algorithms when all three 3 UEs simultaneously downloading data with time varying channel.

A difference in the performance of the RR and PF schedulers can be seen when the channel is time varying (of the order of the coherence time which is 10ms). To induce time varying randomness in the channel we enable fading and beamforming. Thus, after every 10ms, NetSim draws an i.e. fading random variable, as the additional loss. Under these conditions, the RR scheduler would allot resources to the UEs in a round robin fashion, whereas the PF scheduler would give preference to the UE which sees the best channel (highest SINR). The reason why the RR scheduler yields lower throughputs than the PF scheduler is that the RR scheduler is not "opportunistic," i.e., it does not take advantage of the knowledge that a UE has a good channel in the next slot and continues to serve the UEs cyclically. The results are shown in Table 9-18, observe how this is different from Table 9-15 where the channel is not time varying.

10 5G NR

10.1 MIMO Beamforming in 5G: A start with MISO and SIMO

Objective: Consider 5G communication between a gNB and single UE, over a fading channel. Setup the simplest MIMO cases, namely MISO and SIMO, and investigate the questions:

- How does beamforming gain vary with antenna count?
- How does throughput vary with antenna count?

Theory: Multiple input multiple output (MIMO) is a method for increasing the capacity of the wireless channel using multiple transmitting and receiving antennas. Multiple antennas exploit the spatial dimension, i.e., multiple paths from transmitter to receiver, under suitable spacing within the antenna array, on each side, and channel scattering conditions.

Consider a $N_t \times N_r$ MIMO system where N_t is the number of transmit antennas and N_r is the number of receive antennas. The simplest MIMO instantiations are when:

- $N_r = 1$, a special case where the MIMO system reduces to a Multiple Input Single Output (MISO) channel, and
- Reciprocally when, $N_t = 1$, a special case where the MIMO system simplifies into a Single Input Multiple Output (SIMO) channel.

In both SIMO and MISO, the number of layers (i.e., spatial streams with independent data) is $min(N_t, N_r)$, which equals 1.

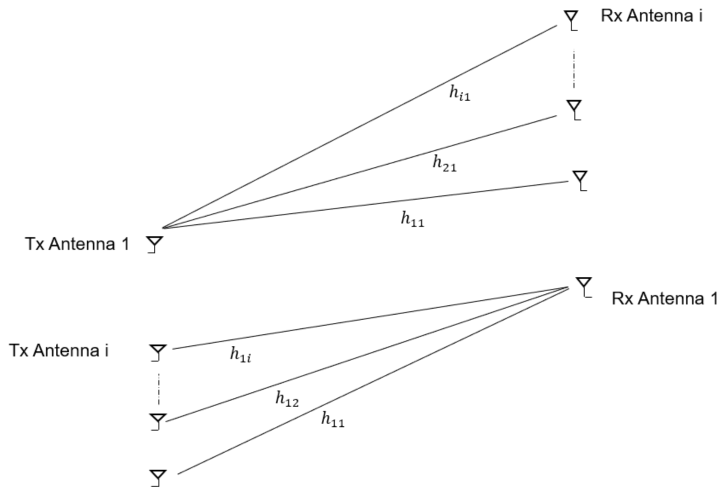


Figure 10-1: Top) Single transmit antenna and multiple receive antennas. Bottom) Multiple Transmit and single receive antenna. In both, h_{ij} represents the channel between the i^{th} receive antenna and the j^{th}transmit antenna.

SIMO: SIMO occurs where the transmitter has a single antenna, and the receiver has multiple antennas. The signal received on multiple antennas is combined in order to maximize an appropriate metric. For example, when the goal is to maximize the received SNR, under additive white Gaussian noise, the optimal receiver is called maximal ratio combining. In the case of fading channels (e.g., with Rayleigh fading, explained in the next section), the channels between the transmitter and the different receive antennas is modelled as independent and identically distributed with unit variance entries; this is shown in Figure 10-1 above. In this case, maximal ratio combining uses the channel coefficients as the weights to combine the signals, and in turn, this provides receive diversity gain. The average SNR at the receiver improves by $10 \log_{10}(N_r)$. However, the improvement in SNR is not exactly the same for every channel instantiation since the channel is random. In this experiment, we will quantify the improvement in the data rate as we vary N_r .

MISO: The phase of the signal from each of the transmit antennas is adjusted so that they add constructively at the receiver, yielding an N_t fold power gain on average. As with SIMO, the instantaneous SNR gain, however, will differ from N_t since the channel is random. So, the question is, given a choice between having multiple antennas at either the transmitter or the receiver, which option yields better improvement in the throughput? Or will they be the same? Note that, with a single-antenna receiver, only one data stream is transmitted, and therefore multiple antennas offer only a diversity gain, but not a multiplexing gain. Now, practically speaking, is it better to have multiple antennas at the base station or at the user? In terms of antenna placement, there is more space to install antennas at the base station. In addition, the signal processing capability of the base station is much higher than that of a mobile phone. Thus, multiple antennas at the base station are easier to implement than multiple antennas at the mobile phone.

The Rayleigh Fading Channel: For a transmitter (gNB) with N_t antennas and a receiver with N_r antennas, the $N_r \times N_t$ baseband channel gain matrix (to model fading between every transmit-receive antenna pair) has complex Gaussian distributed elements. The standard model (under the assumption of Rayleigh fading) is that the complex elements are statistically independent across antennas, and each element is a circularly symmetric complex Gaussian distributed with zero mean and unit variance. We denote this matrix by H.

For the channel matrix H defined as above, consider the complex Wishart Matrix defined as follows:

$$W = H\,H^{\dagger} \quad r < t,$$

$$W = H^{\dagger}H \quad r \geq t$$

Therefore, letting $m = \min(r,t),\ W$ is an $m \times m$ nonnegative definite matrix, with eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_L > 0 = \lambda_{L+1} = \cdots = \lambda_m$. It is these eigenvalues that determine the gains in the parallel SISO models that arise from eigen-beamforming at the transmitter and receiver.

NetSim permits the user to enable or disable a stochastic fading model. Fading is modelled by the elements of H being time varying, with some coherence time. Such time variation results in the eigenvalues of W also to vary over time. NetSim models such time variation by letting the user define a coherence time during which the eigenvalues are kept fixed. For each (r,t) value, NetSim maintains a list of samples of eigenvalues for the corresponding Wishart matrix.

10.1.1 Network Simulation set up

Open NetSim and click on **Experiments > 5G NR > MIMO Beamforming in 5G > Multiple input single output (MISO) in 5G** then click on the tile in the middle panel to load the example as shown in below.

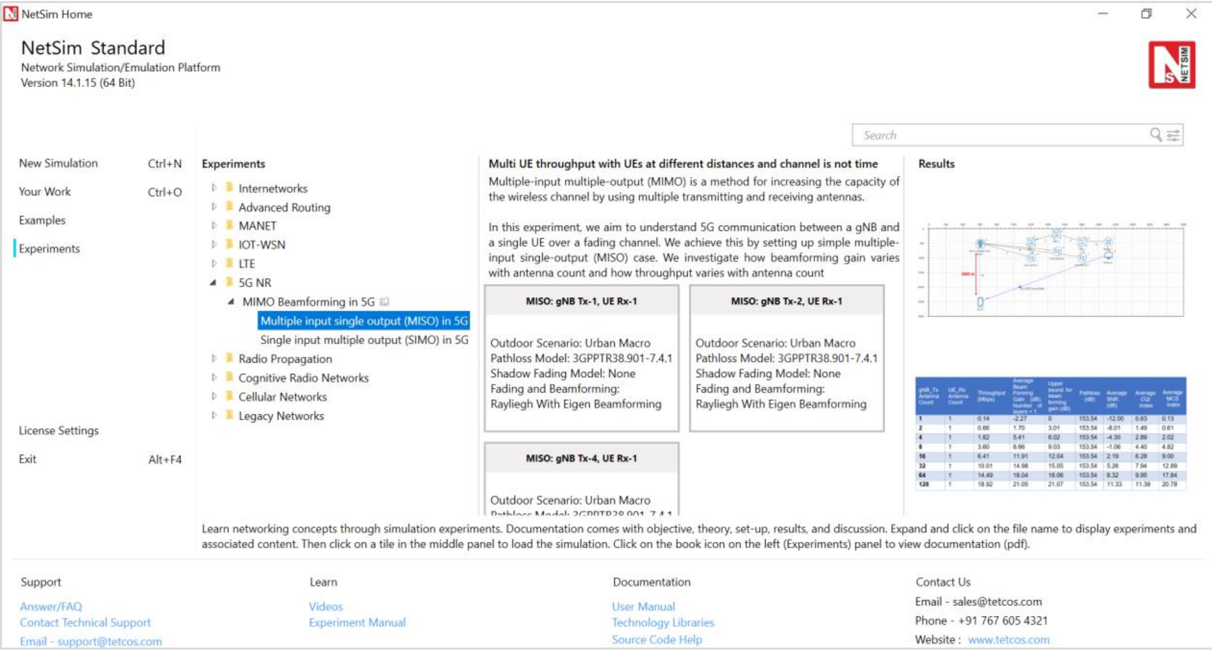


Figure 10-2: List of samples under MIMO Beamforming in 5G for MISO.

10.1.2 Network Scenario:

NetSim UI would display the network topology shown in the screenshot below when you open the example configuration file.

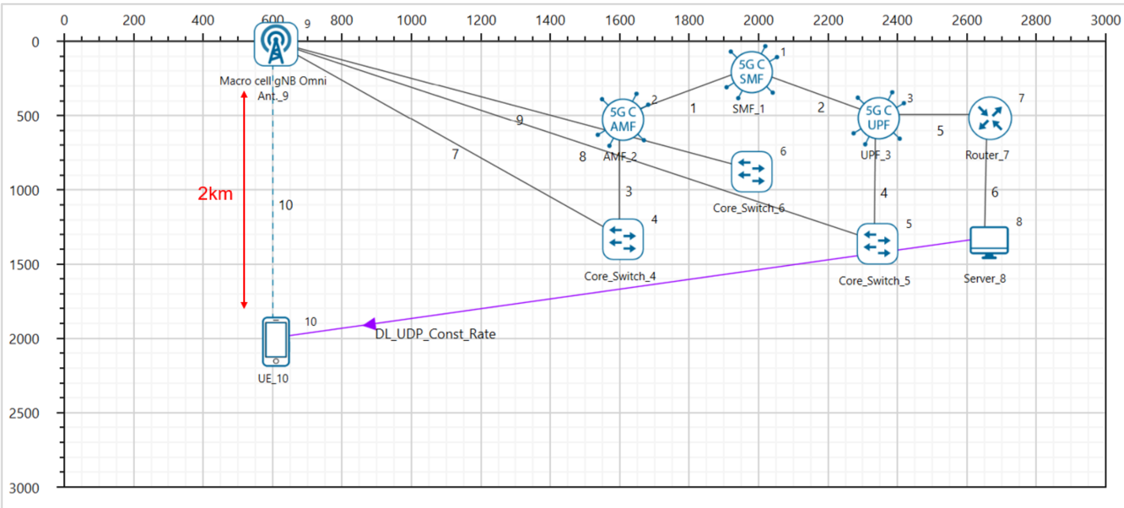


Figure 10-3: Network topology in this experiment.

10.1.3 Part 1- MISO. Network Configuration

The following parameters were configured in the network setup:

1. The gNB- Interface 5G RAN were set with the following properties:

gNB- Interface 5G RAN Parameters	
gNB Height	10m
Tx Power	40 dBm
Duplex Mode	TDD

CA Type	Single band
CA Configuration	n78
Component Carrier 1	
DL: UL Ratio	4:1
Numerology	0
Channel Bandwidth (MHz)	10
Antenna	
Tx Antenna Count	Varied from 1 to 128
Rx Antenna Count	1
PDSCH and PUSCH Configuration	
MCS Table	QAM64
CSI Report Configuration	
CQI Table	Table1
Channel Model	
Pathloss model	3GPPTR38.901-7.4.1
Outdoor Scenario	Urban macro
LOS NLOS Selection	User defined
LOS Probability	0 (NLOS)
Shadow Fading Model	None
Fast Fading Model	Rayleigh
Channel rank/MIMO Layers	Max Rank
MIMO Beamforming Model	Eigen BF
Coherence Time (ms)	10
Additional Loss Model	None

Table 10-1: gNB properties

2. The UE properties were configured with the following parameters:

UE Interface 5G RAN	
Physical Layer Properties	
UE Height	1.5m
Tx Power	23 dBm
Antenna	
Tx Antenna Count	1
Rx Antenna Count	1

Table 10-2: UE properties

3. The wired link speed was set to 10 Gbps and the Uplink and Downlink BER were set to 0 in the wired links.
4. A downlink CBR application was configured from wired node to UE with Transport protocol as UDP and Packet Size of 1460 Bytes and Inter Arrival time of 179.69 µs and the Start Time was set to 1s.
5. Click on the Configure report tabs and then click on the Plots icon in the toolbar to enable LTENR Radio measurements under Network Logs and Beamforming Gain vs. Time in LTENR Radio Measurements as shown in Figure 10-4 and Figure 10-5.

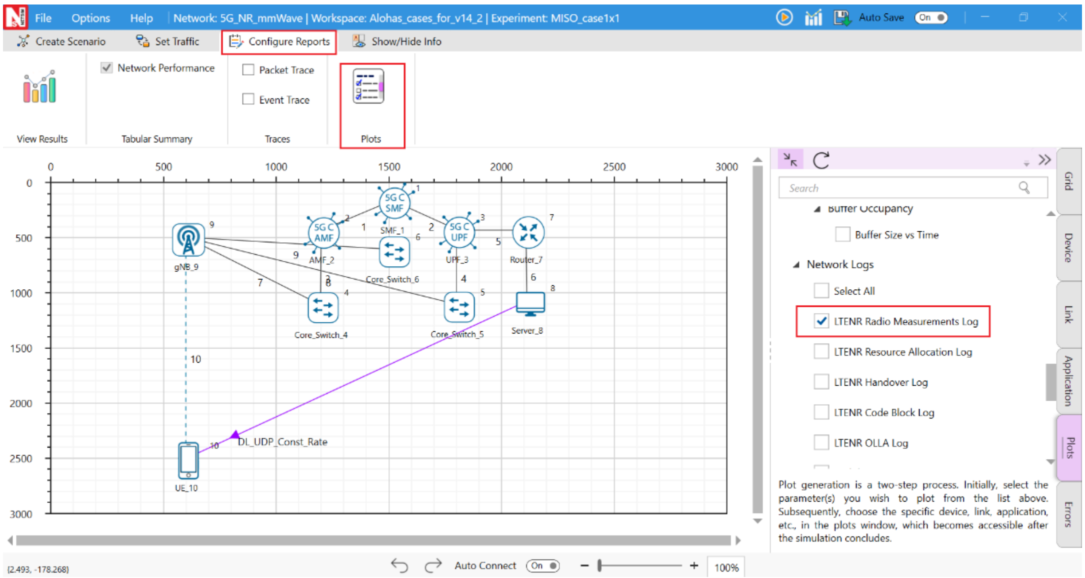


Figure 10-4: Enabling the LTENR Radio measurement log.

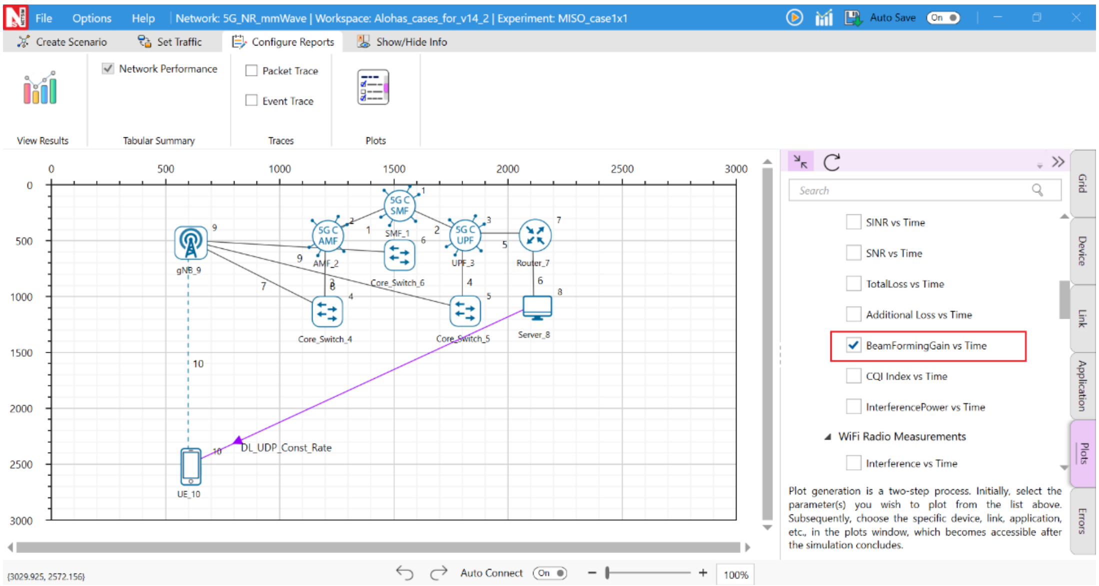


Figure 10-5: Enabling Beamforming Gain vs Time plot.

6. Run simulation for 10s, note down the Application Throughput obtained from the Application Metrics table in the NetSim Results dashboard. Similarly, observe the average Beamforming Gain in dB obtained for the DL application from the generated NetSim plot.

10.1.4 Part 2- SIMO. Network Configuration:

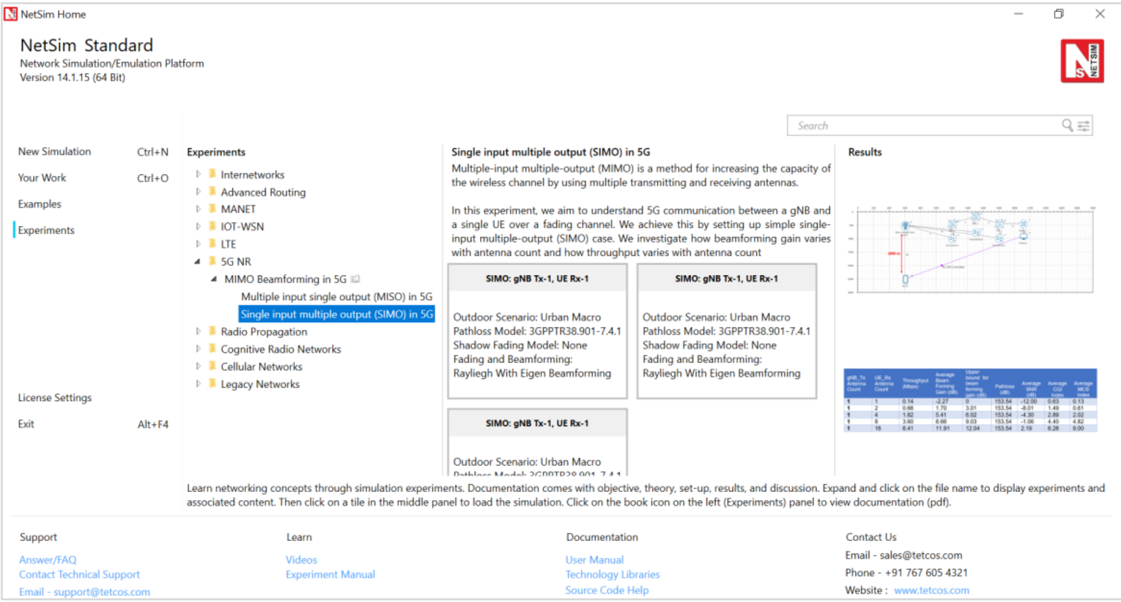


Figure 10-6: List of experiments under MIMO Beamforming in 5G for SIMO

1. Set all the properties same as part 1- MISO.
2. Set the Tx Antenna count in 5G RAN interface of gNB to 1.
3. Vary the Rx Antenna count in 5G RAN interface of UE from 1 to 16.
4. Run simulation for 10s.
5. After the simulation, note down the Application Throughput obtained from the Application Metrics table in the NetSim Results dashboard. Similarly, observe the average Beamforming Gain in dB obtained for the DL application from the generated NetSim plot.

10.1.5 Simulation Output:

Steps to calculate the Throughput, Beamforming Gain, SNR, Pathloss and CQI Index:

1. After the simulation, open NetSim Result dashboard and note down the throughput from the Application Metrics Table as shown below:

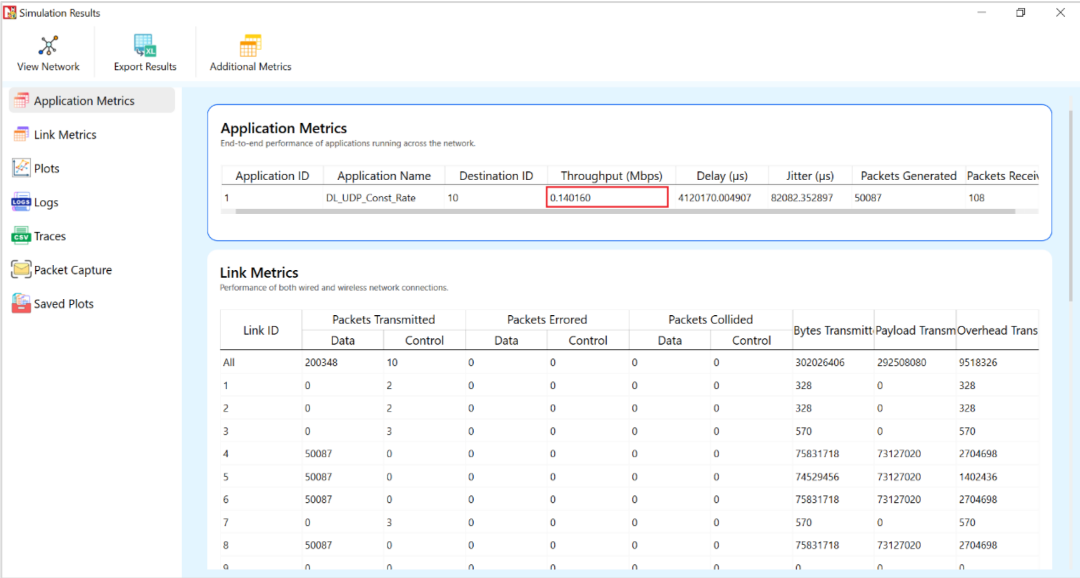


Figure 10-7: NetSim Results window showing Application Throughput obtained after the simulation.

1 In the results window, click on the Logs option in the left panel and select LTENR Radio Measurements Log.csv file.

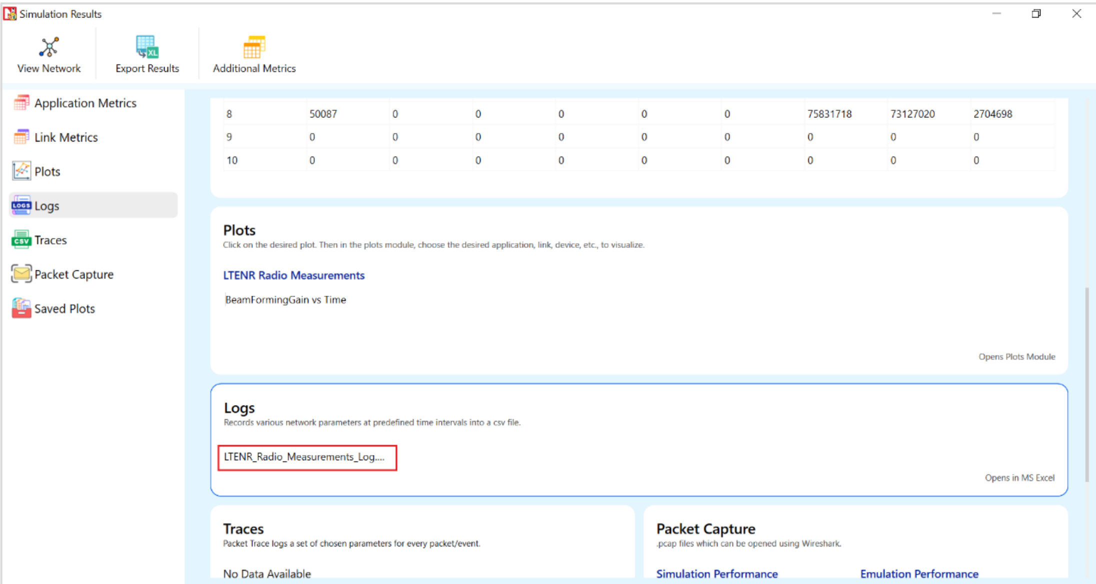


Figure 10-8: NetSim Results window showing access to log file generated.

2. This will open the csv file which logs the parameters beamforming gain, CQI and MCS Indexes, Pathloss etc. over time as shown below.

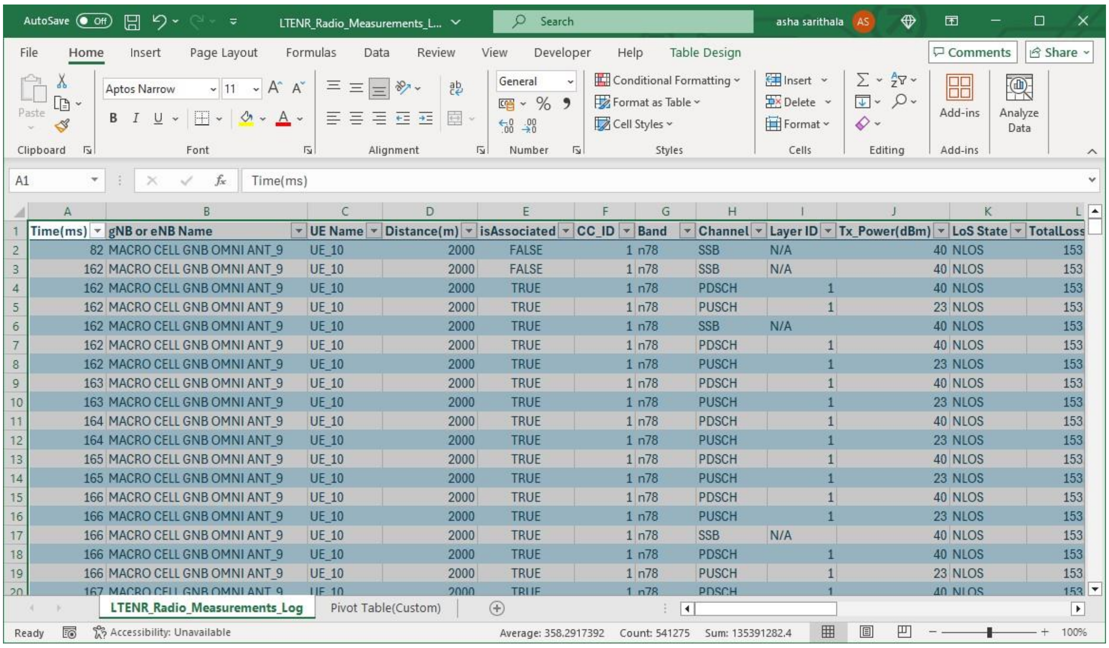


Figure 10-9: LTENR Radio measurement log file generated post simulation.

3. Filter the Channel to only PDSCH and click on ok since we have considered a DL application from server to UE.

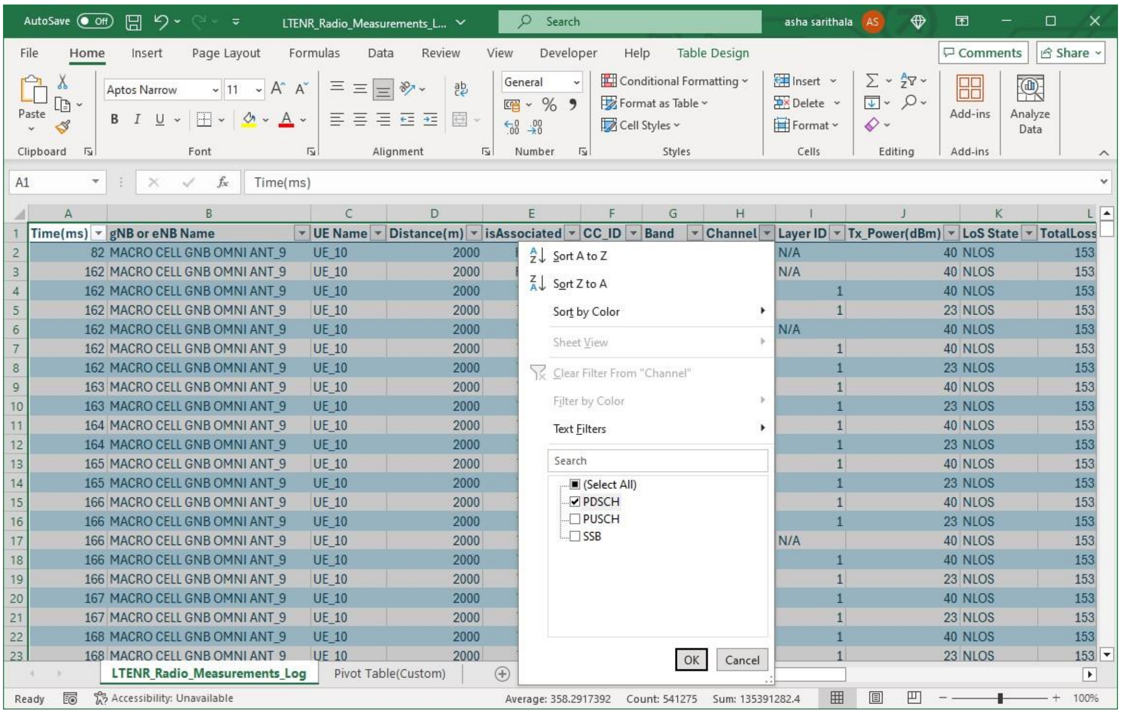
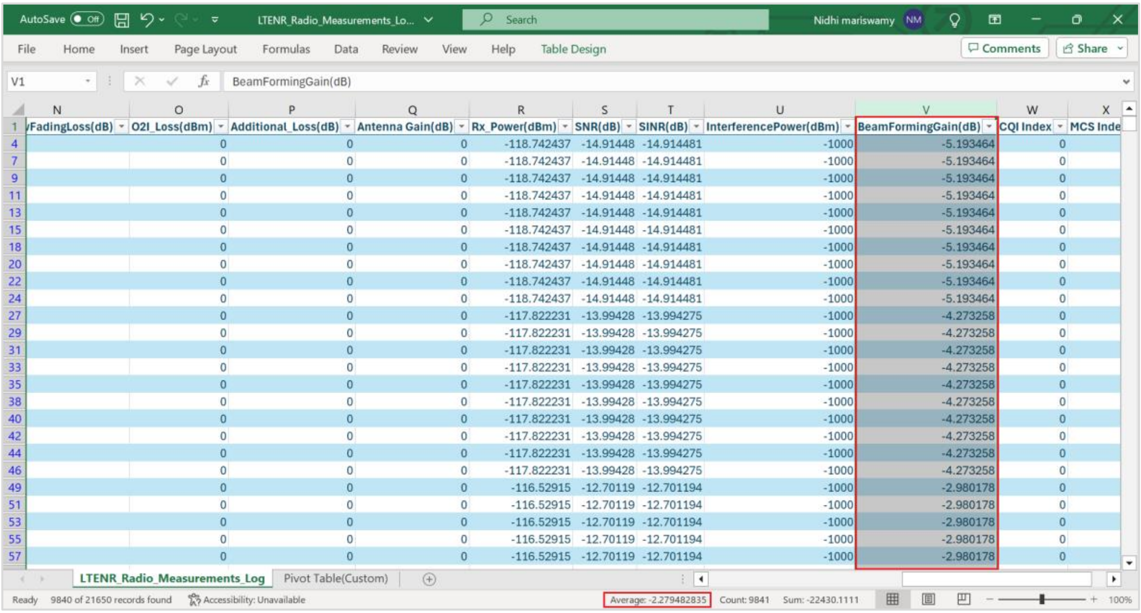


Figure 10-10: LTENR Radio measurement log file showing the filtering process of PDSCH/PUSCH column

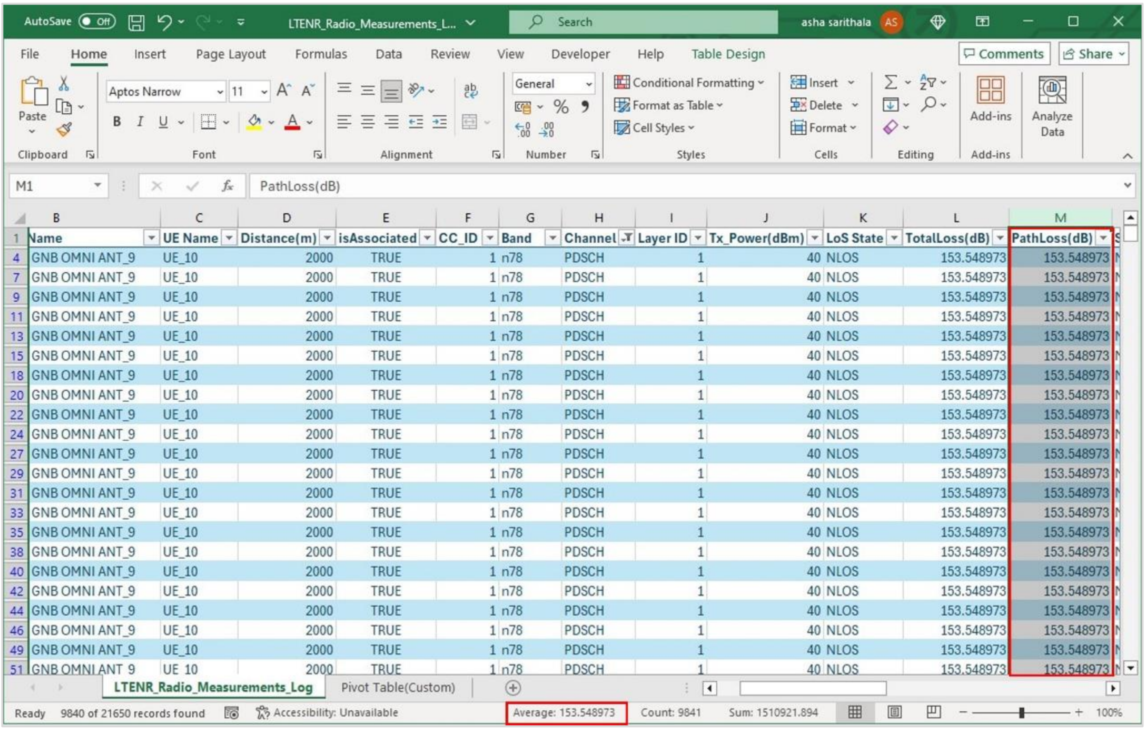
4. Now select the Beamforming Gain column and note down the average Beamforming Gain in dB per layer.



Row	FadingLoss(dB)	O2I_Loss(dBm)	Additional_Loss(dB)	Antenna Gain(dB)	Rx_Power(dBm)	SNR(dB)	SINR(dB)	InterferencePower(dBm)	BeamFormingGain(dB)	CQI Index	MCS Inde
4		0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	
7		0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	
9		0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	
11		0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	
13		0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	
15		0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	
18		0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	
20		0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	
22		0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	
24		0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	
27		0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	
29		0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	
31		0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	
33		0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	
35		0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	
38		0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	
40		0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	
42		0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	
44		0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	
46		0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	
49		0	0	0	-116.52915	-12.70119	-12.701194	-1000	-2.980178	0	
51		0	0	0	-116.52915	-12.70119	-12.701194	-1000	-2.980178	0	
53		0	0	0	-116.52915	-12.70119	-12.701194	-1000	-2.980178	0	
55		0	0	0	-116.52915	-12.70119	-12.701194	-1000	-2.980178	0	
57		0	0	0	-116.52915	-12.70119	-12.701194	-1000	-2.980178	0	

Figure 10-11: LTENR Radio measurement log file showing average Beamforming Gain obtained.

5. Select the Pathloss column and note down the average pathloss value obtained.



Row	Name	UE Name	Distance(m)	isAssociated	CC_ID	Band	Channel	Layer ID	Tx_Power(dBm)	LoS State	TotalLoss(dB)	PathLoss(dB)
4	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
7	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
9	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
11	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
13	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
15	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
18	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
20	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
22	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
24	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
27	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
29	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
31	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
33	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
35	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
38	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
40	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
42	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
44	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
46	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
49	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973
51	GNB OMNI ANT_9	UE_10	2000	TRUE	1	n78	PDSCH	1	40	NLOS	153.548973	153.548973

Figure 10-12: LTENR Radio measurement log file showing average Pathloss obtained.

6. In the same way, select the SNR column and note down the average SNR obtained.

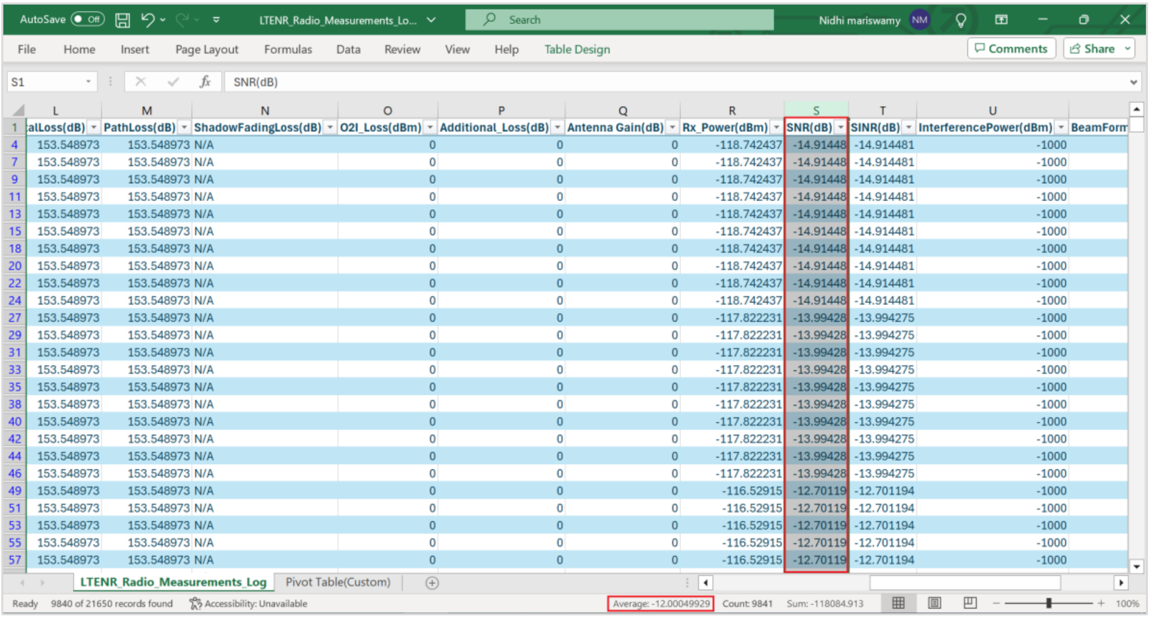


Figure 10-13: LTENR Radio measurement log file showing average SNR obtained

7. Similarly, calculate the average CQI Index and MCS Index.

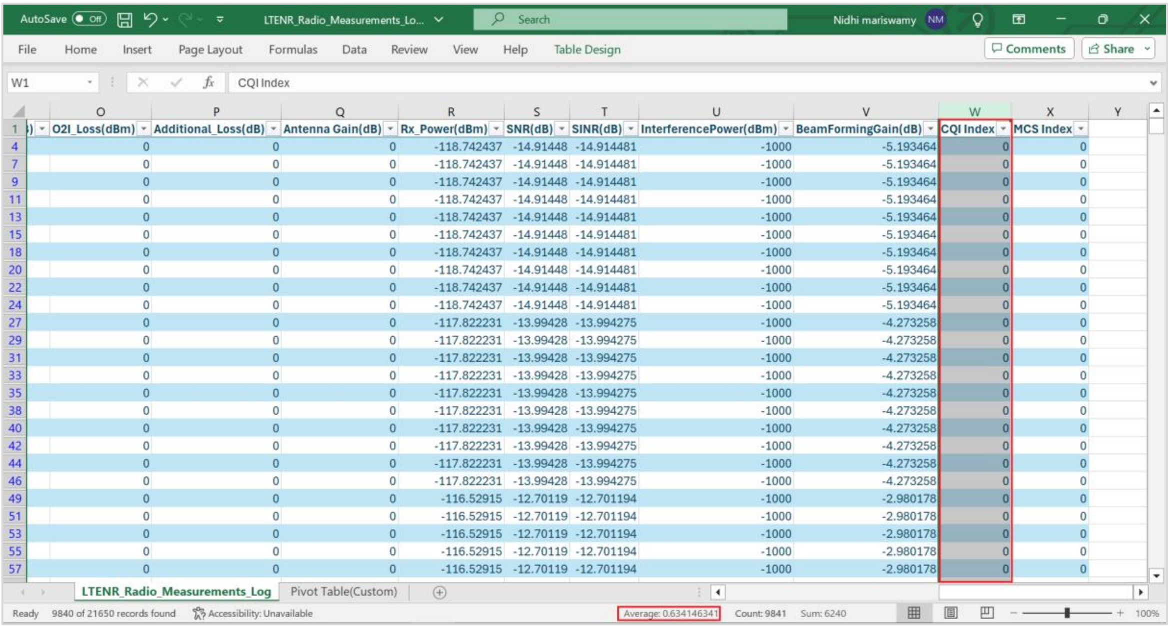
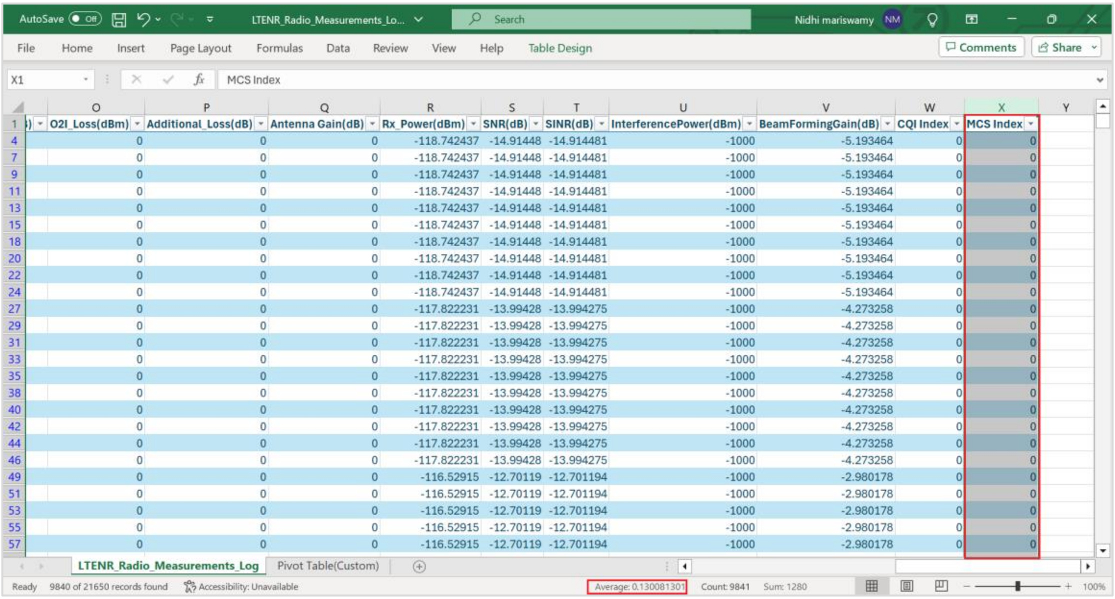


Figure 10-14: LTENR Radio measurement log file showing average CQI Index obtained.



	O2I_Loss(dBm)	Additional_Loss(dB)	Antenna Gain(dB)	Rx_Power(dBm)	SNR(dB)	SINR(dB)	InterferencePower(dBm)	BeamFormingGain(dB)	CQI Index	MCS Index
4	0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	0
7	0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	0
9	0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	0
11	0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	0
13	0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	0
15	0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	0
18	0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	0
20	0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	0
22	0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	0
24	0	0	0	-118.742437	-14.91448	-14.914481	-1000	-5.193464	0	0
27	0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	0
29	0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	0
31	0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	0
33	0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	0
35	0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	0
38	0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	0
40	0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	0
42	0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	0
44	0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	0
46	0	0	0	-117.822231	-13.99428	-13.994275	-1000	-4.273258	0	0
49	0	0	0	-116.52915	-12.70119	-12.701194	-1000	-2.980178	0	0
51	0	0	0	-116.52915	-12.70119	-12.701194	-1000	-2.980178	0	0
53	0	0	0	-116.52915	-12.70119	-12.701194	-1000	-2.980178	0	0
55	0	0	0	-116.52915	-12.70119	-12.701194	-1000	-2.980178	0	0
57	0	0	0	-116.52915	-12.70119	-12.701194	-1000	-2.980178	0	0

Figure 10-15: LTENR Radio measurement log file showing average MCS Index obtained.

10.1.5.1 Results

MISO: Varying Tx Antenna count in the gNB and 1 Rx Antenna in the UE

gNB_Tx Antenna Count	UE_Rx Antenna Count	Throughput (Mbps)	Average Beam Forming Gain (dB). Number of layers = 1	Upper bound for beam forming gain (dB)	Pathloss (dB)	Average SNR (dB)	Average CQI Index	Average MCS Index
1	1	0.46	-2.68	0	153.55	-12.40	0.59	0.19
2	1	1.12128	1.80	3.01	153.55	-7.92	1.51	0.75
4	1	2.268516	5.46	6.02	153.55	-4.26	2.92	2.45
8	1	4.212587	8.76	9.03	153.55	-0.96	4.46	5.42
16	1	7.097547	11.89	12.04	153.55	2.17	6.26	9.64
32	1	10.93378	14.99	15.05	153.55	5.26	7.94	13.40
64	1	15.39684	18.02	18.06	153.55	8.30	9.93	18.23
128	1	20.29854	21.06	21.07	153.55	11.33	11.39	21.38

Table 10-3: NetSim simulation output showing Throughput, Average beamforming gain and the upper bound (from Jensen's inequality) on the beamforming gain for a $N_t \times 1$ channel.

SIMO: Varying Rx Antenna count in the UE and 1 Tx Antenna in the gNB

gNB_Tx Antenna Count	UE_Rx Antenna Count	Through put (Mbps)	Average Beam Forming Gain (dB).	Upper bound for beam forming gain (dB)	Pathloss (dB)	Average SNR (dB)	Average CQI Index	Average MCS Index
1	1	0.46	-2.68	0	153.55	-12.40	0.59	0.19
1	2	1.12	1.79	3.01	153.55	-7.92	1.51	0.75
1	4	2.26	5.46	6.02	153.55	-4.26	2.92	2.45
1	8	4.21	8.77	9.03	153.55	-0.96	4.46	5.42
1	16	7.09	11.88	12.04	153.55	2.17	6.26	9.64

Table 10-4: NetSim simulation output showing Throughput, Average beamforming gain and the upper bound (from Jensen's inequality) on the beamforming gain for a $1 \times N_r$ MIMO channel. N_r is limited to 16 since this is the maximum antenna count supported in UEs in NetSim.

10.1.5.2 Beamforming Gain Plot

Open the beamforming gain plot from the simulation result window and disable the Accelerate Plotting and filter the channel to PDSCH and layer ID to 1 and click on plot and observe as following.

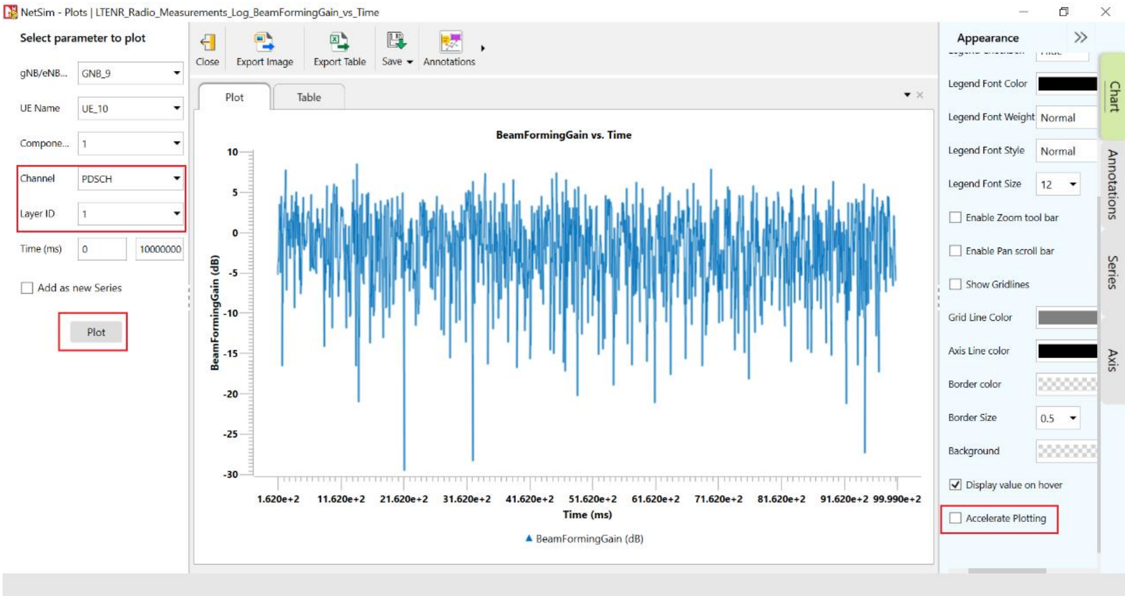


Figure 10-16: NetSim Beamforming Gain vs time plot showing variation in beamforming gain (dB) over the course of simulation. The beamforming gain changes every "coherence time".

10.1.5.3 Discussion

From the tabulated results, we observe:

- An increase in beamforming gains as
  - N_t increases in the MISO case, and as
  - N_r increases in the SIMO case.
- The beamforming gain when N_t varies (with N_r fixed) is precisely the same as when N_r varies (with N_t fixed).

Next, we turn to the question a network engineer would be interested in: how does beamforming impact throughput? While link level simulators may perform the beamforming computations and provide the SNR at a link level, the power of a "system" level simulator like NetSim lies in its ability to compute the impact of link level factors (such as beamforming) on the system (or network). These computations are explained in an earlier experiment.

Since the distance between the gNB and UE is fixed, the common pathloss for all Tx-Rx antenna pairs is the same. This common pathloss value is factored (or "pulled out") from the individual N_t - N_r path loss calculations. With this factorization done, the only parameter affecting SNR is the channel fading, the effect of which shows up in the output as beamforming gain. Quite simply as the (average) beamforming gain increases, the (average) SNR proportionally increases. Notice that every time the antenna count is doubled the SNR increases *by ≈ 3 dB* (which matches intuition). An increase in SNR improves the channel quality (the CQI), and thereby a higher modulation and coding scheme (MCS) is chosen for data transmission.

Remark. Note that these are “average” arguments: in practice, since the fading coefficients are random, one does not obtain a 3 dB improvement by doubling the number of antennas for every channel instantiation. Consequently, the improvement in the spectral efficiency (the reader should study and understand this terminology) is not exactly 1 bit/s/Hz on average. In other words, there is a difference between using the average SNR for computing the data rate versus computing the average data rate by averaging the rate obtained across different channel instantiations. The reader should carry out experiments with different values of N_t and observe the variation in the rate obtained and understand this phenomenon.

Continuing from before the remark, from the MCS the PHY rate is calculated via the procedure for TBS determination per the 3GPP standard. Without getting into the details of these computations, the simplistic inference is that higher MCS leads to higher throughputs.

And finally, the underlying mathematics. The beamforming gains (in linear scale) are the Eigen values of the Wishart matrix. In the MISO and SIMO cases the Wishart matrix has just one element, which itself is the eigenvalue, i.e., the beamforming gain is

$$\mu = E(\lambda) = \sum_{i=1}^{N} |h_i|^2$$

Where h_i are the elements of the Wishart matrix, and $N = N_t$ or $N = N_r$, as the case may be. Since $\mathbb{E}|h_i|^2 = 1$,

$$\mu \coloneqq \mathbb{E}(\lambda) = \begin{cases} N_t & for\ a\ N_t \times 1\ MIMO\ system \\ N_r & for\ a\ 1 \times N_r MIMO\ system \end{cases}$$

Since the standard deviation of an exponentially distributed random variable is the square of its mean, and since the $|h_i|\ 1 \leq i \leq N,$ are independent,

$$VAR\ (\lambda) = \begin{cases} N_t & for\ a\ N_t \times 1\ MIMO\ system \\ N_r & for\ a\ 1 \times N_r\ MIMO\ system \end{cases}$$

However, the beamforming gains output by NetSim are in dB (log) scale. How does one analytically verify its correctness? The answer lies in Jensen’s inequality. Since the log function is concave, Jensen’s inequality leads to

$$E \log_{10}(\lambda) \leq \log_{10}(E(\lambda))$$

Here λ is the eigen value of the Wishart matrix, and $10\log_{10}\lambda$ is the beamforming gain in dB scale. Therefore, the beamforming gains (in the dB domain) are bounded as

$$BFGain\ (dB) \leq 10\log_{10}(E(\lambda))$$

$$E(\lambda) = N$$

$$BFGain\ (dB) \leq 10\log_{10} N$$

In this experiment (and in NetSim), the number of antennas, N, is of the form 2^p ,where $p = 0, 1, 2 \ldots$ and therefore the upper bound on the beam forming gain is

$$BFGain\ (dB) \leq 10 \times p\log_{10} 2 \leq 3.01 \times p$$

Exercises:

1. Quantify the improvement in data rate as a function of N_t (MISO) and N_r (SIMO) in
   a. Low SNR case (SNR << 1), and
   b. High SNR case (SNR >> 1).
2. (For the Instructor or TA) Assign a set of personalized questions that will require each student to run the simulator and generate the results needed to write their reports. For example, different distances between the gNB and UE (which will vary the path loss), different Tx powers, different ranges for N_t and N_r, etc.

10.2 Throughput and fairness of 5G scheduling algorithms in a complex network environment

In the previous experiment, understood the working of the Round Robin, Proportional Fair, and Max CQI (Max Throughput) scheduling algorithms and evaluated their performance in a scenario with one gNB and three UEs under full buffer traffic conditions. The current experiment examines a multi-gNB, multi-UE environment with full buffer and non-full buffer traffic scenarios. It also incorporates user mobility and channel fading effects to model the dynamic wireless communication environment.

The goal is to analyze the performance of these scheduling algorithms in practical 5G network deployments where users move, the channel quality varies, and traffic demands fluctuate.

10.2.1 Network Setup

Open NetSim and click on **Experiments > 5G NR >Throughput and fairness of 5G scheduling algorithms in a complex network environment** then click on the tile in the middle panel to load the example as shown in below.

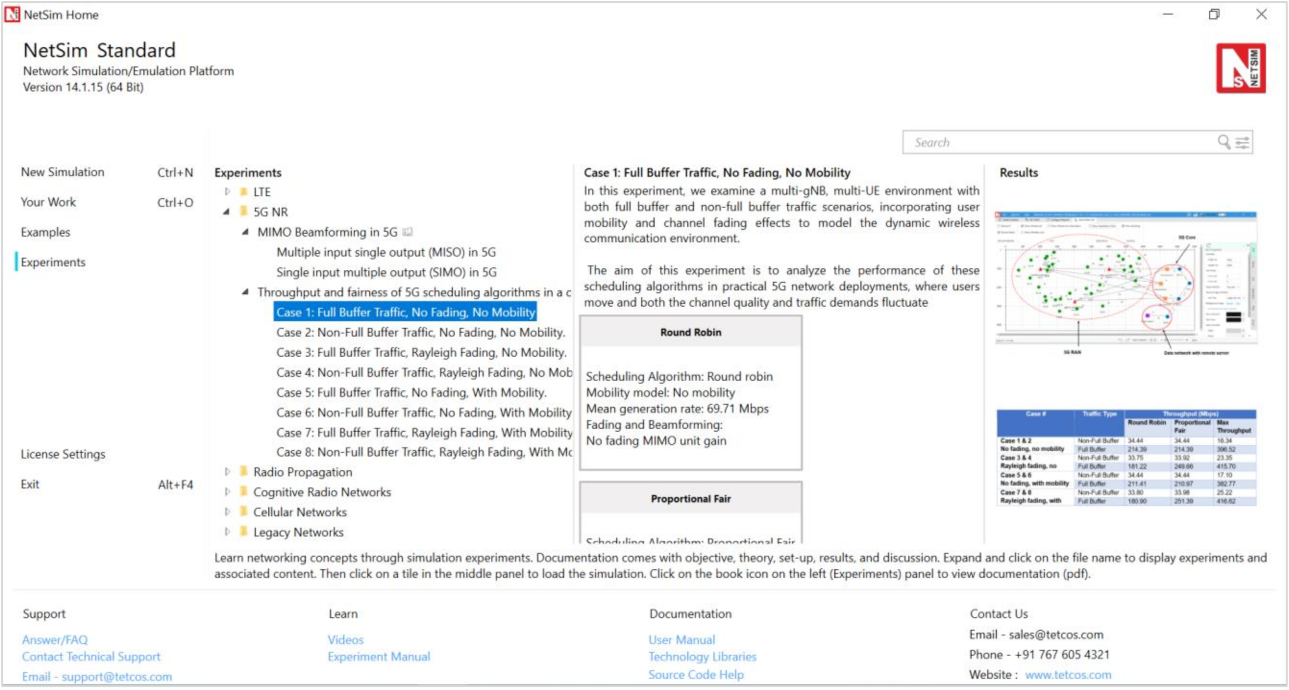


Figure 10-17: List of samples under Throughput and fairness of 5G scheduling algorithms in a complex network environment.

The scenario comprises of

- 3 gNBs placed in a triangular configuration. 10 UEs per gNB.
- Inter gNB distance: 2 km.

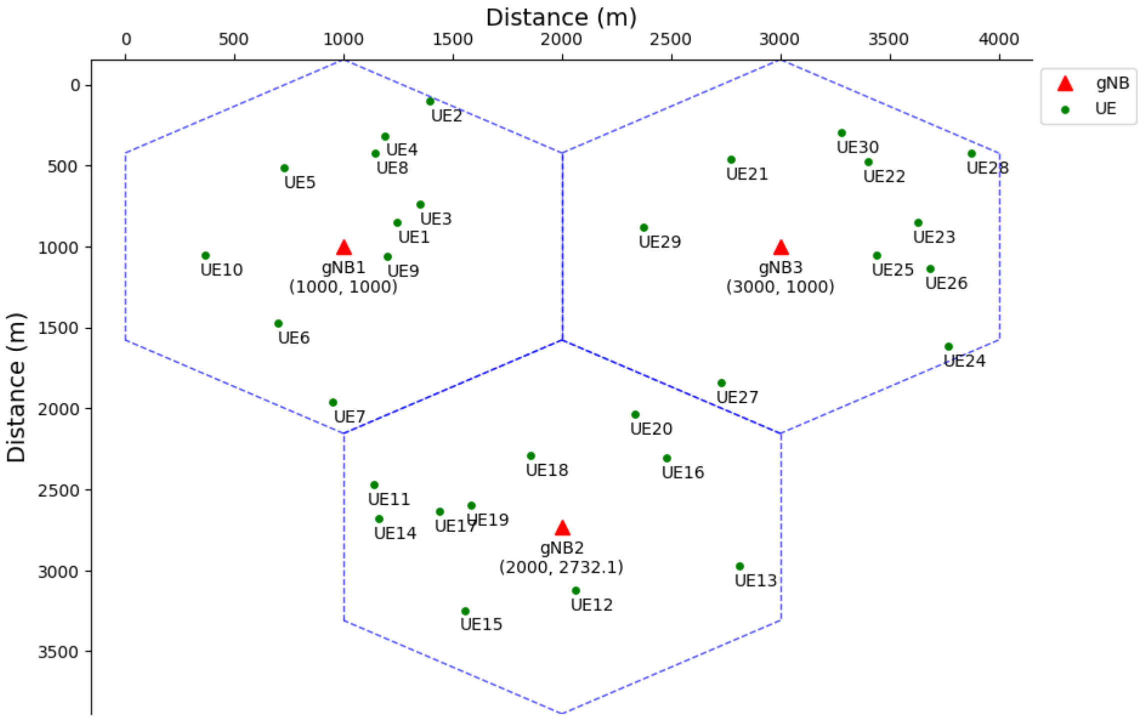


Figure 10-18: Illustration of simulation scenario with 3 gNBs and 30 UEs; 10UEs attached to each gNB. Hexagonal tessellation is shown.

NetSim UI would display the network topology shown in the screenshot below when you open the example configuration file.

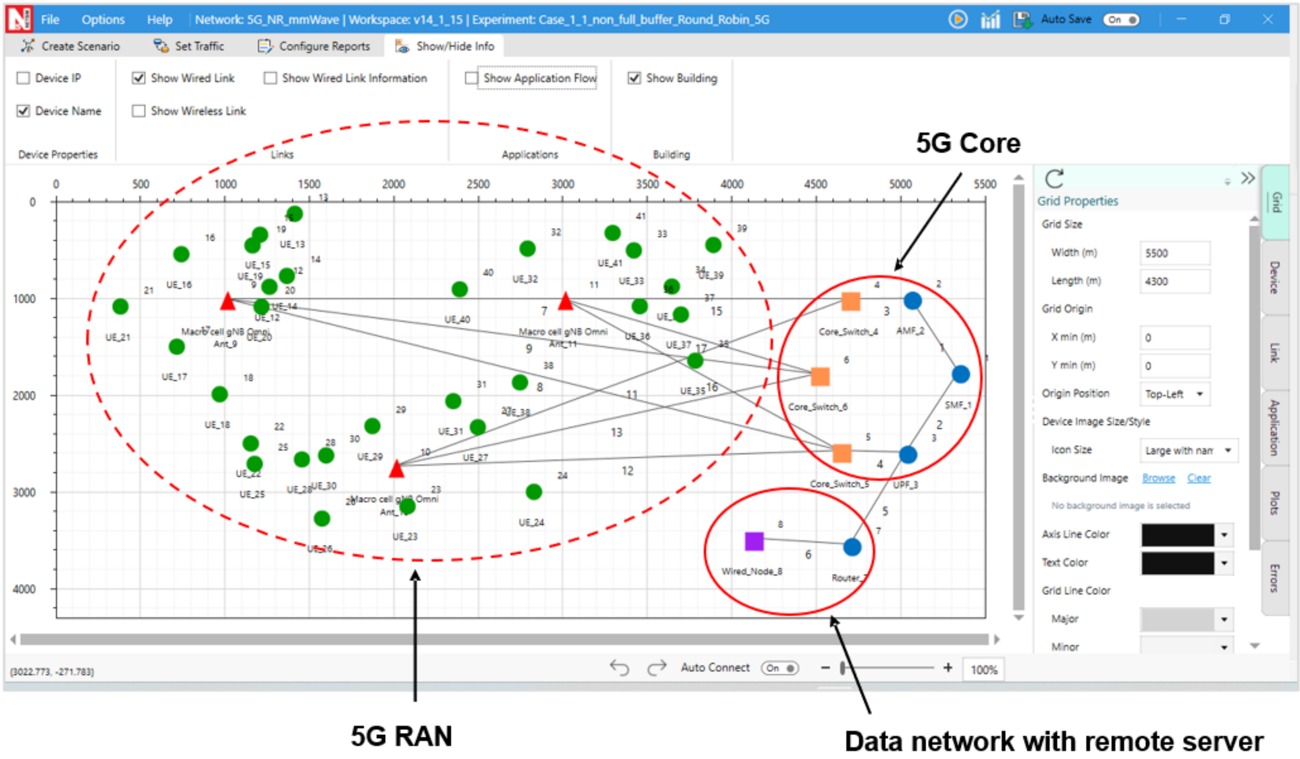


Figure 10-19: The same simulation scenario is replicated in NetSim. The hexagonal tessellation in the RAN is not shown, while the core and data network devices are seen in the NetSim screen shot.

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

Step 1: Set the grid width to 8000m & length to 4000m.

Step 2: Place 3 gNBs on the grid, with a distance of 2km between each gNB, positioned at the vertices of an equilateral triangle. Set the properties of each gNB as described in Table 10-5.

Step 3: Deploy 30 UEs, ensuring that each gNB has 10 associated UEs, as shown in Figure 10-18. The UE coordinates (positions) are obtained using a Python program, which can be saved as a CSV file and imported into our NetSim via the rapid configurator. The details of the Python code are explained in the appendix. Configure the UE settings as shown in Table 10-6.

Step 4: Configure the downlink application for all 30 UEs for both full buffer and non-full buffer cases as shown in Table 10-8.

Step 5: After configuring the settings below, enable the LTENR Radio Measurement log in the Networks Log panel and run the simulation for 10 seconds for different cases as mentioned in Table 10-11.

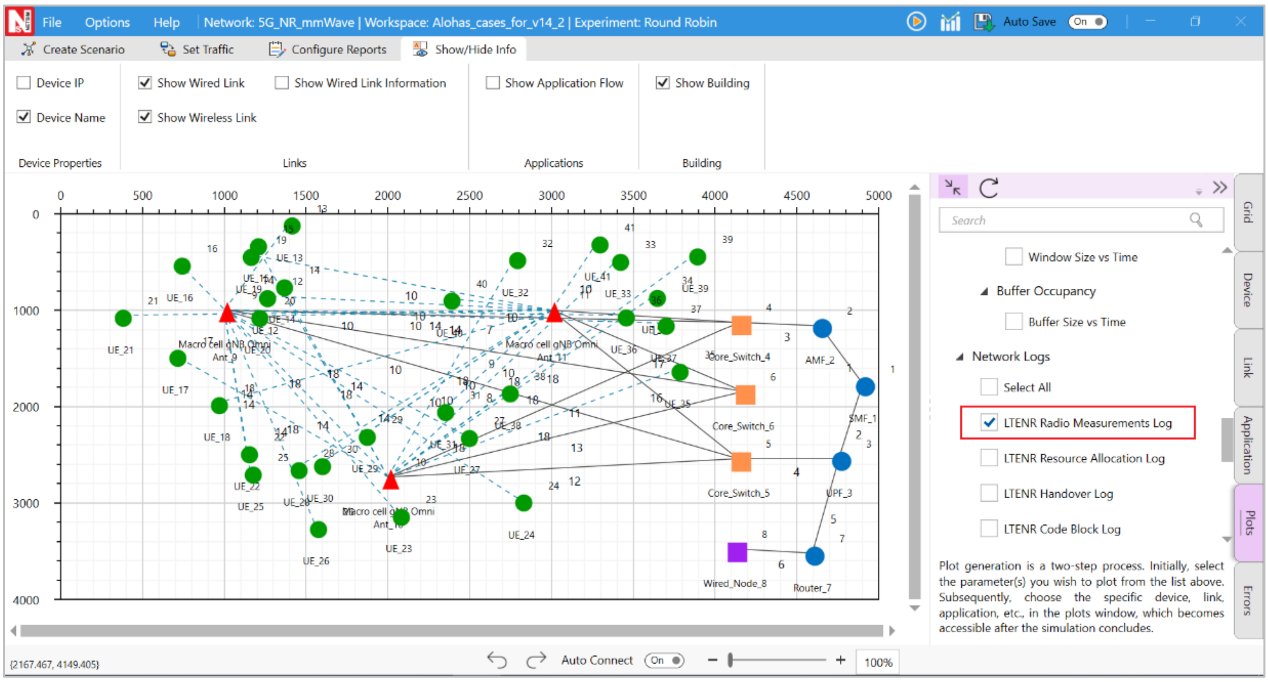


Figure 10-20: Enabling LTENR Radio Measurement log file

10.2.1.1 Device setting

gNB Properties (Interface 4 (5G RAN)->Physical Layer)	
Datalink layer properties	
Scheduling Type	Round Robin / Proportional Fair / Max Throughput
Physical layer properties	
TX Power (dBm)	40
Duplex Mode	TDD
CA Type	Single band
CA Configuration	n78
Component Carrier 1	
DL UL Ratio	01:01
Numerology	2
Channel Bandwidth (MHz)	100
Antenna	
TX x RX Antenna Count	1 x 1
PDSCH and PUSCH Configuration	
MSC Table	QAM256
CSI Report Configuration	
CQI Table	Table2
Channel Model	
Pathloss model	Log distance
Pathloss exponent (μ)	3
Shadowing Model	None
Fast Fading Model	No fading / Rayleigh
Interference Model	

Downlink Interference Model	Exact geometric model
Uplink Interference Model	No interference

Table 10-5: gNB properties configured in Physical Layer of Interface 4 (5G RAN)

UE Properties	
Position Properties	
Mobility Model	No mobility / Random walk
Interface-5G RAN (Physical Layer)	
Antenna	
TX x RX Antenna Count	1 x 1

Table 10-6:UE properties configuration

10.2.1.2 Mobility setting

UE Properties	
Position Properties	
Mobility Model	Random walk
Velocity (m/s)	10
Calculation Interval (s)	1

Table 10-7: Mobility configuration as random walk

10.2.1.3 Application Settings for Non-Full Buffer Traffic (Downlink) and Full Buffer Traffic (Downlink)

Application Properties		
Application settings	Non-full buffer traffic	Full buffer traffic
Source ID	8	8
Destination ID	UE 12 to UE 41	UE 12 to UE 41
Packet Size (B)	1460	1460
Inter Arrival Time (μs)	10000	170
Mean Generation Rate (Mbps)	1.17	68.71

Table 10-8: Non-full buffer traffic downlink application properties, configured from remote server.

10.2.1.4 Table of cases for full and non-full buffer traffic under various fading models

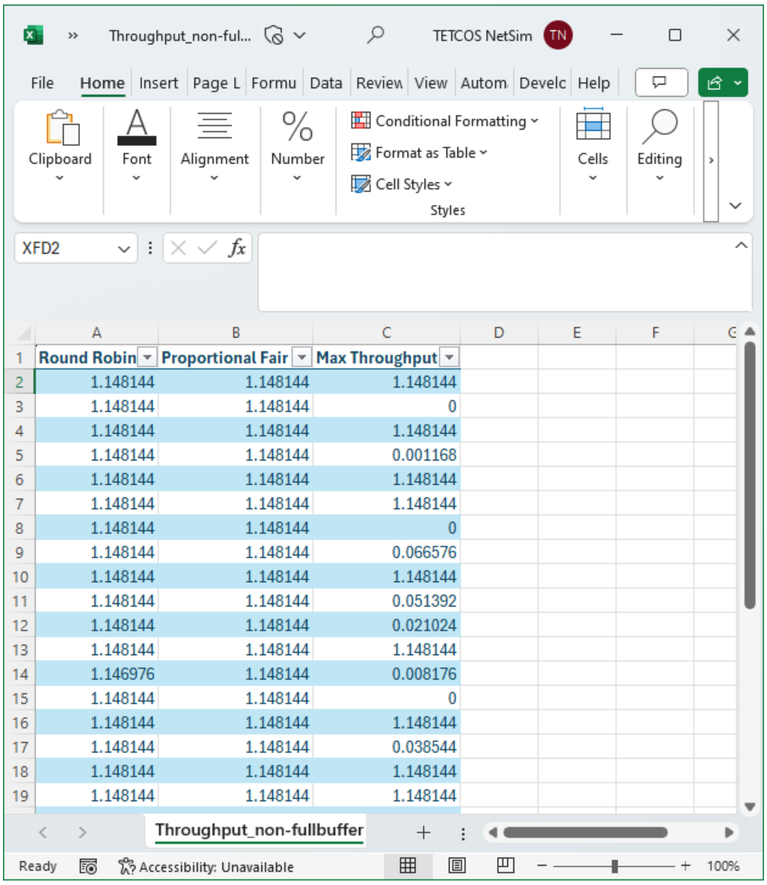
Case #	Traffic	Rayleigh Fading	Mobility
Case 1	Full Buffer	No	No
Case 2	Non-Full Buffer	No	No
Case 3	Full Buffer Traffic	Yes	No
Case 4	Non-Full Buffer	Yes	No
Case 5	Full Buffer	No	Yes
Case 6	Non-Full Buffer	No	Yes
Case 7	Full Buffer Traffic	Yes	Yes
Case 8	Non-Full Buffer Traffic	Yes	Yes

Table 10-9: List of cases

10.2.2 Results and Analysis

The CDF plots shown below can be generated using a Python program. You can access the program via the following link: https://github.com/NetSim-TETCOS/Throughput-and-fairness-of-5G-scheduling-algorithms/archive/refs/heads/main.zip

To generate the throughput plots, copy the throughput of all applications obtained post-simulation in the NetSim simulation result window into an Excel file for different scheduling algorithms, as illustrated in Figure 10-20. Then, run the Python program to generate the plot.



Round Robin	Proportional Fair	Max Throughput
1.148144	1.148144	1.148144
1.148144	1.148144	0
1.148144	1.148144	1.148144
1.148144	1.148144	0.001168
1.148144	1.148144	1.148144
1.148144	1.148144	1.148144
1.148144	1.148144	0
1.148144	1.148144	0.066576
1.148144	1.148144	1.148144
1.148144	1.148144	0.051392
1.148144	1.148144	0.021024
1.148144	1.148144	1.148144
1.146976	1.148144	0.008176
1.148144	1.148144	0
1.148144	1.148144	1.148144
1.148144	1.148144	0.038544
1.148144	1.148144	1.148144
1.148144	1.148144	1.148144

Figure 10-21: Case 1 - Non-full buffer, No fading, No mobility. Throughput of all 30 applications for all scheduling algorithms.

Similarly, the CDF plots of SINR can be generated using a Python program. To generate the SINR plots, follow these steps:

- Open the LTENR Radio Measurement log in the simulation result window, as shown in Figure 10-22.
- Filter the Channel to PDSCH, as Figure 10-23.
- Copy the SINR for each scheduling algorithm into an Excel file, following the format shown in Figure 10-24.

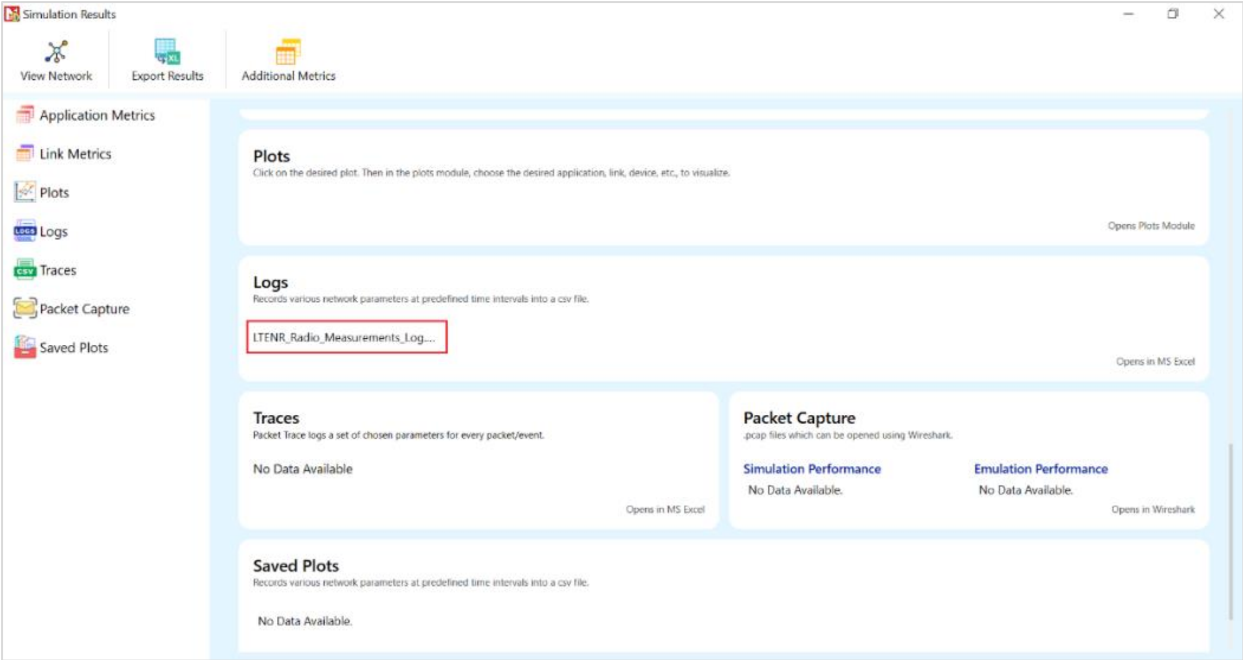


Figure 10-22: Enabling LTENR Radio Measurement log file.

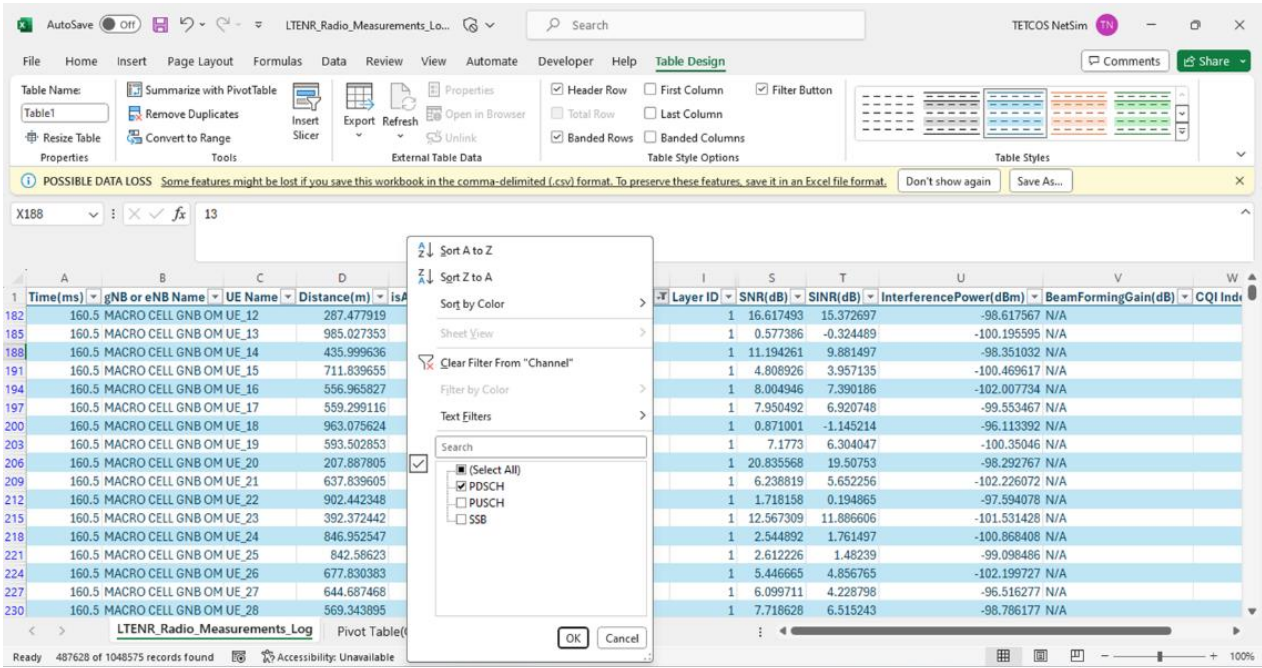
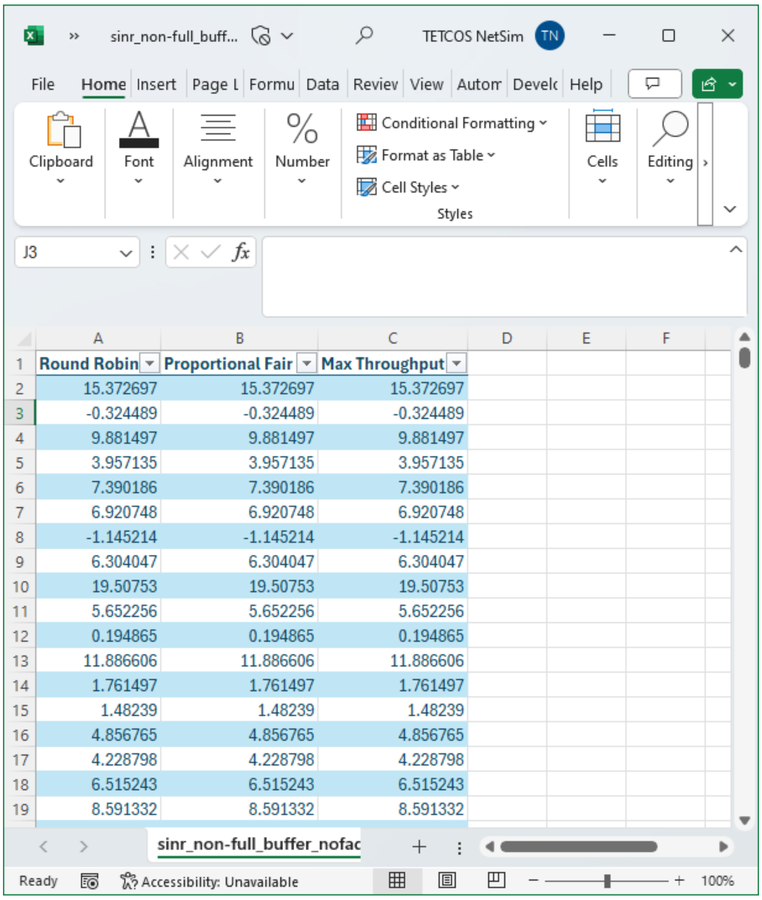


Figure 10-23: In LTENR Radio Measurement log, filter Channel to PDSCH.



	Round Robin	Proportional Fair	Max Throughput
2	15.372697	15.372697	15.372697
3	-0.324489	-0.324489	-0.324489
4	9.881497	9.881497	9.881497
5	3.957135	3.957135	3.957135
6	7.390186	7.390186	7.390186
7	6.920748	6.920748	6.920748
8	-1.145214	-1.145214	-1.145214
9	6.304047	6.304047	6.304047
10	19.50753	19.50753	19.50753
11	5.652256	5.652256	5.652256
12	0.194865	0.194865	0.194865
13	11.886606	11.886606	11.886606
14	1.761497	1.761497	1.761497
15	1.48239	1.48239	1.48239
16	4.856765	4.856765	4.856765
17	4.228798	4.228798	4.228798
18	6.515243	6.515243	6.515243
19	8.591332	8.591332	8.591332

Figure 10-24: SINR for each scheduling algorithms for non-full buffer.

10.2.2.1 CDF of throughput for the various cases

The obtained plots for full and non-full buffer traffic under various fading models are shown below.

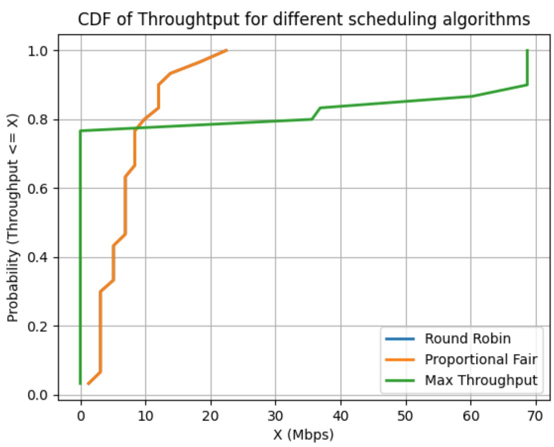


Figure 10-25: Case 1-Full Buffer Traffic. No Fading. No Mobility.

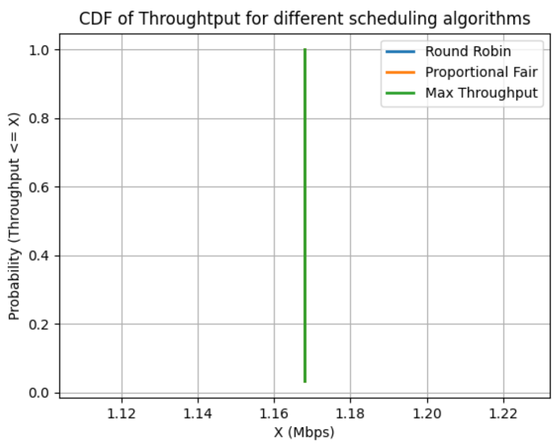


Figure 10-26: Case 2-Non-Full Buffer Traffic. No Fading. No Mobility.

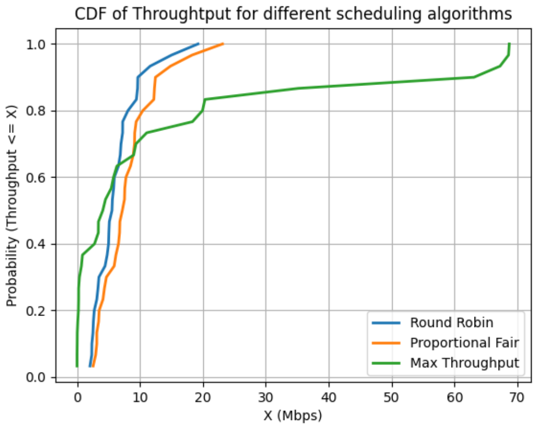


Figure 10-27: Case 3-Full Buffer Traffic, Rayleigh Fading. No Mobility.

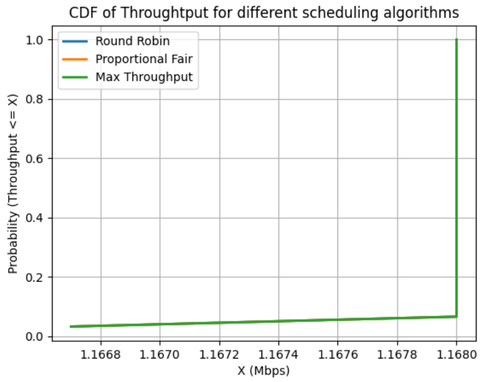


Figure 10-28: Case 4-Non-Full Buffer Traffic, Rayleigh Fading. No Mobility.

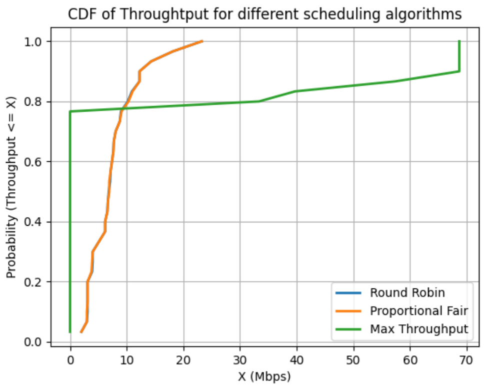


Figure 10-29: Case 5-Full Buffer Traffic. No Fading. With Mobility.

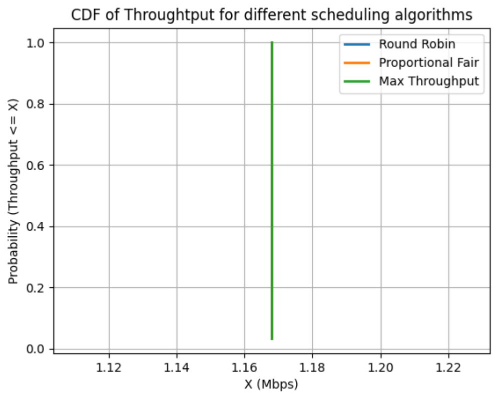


Figure 10-30: Case 6-Non-Full Buffer Traffic. No Fading. With Mobility.

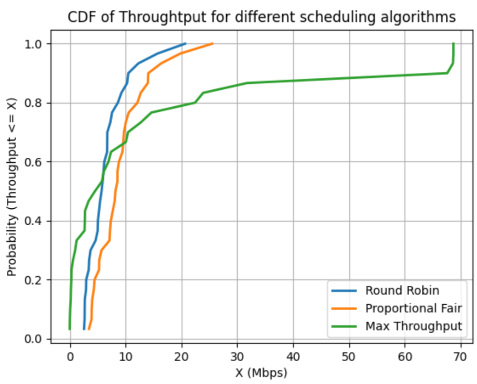


Figure 10-31: Case 7-Full Buffer Traffic. Rayleigh Fading. With Mobility.

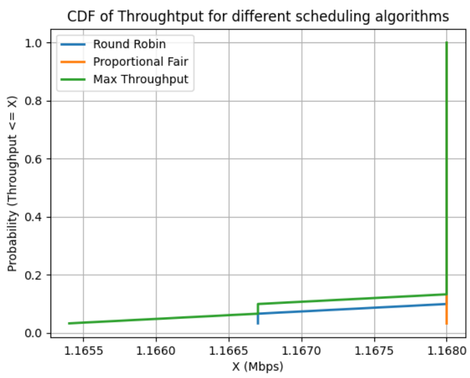


Figure 10-32: Case 8-Non-Full Buffer Traffic. Rayleigh Fading. With Mobility.

Full buffer scenarios generally show higher throughput due to continuous data availability, which allows the Max Throughput algorithm, in particular, to perform optimally by prioritizing users with the best channel conditions. Conversely, the performance in non-full buffer scenarios, where data demand is intermittent, exhibits lower throughput. This is because the Max Throughput algorithm does not perform as well when the user with the best channel condition has no data to send, thus leading to underutilization of resources.

When comparing the no fading/no mobility cases to those involving Rayleigh fading and/or mobility, there is a noticeable drop in throughput and an increase in variability across all scheduling algorithms. This is particularly evident with the Max Throughput algorithm, which shows larger swings in performance due to its sensitivity to channel quality changes.

10.2.2.2 CDF of SINR for the various cases

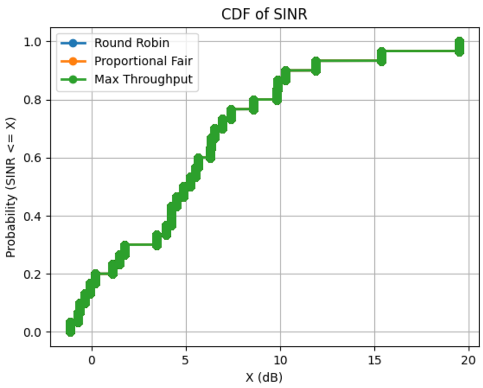


Figure 10-33: Case 1/2-Full Buffer/Non-Full Buffer traffic. No Fading. No Mobility.

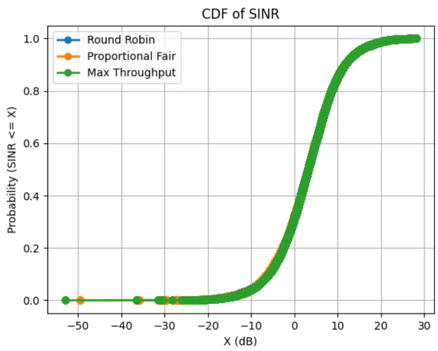


Figure 10-34: Case 3/4-Full Buffer/Non-Full Buffer traffic. Rayleigh Fading. No Mobility

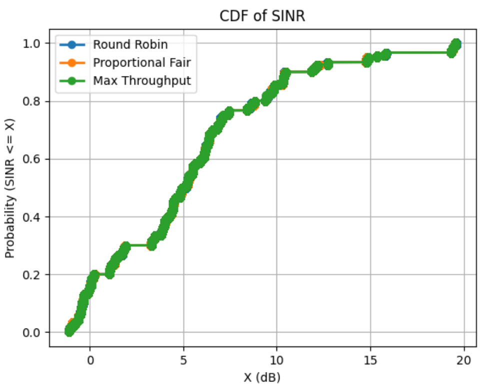


Figure 10-35: Case 5/6-Full Buffer/Non-Full Buffer traffic. No Fading. With Mobility.

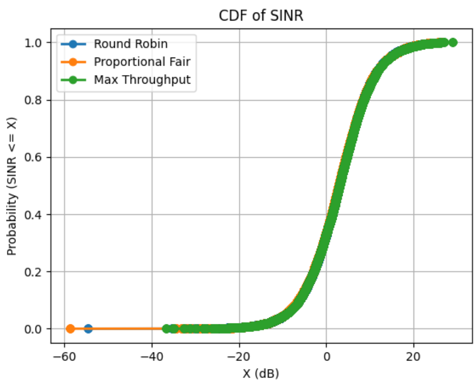


Figure 10-36: Case 7/8-Full Buffer/Non-Full Buffer traffic. Rayleigh Fading. With Mobility.

The plots displayed show a single curve for the CDF of SINR because the SINR values are identical across all three scheduling algorithms—Round Robin, Proportional Fair, and Max Throughput. This results in the curves for the other two algorithms overlapping perfectly with the visible one, rendering them indistinguishable. As expected, SINR, a PHY layer measure is independent of the MAC layer scheduling technique.

Rayleigh fading introduces more variability in the channel conditions. The CDF curve is shifted towards lower SINR values compared to the no fading scenario.

Interestingly, the introduction of mobility, when combined with Rayleigh fading, does not significantly alter the SINR distribution compared to just Rayleigh fading alone. This can occur if the mobile path variation remains within a typical range of fading depths that the Rayleigh model already accounts for.

10.2.2.3 Throughput analysis and Comparison

Case #	Traffic Type	Sum Throughput (Mbps)		
		Round Robin	Proportional Fair	Max Throughput
Case 1 & 2 No fading, no mobility	Non-Full Buffer	35.04	35.04	35.04
	Full Buffer	232.87	232.90	407.48
Case 3 & 4 Rayleigh fading, no mobility	Non-Full Buffer	35.03	35.03	35.03
	Full Buffer	201.68	276.07	447.58
Case 5 & 6 No fading, with mobility	Non-Full Buffer	35.04	35.04	35.04
	Full Buffer	231.09	231.05	231.09
Case 7 & 8 Rayleigh fading, with mobility	Non-Full Buffer	35.03	35.04	35.03
	Full Buffer	198.56	275.74	445.60

Table 10-10: Sum throughput for all the cases

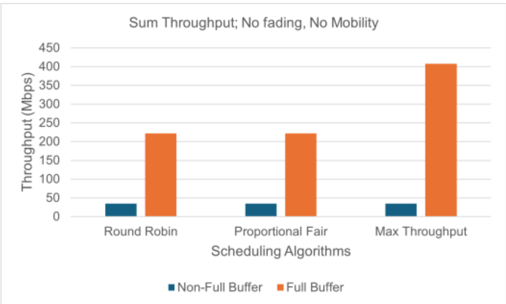


Figure 10-37: Case 1&2 - Sum throughput for No Fading and No mobility.

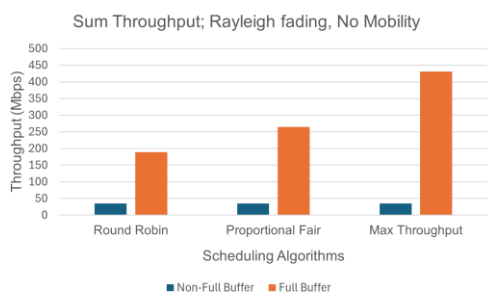


Figure 10-38: Case 3&4 - Sum throughput for Rayleigh Fading and No mobility.

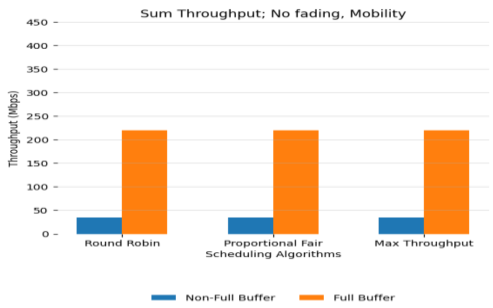


Figure 10-39: Case 5&6 - Sum throughput for No Fading and with mobility.

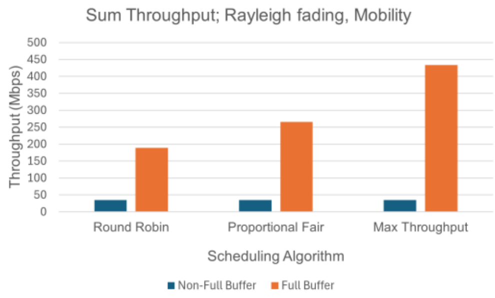


Figure 10-40: Case 7&8 - Sum throughput for Rayleigh Fading and with mobility.

Why does the Max Throughput Scheduler perform better than RR and PFS in full buffer cases?

- MT scheduler prioritizes users with better channel conditions for transmission.
- With full buffer traffic, there is always data to transmit from all users.
- By prioritizing users with better channels, MT can transmit more data than PFS or RR, maximizing throughput

Why is the performance of RR, PFS and Max throughput algorithms similar in the non-full buffer cases?

- In "Non-Full Buffer" scenarios, the data traffic does not fully occupy the available bandwidth because the amount of data being transmitted is lesser than the channel's capacity. With fewer users actively needing resources at any given time, there's less competition for the available bandwidth. This reduces the differences between the scheduling algorithms.
- The non-full buffer condition creates a scenario where the unique characteristics of each algorithm become less influential. The intermittent nature of data transmission requests, combined with varying channel conditions and reduced competition for resources, leads to a convergence in performance across these different scheduling approaches.
- This similarity breaks down under full buffer conditions or with a very large number of users, where the distinct characteristics of each algorithm would become more apparent.

Jain's Fairness Index

Raj Jain's equation

$$\mathcal{J}(x_1, x_2, \dots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{\left(n.\sum_{i=1}^{n} x_i^2\right)}$$

rates the fairness of a set of values where there are n users, x_i is the throughput for the ith connection. The result ranges from $\frac{1}{n}$ (worst case) to 1 (best case), and it is maximum when all users receive the same allocation

Case #	Traffic Type	Round Robin	Proportional Fair	Max Throughput
Case 1 & 2 No fading, no mobility	Non-full buffer	1	1	1
	Full buffer	0.71	0.71	0.22
Case 3 & 4 Rayleigh fading, no mobility	Non-full buffer	1	1	1
	Full buffer	0.73	0.75	0.29
Case 5 & 6 No fading, with mobility	Non-full buffer	1	1	1
	Full buffer	0.71	0.71	0.21
Case 7 & 8 Rayleigh fading, with mobility	Non-full buffer	1	1	1
	Full buffer	0.71	0.73	0.29

Table 10-11: Jain's Fairness Index for all the cases

Case 1, 2, 3 & 4 - No Mobility, With and Without Fading

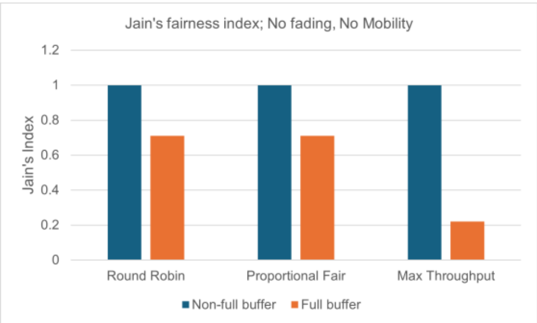


Figure 10-41: Case 1/2 - Jain's fairness index for no mobility, no fading configuration.

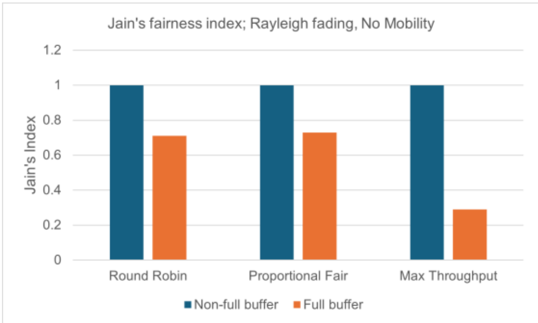


Figure 10-42: Case 3/4 - Jain's fairness index for no mobility, fading configuration.

Case 5, 6, 7 & 8 - With Mobility, With and Without Fading

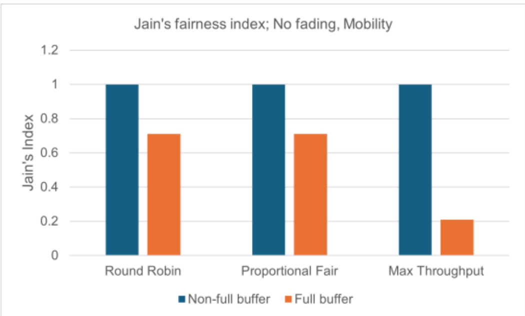


Figure 10-43: Case 5/6 - Jain's fairness index for no mobility, no fading configuration.

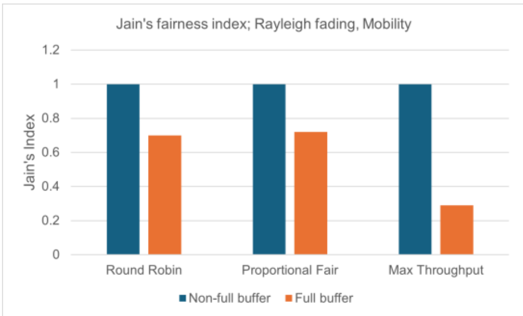


Figure 10-44: Case 7/8 - Jain's fairness index for no mobility, fading configuration.

Discussion

The fairness index is 1 in all cases involving non-full buffer traffic. The reasons for this can be found in the explanation in the previous section. In the cases involving full buffer traffic i.e., when the UEs are in full load conditions, Round Robin and Proportional Fair algorithms show relatively high fairness scores. In contrast, Max Throughput has a significantly lower fairness index, as it tends to favour users with better channel conditions, leading to unequal resource distribution in the no-fading scenario. However, it improves slightly in the fading scenario, likely due to the overall reduction in SINR, which somewhat levels the playing field among users.

Conclusion

The analysis of the fairness and throughput tables and plots, show how scheduling algorithms significantly impact both the fairness of resource distribution and the achievable throughputs in complex 5G deployment scenarios.

Appendix – Python codes

The different Python codes relevant to this experiment are:

- gNB UE positioning and hexgonal tessellation (python tessallation.py)
- CDF plots for throughput (cdf_throughput_plot.py)
- CDF plot for SINR (cdf_sinr_plot.py)

These files are available in the GitHub link : https://github.com/NetSim-TETCOS/Throughput-and-fairness-of-5G-scheduling-algorithms/archive/refs/heads/main.zip

10.3 Understanding the 5G NR PHY

10.3.1 Objective

This experiment has four goals. First, to gain an appreciation for the 5G NR physical layer, i.e., the time-frequency resource grid in the OFDM access scheme. Second, to understand how a packet is transmitted over this OFDM PHY in NetSim, and the assumptions involved. Third, to analytically estimate (per 3GPP standards) the application throughput for a simple use case. And finally, simulate and analyze throughput as different PHY parameters are varied.

10.3.2 Introduction

OFDM: 5G uses Orthogonal Frequency Domain Multiplexing (OFDM) as the multiple access scheme for both downlink and uplink transmissions with the flexibility of multiple subcarriers spacing that supports diverse application scenarios. The smallest physical resource, known as the resource element (RE), comprises one subcarrier and one OFDM symbol.

The time-domain transmission structure comprises of frames 10 ms (to support backward compatibility with LTE). Each frame is composed of 10 subframes of 1 ms each. The 1 ms subframe is then divided into one or more slots in 5G, whereas LTE had exactly two slots in a subframe. The slot size depends on the numerology, μ, and is equal to $\frac{1}{2^{\mu}}$ ms. The number of OFDM symbols per slot is 14 for a configuration using a normal cyclic prefix. For extended cyclic prefixes, the number of OFDM symbols per slot is 12. Data is transmitted over these symbols.

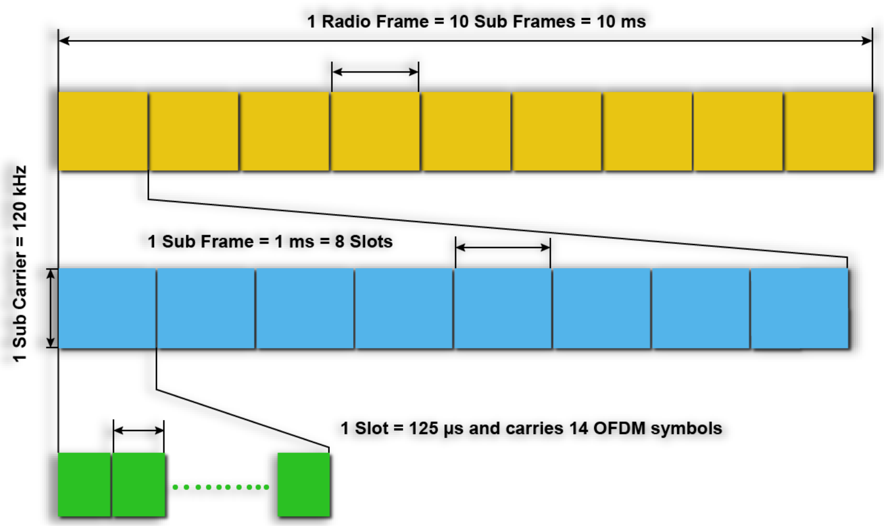


Figure 10-45: NR Frame Structure when numerology μ is set to 3.

In the time division duplex version, each frame is partitioned into downlink subframes and uplink subframes. The downlink part of each frame is used to send data from the gNB to the UEs. The uplink part of the frame is used to send data from the UEs to the destinations, via the gNB. The uplink-downlink ratio is a GUI parameter in NetSim. If Internet access is a major application in a system, then the downlink part of the frame would be substantially larger than the uplink part, due to the asymmetry of Internet access traffic.

In the frequency domain, the group of 12 consecutive sub-carriers forms a resource block (RB). The sub carrier spacing (SCS) is also dependent on numerology, μ and is equal to $2^{\mu} \times 15$ KHz. 5G supports total carrier bandwidth up to 400 MHz with a maximum of 275 RBs.

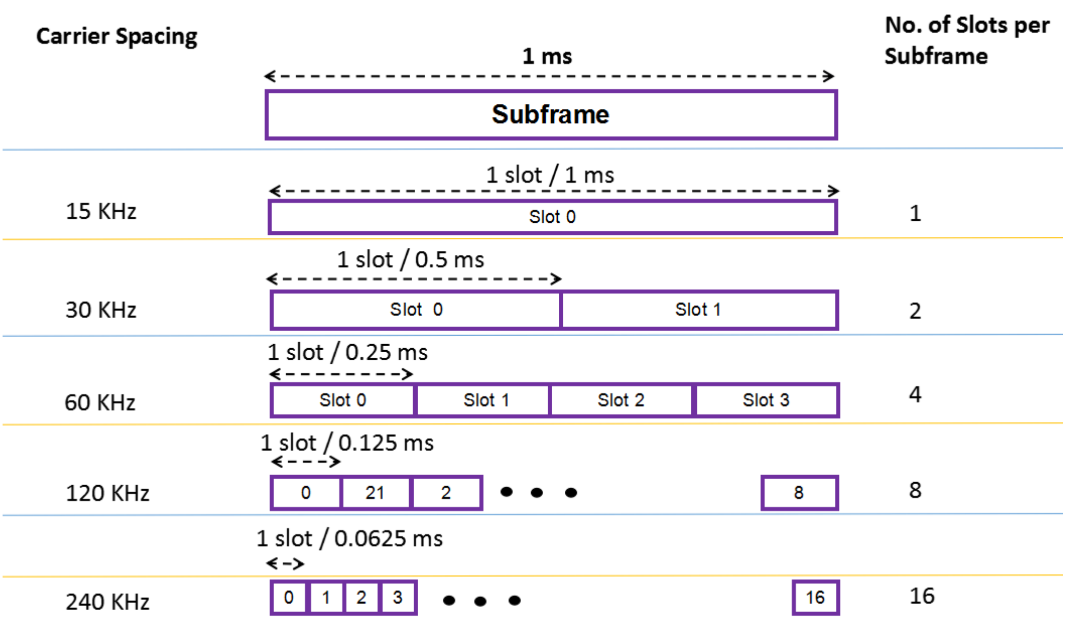


Figure 10-46: The OFDM frame structure. Slot times get shorter as the sub-carrier spacing gets larger.

Data Transmission in NetSim

- In TDD operation the UL and DL transmissions are separated in the time-domain over different frames/subframes/slots/symbols and use the same carrier frequency. In FDD operation UL and DL transmissions are separated in the frequency domain, with different frequencies used for UL and for DL transmissions. NetSim does slot-based scheduling. For example, if the DL:UL ratio is 4:1 then 4 slots are allotted to DL and 1 slot to UL.
- Higher layer packets arrive at the RLC buffer for each UE and each gNB.
- The MAC Scheduler determines the Transport block size (TBS) based on the channel quality index (CQI). The CQI is determined by the Adaptive Modulation and Coding (AMC) function based on the SNR.

- Now, the SNR is determined from a) large-scale path loss and shadowing calculated per the 3GPP's stochastic propagation models, and b) the small-scale fading which leads to beamforming gains when using MIMO. These models provide signal attenuation as an output. Several parameters are used in the model, including the distance between the transmitter and the receiver. These computations are executed each associated UE-gNB pair, in DL and UL, at the start of simulation and again at every mobility event. In calculating SNR, the noise power is obtained from $N = k \times T \times B$.
- Note that the SNR/CQI is not computed/fed-back using reference signals but is computed on the data channel. Then it is assumed to be instantaneously known to the transmitter and receiver. This assumption is known as perfect CSIT and CSIR. With perfect CSIT the transmitter can adapt its transmission rate (MCS) relative to the instantaneous channel state (SNR).
- Based on this SNR the AMC determines a wideband CQI which indicates the highest rate Modulation and coding scheme (MCS), that it can reliably decode, if the entire system bandwidth were allocated to that user. The rate adaptation is discrete (not continuous), and the modulation and coding scheme (MCS) is selected from a standard specified table. The modulation scheme defines the number of bits, that can be carried by a single RE. Modulation schemes supported by 5G include QPSK (2 bits), 16 QAM (4 bits), 64 QAM (6 bits), and 256 QAM (8 bits). The code rate defines the proportion of bits transmitted that are useful. It is computed as the ratio of useful bits by total bits that are transmitted. The modulation order Q_m, which denotes the number of bits per RE, and the code rate denoted by R are jointly encoded as modulation and coding scheme (MCS) index. These values of Q_m and R are then passed to the TBS determination function.
- At each gNB a frame of length 10ms is started. Each frame in turn starts 10 sub frames each of length 1ms. Each sub frame then starts a certain number of slots based on numerology.
- The PHY layer in NetSim then notifies the MAC about the slot start. The MAC sub layer in turn seeks a buffer status report from the RLC layer and invokes the MAC scheduler. It then notifies the RLC of the transmission. The RLC then transmits the transport block to the PHY layer. The downlink and uplink data channels (PDSCH, PUSCH) receive this transport block as its service data unit (SDU), which is then processed and transmitted over the radio interface.

10.3.3 Network simulation setup

Open NetSim and click on Experiments > 5G NR > Understanding the 5G NR PHY then click on the tile in the middle panel to load the example as shown in below.

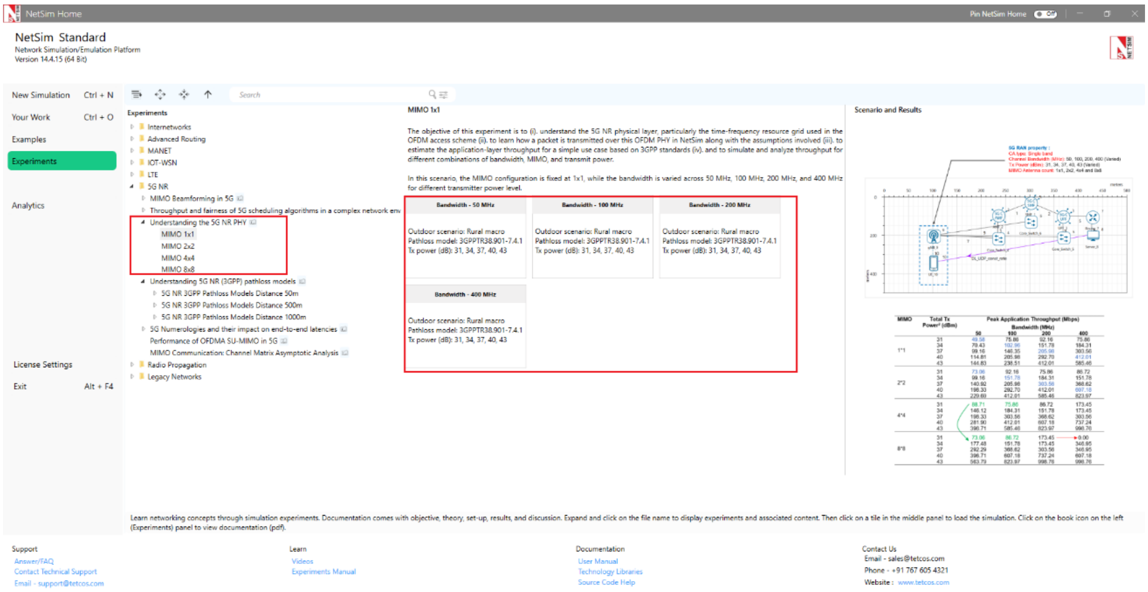


Figure 10-47: NetSim Home Window

NetSim UI would display the following network topology when you open the example configuration file as shown below screenshot.

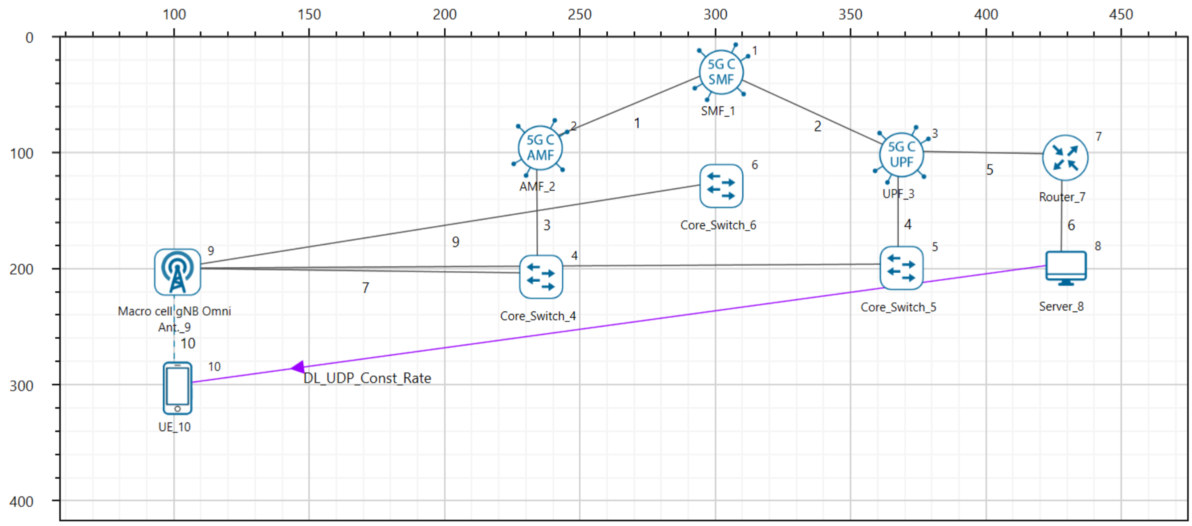


Figure 10-48: Network topology in this experiment

10.3.4 Settings

The following settings were configured in the network setup.

- The UE is placed 100m away from the gNB.
- The following properties were set in Interface 5G RAN, Physical Layer of gNB.

gNB Interface 5G RAN	
gNB Height (m)	10
Tx Power (dBm)	31, 34, 37, 40, 43 (Varied)

Tx Antenna Count	1* (varied from 1 to 8)
Rx Antenna Count	4
CA Type	Single Band
CA Configuration	n261
DL: UL Ratio	4:1
Numerology	3
Channel Bandwidth (MHz)	50, 100, 200, 400 (Varied)
MCS Table	QAM256
CQI Table	TABLE2
Outdoor Scenario	Rural Macro
Indoor Office Type	Mixed Office
Pathloss Model	3GPPTR38.901-7.4.1
LOS Mode	User Defined
LOS Probability	0
Shadow Fading Model	None
Fast Fading Model	No Fading

Table 10-12: gNB properties

- Tx Antenna Count =4 and Rx Antenna Count is varied from 1 to 8 in UE > Interface 5G RAN > Physical Layer
- A downlink CBR application was configured from Wired Node (Server) to UE with Packet Size 1460B and IAT (varied for each layer count) and the start time was set to 1s.
- Run the simulation for 1.2s.

- Run the simulation for different MIMO Layers and different Tx Powers in gNB and note down the throughput obtained.

10.3.5 Analytical Estimation of Data Throughput

We derive the throughput for the setting involving 2×2 MIMO (2-layers) with 100MHz Bandwidth and 31 dBm Transmit power. The procedure for TBS determination given in the steps below is per 3GPP TS 38.214 Section 5.1.3.2 (DL).

1. Initially, the Pathloss (dB) is calculated based on the Pathloss models specified in the 3GPP standards[7]. Pathloss in this example turns out to be $122.26\ dB$
2. The Total Loss is then calculated using the following equation:

$$TotalLoss\ (dB) = Pathloss\ (dB) + ShadowFading\ Loss + O2I\ Loss + Additional\ Loss$$

In this example, $Shadow\ Fading\ Loss = 0,\ O2I\ Loss = 0,\ Additional\ Loss = 0$

$$Total\ Loss = 122.26 + 0 + 0 + 0 = 122.26\ dB$$

[7] We do not get into the details of the pathloss computations here. The specifics are explained in Experiment 4: Understand 3GPP 5GNR pathloss models.

3. The Received power (Per layer) is calculated, using the Tx Power (per layer), the Total Loss and the Beamforming Gain (per layer). Since fading is turned off the Beamforming (BF) gain per layer is 0 dB.

$$Rx\ Power_{Layer}\ (dBm) = Tx\ Power_{Layer}\ (dBm) - Total\ Loss\ (dB) + BFGain_{Layer}$$

$$Rx\ Power_{Layer\ 1} = 27.98 - 122.26 + 0 = -94.28\ dBm$$

$$Rx\ Power_{Layer\ 2} = 27.98 - 122.26 + 0 = -94.28\ dBm$$

4. Thermal Noise computation

$$Thermal\ Noise = k \times T \times B$$

$k\ (Boltzmann's\ constant) = 1.38 * 10^{-23}, T\ (Temperature) = 300\ K, B = 100\ MHz$

$Thermal\ Noise = 4.14 \times 10^{-13} W$

$$= -93.829\ dBm$$

5. From the Rx Power and Thermal Noise, SNR is calculated
   a) $Rx\ Power\ in\ dBm\ is\ converted\ into\ mW$

   $$Rx\ Power, P = -94.28\ dBm = 3.73 \times 10^{-10} mW$$

   b) $Thermal\ Noise\ in\ dBm\ is\ converted\ into\ mW$

   $$Thermal\ Noise, N = -93.82\ dBm = 4.14 * 10^{-10}\ mW$$

   c) $SNR(Linear) = \frac{E_b}{N_0} = \frac{Rx\ Power}{Thermal\ Noise} = \frac{P}{N} = \frac{3.73\times10^{-10}}{4.14\times10^{-10}} = 0.902$

6. From SNR, the Spectral Efficiency is calculated as follows:

$$Spectral\ Efficiency_{Layer} = \log_2\left(1 + \left(\frac{E_b}{N_0}\right)\right)$$

$$= \log_2\big(1 + (0.902)\big) = 0.927$$

7. The CQI Index is then looked up from the respective CQI Table using the spectral efficiency obtained. The table is given below:

{0,	Modulation_Zero,	0,	0},	//out of range
{1,	Modulation_QPSK,	78,	0.1523},	
{2,	Modulation_QPSK,	193,	0.3770},	
{3,	Modulation_QPSK,	449,	0.8770},	
{4,	Modulation_16_QAM,	378,	1.4766},	
{5,	Modulation_16_QAM,	490,	1.9141},	
{6,	Modulation_16_QAM,	616,	2.4063},	
{7,	Modulation_64_QAM,	466,	2.7305},	
{8,	Modulation_64_QAM,	567,	3.3223},	
{9,	Modulation_64_QAM,	666,	3.9023},	
{10,	Modulation_64_QAM,	772,	4.5234},	
{11,	Modulation_64_QAM,	873,	5.1152},	
{12,	Modulation_256_QAM,	711,	5.5547},	
{13,	Modulation_256_QAM,	797,	6.2266},	
{14,	Modulation_256_QAM,	885,	6.9141},	
{15,	Modulation_256_QAM,	948,	7.4063},	

Since the Spectral Efficiency is 0.927, from the CQI Table, $CQI\ Index = 3$ is chosen.

8. Similarly, the MCS Index is taken from the respective MCS Table with respect to the spectral efficiency from the CQI Table:

```
{0,   2,  Modulation_QPSK,      120,   0.2344},
{1,   2,  Modulation_QPSK,      193,   0.3770},
{2,   2,  Modulation_QPSK,      308,   0.6016},
{3,   2,  Modulation_QPSK,      449,   0.8770},
{4,   2,  Modulation_QPSK,      602,   1.1758},
{5,   4,  Modulation_16_QAM,    378,   1.4766},
{6,   4,  Modulation_16_QAM,    434,   1.6953},
{7,   4,  Modulation_16_QAM,    490,   1.9141},
{8,   4,  Modulation_16_QAM,    553,   2.1602},
{9,   4,  Modulation_16_QAM,    616,   2.4063},
{10,  4,  Modulation_16_QAM,    658,   2.5703},
{11,  6,  Modulation_64_QAM,    466,   2.7305},
{12,  6,  Modulation_64_QAM,    517,   3.0293},
{13,  6,  Modulation_64_QAM,    567,   3.3223},
{14,  6,  Modulation_64_QAM,    616,   3.6094},
{15,  6,  Modulation_64_QAM,    666,   3.9023},
{16,  6,  Modulation_64_QAM,    719,   4.2129},
{17,  6,  Modulation_64_QAM,    772,   4.5234},
{18,  6,  Modulation_64_QAM,    822,   4.8164},
{19,  6,  Modulation_64_QAM,    873,   5.1152},
{20,  8,  Modulation_256_QAM,   682.5, 5.3320},
{21,  8,  Modulation_256_QAM,   711,   5.5547},
{22,  8,  Modulation_256_QAM,   754,   5.8906},
{23,  8,  Modulation_256_QAM,   797,   6.2266},
{24,  8,  Modulation_256_QAM,   841,   6.5703},
{25,  8,  Modulation_256_QAM,   885,   6.9141},
{26,  8,  Modulation_256_QAM,   916.5, 7.1602},
{27,  8,  Modulation_256_QAM,   948,   7.4063},
```

Since the Spectral Efficiency is 0.927, MCS Index 3 which corresponds to this spectral efficiency is chosen and the $Modulation\ Order = 2$

9. The TBS size is then determined using the Modulation Order and code rate.
10. Determination of number of Resource Elements within the slot.

$$N'_{RE} = N_{SC}^{RB} \times N_{Symb}^{Sh} - N_{DMRS}^{PRB} - N_{OH}^{PRB}$$

$$N_{SC}^{RB} = 12. Number\ of\ subcarriers\ in\ Physical\ Resource\ Block$$

$$N_{Symb}^{Sh} = 14.\ Number\ of\ Symbols\ per\ Slot$$

$$N_{DMRS}^{PRB} = 0 \rightarrow Number\ of\ Resource\ Elements\ for\ DM-RS\ per\ PRB$$

$$N_{OH}^{PRB} = 0. PDSCH\ overhead$$

$$N'_{RE} = 12 \times 14 - 0 - 0 = 168$$

11. Total number of Resource Elements allocated for PDSCH

$$N_{RE} = \min(156, N'_{RE}) \times n_{PRB}$$

$$n_{PRB} = \ 1.\ Number\ of\ allocated\ PRBs\ for\ the\ UE$$

$$N_{RE} = \min(156, 168) \times 1 = 156 \times 1 = 156$$

12. Intermediate number of information bits is calculated.

$$N_{info} = N_{RE} \times R \times Q_m$$

$R = \frac{MCS_{CodeRate}}{1024} = \frac{449}{1024} = 0.438$ and $Q_m = 2$ (Modulation order)

$$N_{info} = 156 \times 0.438 \times 2 = 136.65$$

13. Since $N_{info} \leq 3824$, the TBS Size is calculated

$$N'_{info} = \max\left(24, 2^n \times floor\left(\frac{N_{info}}{2^n}\right)\right)$$

Where $n = \max\,(3,\ floor(\log_2(N_{info})) - 6)$

$$= max\ (3, floor(log_2(136.65)) - 6) = max\ \ (3, 1) = 3$$

$$N'_{info} = max\left(24, 2^3 \times floor\left(\frac{136.65}{2^3}\right)\right) = max\ \ (24, 136) = 136$$

Hence, the TBS size will be 136, i.e., index 15.

14. The bits per PRB per layer is determined based on the TBS size.
15. For Layer 1,

$$bitsperPRB = bits\ per\ PRB\ (initial) + TBS\ Size$$
$$= 0 + 136 = 136$$

For Layer 2,

$$bitsperPRB_{L2} = bitsperPRB_{L1} + TBS\ Size$$
$$= 136 + 136 = 272$$

16. The Total PRB available is dependent on bandwidth and μ and is shown in the GUI. In this example $PRB\ available = PRB\ Count = 66$
17. The total PRB available is calculated.

$$Total\ PRB\ available = PRBCount - ceil(PRB\ count \times OH_{Downlink})$$

$PRB\ Count = 66$, $OH_{Downlink} = 0.18$ (Per standard)

$$Total\ PRB\ available = 66 - ceil(66 \times 0.18)$$
$$= 66 - 11 = 54$$

18. Number of PRBs allocated is 54.
19. The slot allocation will then take place and the bits per slot is assigned.

$$bits\ per\ Slot = bits\ per\ PRB \times allocated\ PRB$$
$$= 272 \times 54 = 14688\ bits = 1836\ Bytes$$

$$i.e., a\ slot\ can\ transmit\ a\ maximum\ of\ 1836\ Bytes$$

20. Throughput estimation. $DL\ UL\ Ratio = 4{:}1$, implies a DL fraction of 0.8. Since $\mu = 3$, slot time $\frac{1}{2^3}$ ms.

$$DL\ MAC\ Throughput = \frac{1836 \times 8 \times 0.8}{\left(\frac{1}{2^3}\right) \times 10^{-3}} = 94\ Mbps$$

$$DL\ Application\ Throughput = DL\ MAC\ Throughput\ \times \left(\frac{ApplicationPacketSize}{MACPacketSize}\right)$$

$$= 94 \times \left(\frac{1460}{1488}\right) = 92.23\ Mbps$$

(Matches with NetSim's Result of 92.21 Mbps, which can be seen in entry pertaining to MIMO 2*2, Bandwidth 100MHz and Tx Power 31 dBm. The entry is marked in green.)

10.3.6 Results

The following throughputs were obtained after running simulations with different Antenna counts (MIMO layers), Bandwidths and Transmit Power values.

MIMO	Total Tx Power[8] (dBm)	Peak Application Throughput (Mbps) Bandwidth (MHz)			
		50	100	200	400
1*1	31	54.78	75.86	92.16	75.86
	34	78.26	113.82	151.78	184.31
	37	99.16	162.59	227.64	303.56
	40	125.27	205.98	325.23	455.34
	43	144.83	260.17	412.01	585.46
2*2	31	73.06	92.16	75.86	86.72
	34	109.62	151.78	184.31	151.78
	37	156.57	227.64	303.56	368.62
	40	198.33	325.23	455.34	607.18
	43	250.54	412.01	585.46	910.40
4*4	31	88.70	75.86	86.72	173.44
	34	146.12	184.31	151.78	173.45
	37	219.23	303.56	368.62	303.56
	40	313.20	455.34	607.18	737.24
	43	396.71	585.46	910.40	998.76
8*8	31	73.05	86.72	173.44	0
	34	177.48	151.78	173.45	346.95
	37	292.29	368.62	303.56	346.95
	40	438.47	607.18	737.24	607.18
	43	563.79	910.40	998.76	998.76

Table 10-13: Saturation throughput obtained for n261 band (gNB-UE distance of 100m, Rural macro pathloss) for various Bandwidth-MIMO-TxPower combinations. The blue entries show the doubling of throughput when power and BW is doubled. Red shows examples where throughput decreases with increase in bandwidth for fixed power and MIMO layers. Green entries are where throughput decreases with increase in MIMO layers, for fixed BW and power.

[8] This is the transmit power summed over entire BW and summed over all MIMO streams.

10.3.7 Discussion

In Table 10-13 we observe entries marked in:

- Blue: When both the bandwidth and the power are doubled, with MIMO count kept constant, the peak throughput doubles. This is along expected lines.
- Red: In the high bandwidth and low power regime, when the bandwidth is doubled with the transmit power and MIMO count held constant, the peak throughput does not increase but rather decreases.
- Green: At low power when the MIMO layers are increased with fixed transmit power and bandwidth, the peak throughput surprisingly decreases.

Let us understand the red entries, i.e., throughput of 1*1 MIMO, 31 dBm Tx power for bandwidths of 200 and 400 MHz. We can simplify the PHY rate as equal to $k \times L \times Q \times B \times R$ where k is some constant, L is the number of layers (set to 2 here), Q is the modulation order (2 in this case), R is the code rate and B is the bandwidth. From Table 10-13 we see that when the bandwidth increases the spectral efficiency decreases due to an increase in thermal noise at higher bandwidths. The received power is constant since the transmit power is fixed. Since the drop in the MCS[9] (due to the reduced spectral efficiency) is larger than the bandwidth increase - 0.438×200 vs. 0.188×400 - the net effect is a decline in the throughput.

BW (MHz)	Rx Power (dB)	Noise (KTB)	SNR	Spectral Efficiency	Spec Eff Table cut off	MCS Index	MCS Code Rate	R (Code rate/1024)	Throughput (Mbps)
200	-91.26	-90.81	-0.44	0.927	0.8770		449	0.438	92.21
400	-91.26	-87.80	-3.45	0.537	0.3770	1	193	0.188	75.92

Table 10-14: Rx Power, Noise, SNR, Spectral Efficiency obtained for different bandwidths with Tx power at 31 dBm.

Next, we turn to the green entries. In Table 10-13 notice that when the MIMO layer count is increased from 4 to 8, the received power (per layer) decreases. This happens because the transmit power is equally divided among all the layers. As SNR reduces, the spectral efficiency per layer decreases. Since the MCS drop (due to lower spectral efficiency) is larger than the multiplexing gain got from multiple MIMO streams - 4×0.438 vs 8×0.188 - the consequence is a decrease in throughput.

[9] Refer Steps 5 through 8 in the section Analytical estimation of Data throughput, to understand how spectral efficiency is got from SNR, and then how MCS is set from spectral efficiency.

MIMO Layers	BW (MHz)	Rx Power (dB)	Noise (KTB)	SNR	Spec Efficiency	Spec Eff Table cut off	MCS Index	MCS Code Rate	R (Code rate/ 1024)	Throughput (Mbps)
4	50	-97.28	-96.83	-0.44	0.927	0.8770	3	449	0.438	88.76
8	50	-100.29	-96.83	-3.45	0.537	0.3770	1	193	0.188	73.11

Table 10-15: Rx Power, Noise, SNR, Spectral Efficiency obtained for each MIMO layer with Tx power set to 31 dBm.

10.3.8 Exercises

1. Estimate the data throughput analytically for different values of Transmit power, Bandwidth and MIMO layer count (Each student can be given a personalized experiment)

10.4 Understanding 5G NR (3GPP) pathloss models

10.4.1 Objective

In the elementary case of 1gNB communicating with 1UE over a 5G NR network, in a rural setting, we study the question: How does the UE-gNB pathloss vary with the distance between the UE and the gNB and the gNB height? What is the optimal height of a gNB?

10.4.2 Motivation

We start with a non-technical explanation of the objective. A mobile phone (in the hands of an individual) is the UE; the cell tower is the guns. Assume the person is in a rural area and is outdoors. Pathloss would determine the signal strength displayed on the phone; a higher loss means a lower signal strength. Mobile network operators (think of the top service providers in our country) invest large sums in setting-up the towers. They wish to know the tower height[10] that gives users the highest signal strength. The answer is not obvious: the more the height of the gNB, the more likely it is that there exists a line-of-sight path to a given UE, but the signal has to traverse a longer distance, incurring a higher path loss. The cell radius might also play a role here: perhaps a lower height is better for smaller sized cells, and a greater height is better for large cells. In this experiment, we will understand these trade-offs.

10.4.3 The 5G pathloss equations

To answer these questions, we look at the 5G pathloss equations for a rural scenario as defined in the 3GPP 38.901 standards:

Scenario	LOS/ NLOS State	Pathloss (dB) (f_c in GHz and d in meters)	Shadow Fading (σ)	Parameter values and ranges
	LOS	$PL_{RMA_{LOS}} = \begin{cases} PL_1, 10m \le d_{2D} \le d_{BP} \\ PL_2, d_{BP} \le d_{2D} \le 10Km \end{cases}$ $\mathrm{PL_1} = 20\log_{10}(40\pi d_{3D} f_c/3) + \min(0.03h^{1.72}, 10)\log_{10}(d_{3D}) - \min(0.044h^{1.72}, 14.77) + 0.002\log_{10}(h)d_{3D}$ $PL_2 = PL_1(d_{BP}) + 40\log_{10}(\frac{d_{3D}}{d_{BP}})$	$\sigma_{SF} = 4$	$h_{BS} = 35m$ $h_{UT} = 1.5m$ $W = 20m$

[10] The antenna can be placed at different heights on the cell tower. Hence the term “Antenna height” would be technically precise.

Rural Macro			$\sigma_{SF} = 6$	$h = 5m$ $5m \leq h \leq 50m$ $5m \leq W \leq 50m$ $10m \leq h_{BS} \leq 150m$ $1m \leq h_{UT} \leq 10m$
	NLOS	$PL_{RMa_{NLOS}} = \max(PL_{RMa_{LOS}}, \quad PL_{RMa'_{NLOS}})$ For $10m \leq d_{2D} \leq 5Km$ $PL_{RMa'_{NLOS}} = 161.04 - 7.1 * \log_{10}(W) + 7.5 * \log_{10}(h) - \left(24.37 - 3.7 * \left(\frac{h}{h_{BS}}\right)^2\right) * \log_{10}(h_{BS}) + (43.42 - (3.1 * \log_{10}(h_{BS})) * (\log_{10}(d_{3D}) - 3) + 20 * (\log_{10}(f_c)) - (3.2 * (\log_{10}(11.75 * h_{UT}))^2 - 4.97)$	$\sigma_{SF} = 8$	
NOTE: 1. Break point distance[11] $\boldsymbol{d_{BP} = 2\pi h_{BS} h_{UT} f_c / c}$, where fc is the centre frequency in Hz, $\boldsymbol{c = 3.0 * 10^8\ m/s}$ is the propagation velocity in free space, and hBS and hUT are the antenna heights at the BS and the UT, respectively. **2.** fc denotes the centre frequency normalized by 1 GHz, all distance related values are normalized by 1 m, unless stated otherwise.				

Table 10-16: Pathloss equations for Rural Macro environment for LOS and NLOS states.

[11] A question for the reader: Why is it called break point?

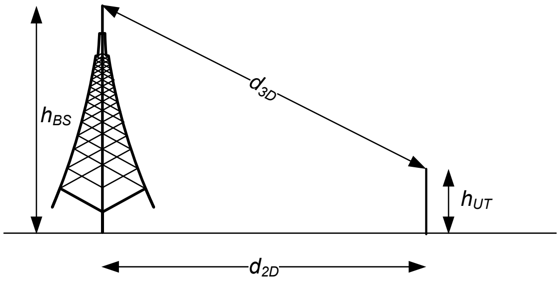


Figure 10-49: Definition of d2D and d3D

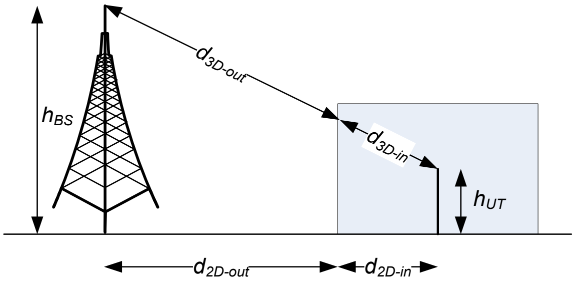


Figure 10-50: Definition of $d_{2D\text{-}out}$, $d_{2D\text{-}in}$ and $d_{3D\text{-}out}$, $d_{3D\text{-}in}$ for indoor UEs

Note that,

$$d_{3D_{out}} + d_{3D_{in}} = \sqrt{\left(d_{2D_{out}} + d_{2D_{in}}\right)^2 + (h_{BS} - h_{UT})^2}$$

Observing the above equations, we see that the pathloss is not a simple expression in terms of gNB height. The other parameters affecting the pathloss are a) the UE-gNB 2D distance and b) the UE state[12].

Consequently, we investigate the revised question: how does the UE-gNB pathloss vary for *combinations* of gNB height, UE-gNB 2D distance, and UE states (LOS/NLOS)?

10.4.4 Network simulation setup

Open NetSim and click on Experiments > 5G NR > Understanding 5G NR (3GPP) pathloss models then click on the tile in the middle panel to load the example as shown in below.

[12] Can the UE directly see the gNB? If yes, it is in a Line-of-sight (LOS) state and if not, it is in the NLOS state.

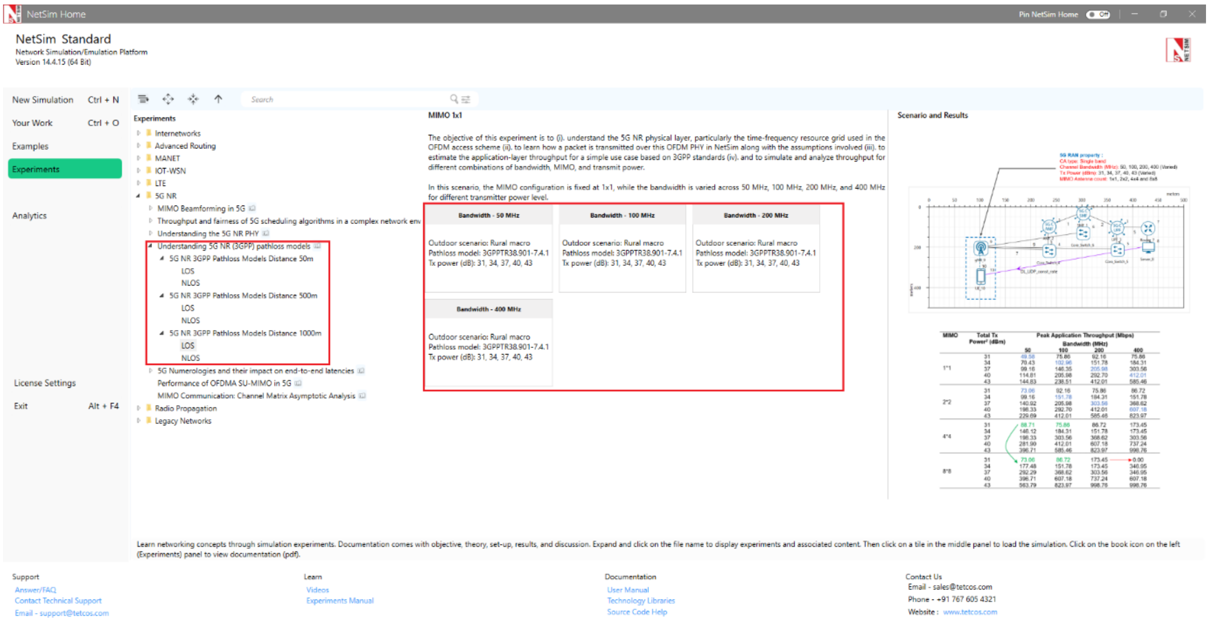


Figure 10-51: List of Experiments

10.4.5 Network scenario:

NetSim UI would display the following network topology when you open the example configuration file as shown below screenshot.

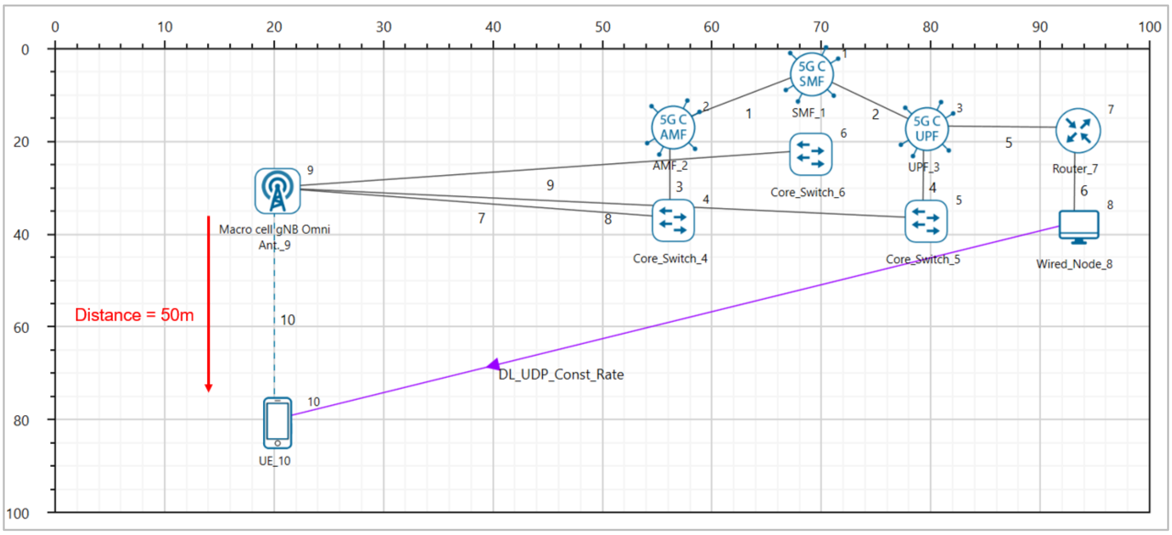


Figure 10-52: Network topology in this experiment

10.4.6 Settings

The following settings were configured in the network setup.

- The UE is placed 50m away from the gNB.
- The following properties were set in Interface 5G RAN, Physical Layer of gNB.

gNB Interface 5G RAN	
gNB Height (m)	Varied from 10 to 150

Tx Power (dBm)	40
Tx Antenna Count	2
Rx Antenna Count	2
CA Type	Single Band
CA Configuration	n78
DL: UL Ratio	4:1
Numerology	0
Channel Bandwidth (MHz)	10
MCS Table	QAM64
CQI Table	TABLE1
Outdoor Scenario	Rural Macro
Indoor Office Type	Mixed Office
Pathloss Model	3GPPTR38.901-7.4.1
LOS Mode	User Defined
LOS Probability	0 or 1 (Varied)
Shadow Fading Model	None
Fast Fading Model	No Fading

Table 10-17: gNB Interface RAN properties

- Tx Antenna Count = Rx Antenna Count = 2 in UE > Interface 5G RAN > Physical Layer
- A downlink CBR application was configured from Wired Node to UE with Packet Size 1460B and IAT 1168µs and the start time was set to 1s.
- Run simulation for 2s.
- In case 2, set the LOS probability to 0 and run simulation for various gNB heights.
- In case 3, place the UE 1000m away from the gNB and repeat the above procedure.
- In case 4, set the LOS probability to 0 and run simulation for 2s.
- Click on the Log/Plots tab in the right panel toolbar to enable LTENR Radio measurement as shown in Figure 10-53.

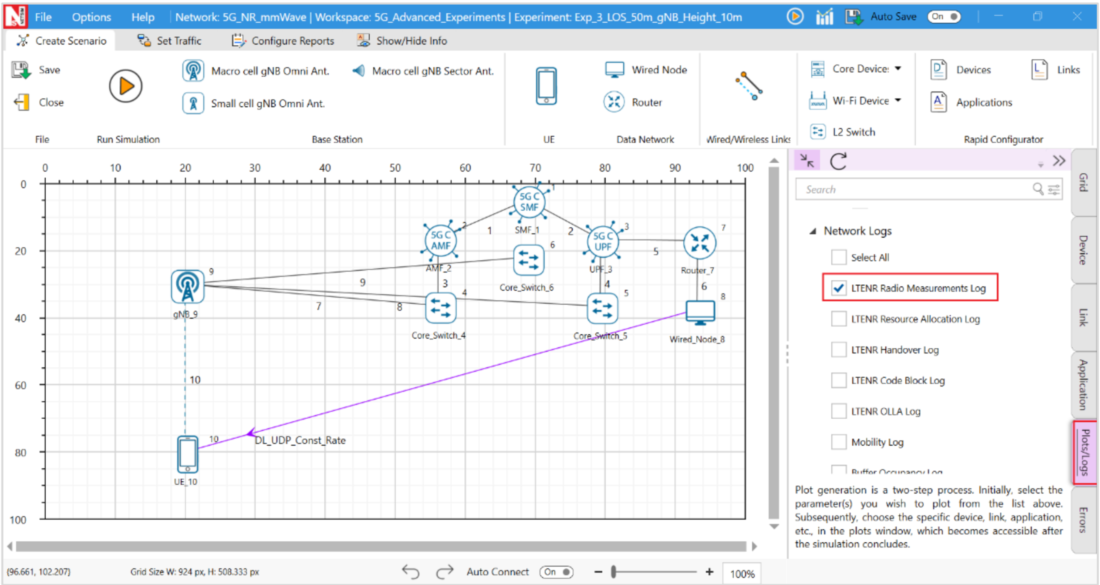


Figure 10-53: Enabling the LTENR Radio measurement log.

- After the simulation, note down the Pathloss from the log file generated for various gNB heights.

10.4.7 Case 02: 5G NR 3GPP Pathloss Models Distance 500m

1. Grid setting is 1000m x1000m.
2. Distance between gNB and UE is 500m.

10.4.8 Case 03: 5G NR 3GPP Pathloss Models Distance 1000m

1. Grid setting is 2000m x2000m.
2. Distance between gNB and UE is 1000m.

10.4.9 Results

1. After simulation, open LTENR Radio measurement Log file from Netsim Result dashboard.

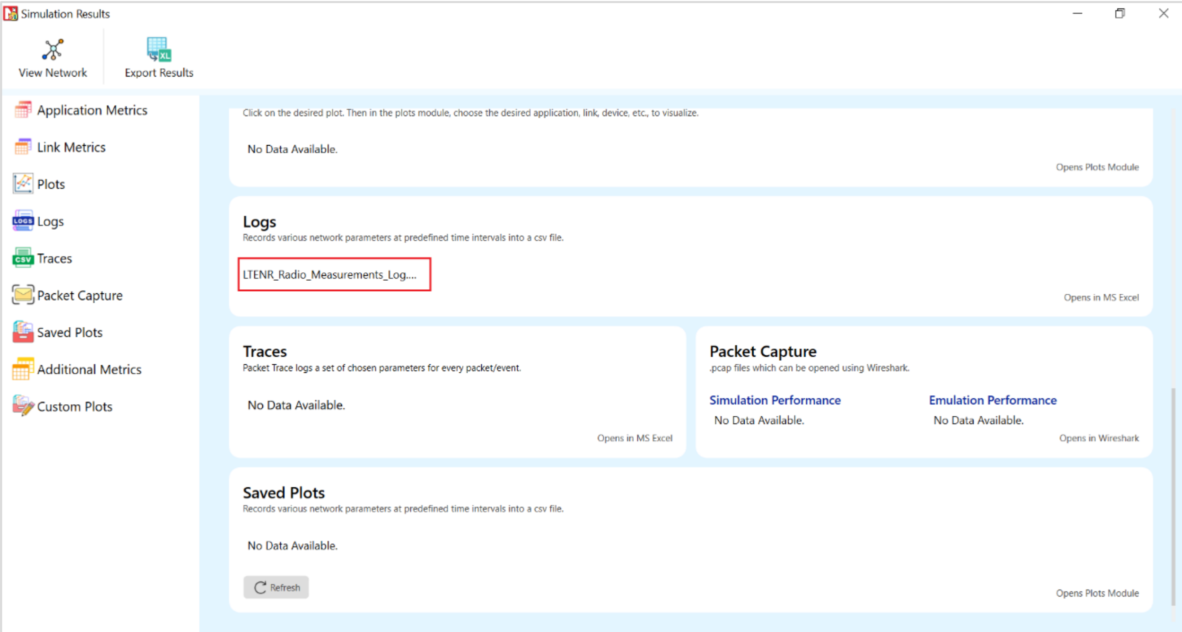


Figure 10-54: NetSim Results window

2. Note down the Pathloss value by filtering the channel to PDSCH, for each gNB height setting

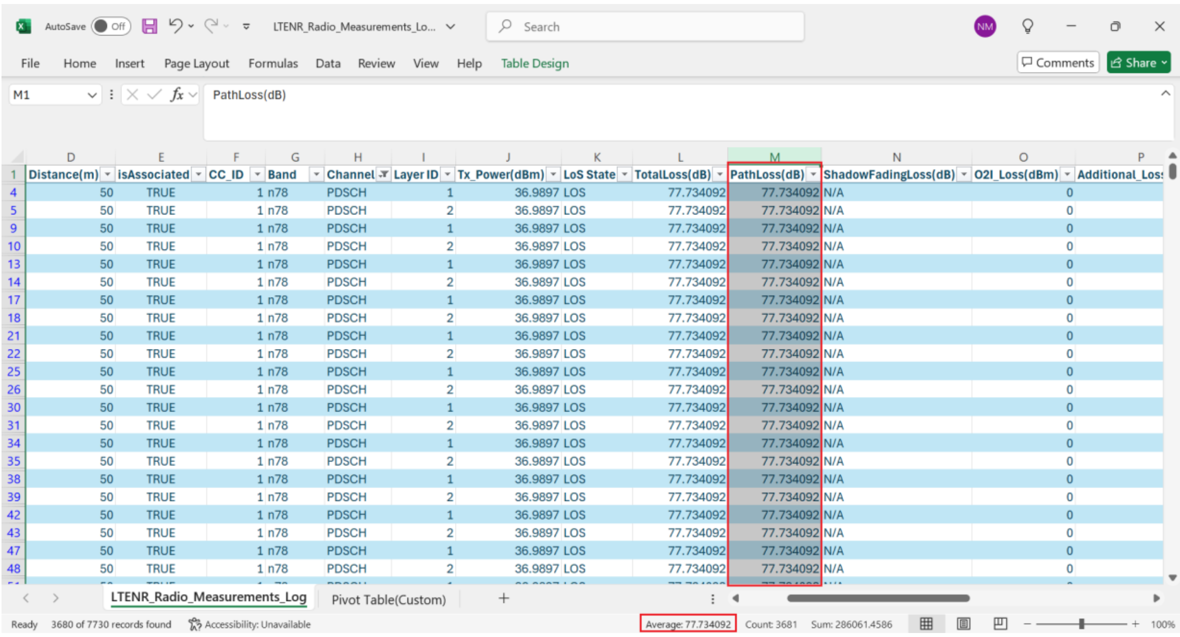


Figure 10-55: NetSim LTENR Radio measurement Log file

Upon running simulations, we can obtain the following table below which contains pathloss values for:

- gNB height varying from 10m to 150m in steps of 20m.
- UE placed at 50m, 500m and 1000m away from gNB, and
- UE states: LOS, NLOS

gNB Height(m)	Pathloss (dB)					
	UE 50m, LOS	UE 50m, NLOS	UE 500m, LOS	UE 500m, NLOS	UE 1 km, LOS	UE 1km, NLOS
10	77.73	92.39	98.71	132.47	105.58	144.61
30	78.87	84.04	98.73	120.54	105.58	132.21
50	80.58	82.57	98.76	115.30	105.59	126.73
70	82.35	82.72	98.80	111.92	105.60	123.16
90	83.99	83.99	98.86	109.45	105.62	120.51
110	85.45	85.45	98.93	107.52	105.64	118.41
130	86.75	86.75	99.02	105.96	105.66	116.68
150	87.91	87.91	99.12	104.66	105.69	115.20

Table 10-18: Pathloss values for various combinations. The gNB heights are shown in Column 1. Other columns show the gNB-UE 2D distance (50m, 500m and 1Km) and the UE state (LOS/NLOS)

Numerical verification of two cases

In this section we numerically calculate the pathloss per the 5G pathloss formula for two cases to verify NetSim's output.

Symbol	Description	Value
$\boldsymbol{d_{BP}}$	$Breakpoint\ distance$	
$\boldsymbol{h_{BS}}$	$Height\ of\ Base\ Station$	$10m$
$\boldsymbol{h_{UT}}$	$Height\ of\ UE$	$1.5m$
$\boldsymbol{f_c}$	$Central\ Frequency\ in\ Hz$	$3550 * 10^6 Hz = 3.55GHz$
$\boldsymbol{c}$	$Speed\ of\ light$	$3 * 10^8 m/s$
$\boldsymbol{W}$	$Street\ width$	$20m$
$\boldsymbol{h}$	$Building\ Height$	$5m$

Table 10-19: Various parameters used in the pathloss calculations and their values

Case 1: gNB height = 10m, UE State is LOS and UE-gNB 2D Distance = 50m

Breakpoint Distance:

$$f_c = \frac{F_{Low}+F_{High}}{2} = \frac{3300+3800}{2} = 3550\ MHz = 3550 * 10^6 Hz$$

$$d_{Bp} = 2 * \pi * h_{BS} * h_{UT} * (f_c * 1000000000) / c)$$

$$d_{BP} = 2 * 3.14 * 10 * 1.5 * \left(\frac{3.55 * 1000000000}{3 * 10^8}\right) = 1114.7m$$

Pathloss Calculation

$$d_{2D} = 50m,\ d_{3D} = \sqrt{(d_{2D})^2 + (H_{BS} - H_{UT})^2} = \sqrt{(50)^2 + (10 - 1.5)^2} = 50.71m$$

If $(10 \leq d_{2D} \leq d_{BP})$

$PL1 = (20 * log10(40 * PI * distance3D * fc_{(GHz)} / 3)) + fmin((0.03 * pow(h, 1.72)), 10) * log10(distance3D) - fmin((0.044 * pow(h, 1.72)), 14.77) + (0.002 * log10(h) * distance3D)$

$= \left(20 * log_{10}\left(40 * 3.14 * 50.71 * \frac{3.55}{3}\right)\right) + f_{min}\left((0.03 * pow(5,1.72)), 10\right) * log_{10}(50.71) - f_{min}\left((0.044 * pow(5,1.72)), 14.77\right) + (0.002 * log_{10}(5) * 50.71) = 77.73\ dB$

Pathloss = 77.73 dB (matches NetSim result)

Case 2: gNB height = 10m, UE State is NLOS and UE-gNB 2D Distance = 50m

Breakpoint Distance:

$$f_c = \frac{F_{Low} + F_{High}}{2} = \frac{3300+3800}{2} = 3550\ MHz = 3550 * 10^6 Hz$$

$$d_{Bp} = 2 * \ \pi * h_{BS} * h_{UT} * (f_c * \ 1000000000) / c)$$

$$d_{BP} = 2 * 3.14 * 10 * 1.5 * \left(\frac{3.55 * 1000000000}{3 * 10^8}\right) = 1114.7m$$

Pathloss Calculation

$$d_{2D} = 50m$$

$$d_{3D} = \sqrt{(d_{2D})^2 + (H_{BS} - H_{UT})^2} = \sqrt{(50)^2 + (10 - 1.5)^2} = 50.71m$$

If $(10 \leq d_{2D} \leq 5Km)$

$$PL_{NLOS} \ = \ \max\left(PL_{LOS}, PL'_{NLOS}\right)$$

Where,

$PL'_{NLOS} = 161.04 - 7.1 * \ \log_{10}(W) + 7.5 * \log_{10}(h) - \left(24.37 - 3.7 * \left(\frac{h}{h_{BS}}\right)^2\right) * \log_{10}(h_{BS}) + (43.42 - (3.1 * \log_{10}(h_{BS})) * (\log_{10}(d_{3D}) - 3) + 20 * (\log_{10}(f_c)) - (3.2 * (\log_{10}(11.75 * h_{UT}))^2 \ - 4.97)$

$= 161.04 - (7.1 * \log_{10}(20)) + 7.5 * (\log_{10}(5)) - (24.37 - 3.7 * \left(\frac{5}{10}\right)^2) * (\log_{10}(10)) + (43.42 - (3.1 * \log_{10}(10)) * (\log_{10}(50.71) - 3) + 20 * (\log_{10}(3.55)) - (3.2 * (\log_{10}(11.75 * 1.5))^2 - 4.97) = 92.39\ dB$

$PL_{LOS} = (20 * log10(40 * PI * distance3D * fc_{(GHz)} / 3)) + fmin((0.03 * pow(h, 1.72)), 10) * log10(distance3D) - fmin((0.044 * pow(h, 1.72)), 14.77) + (0.002 * log10(h) * distance3D)$

$$= \left(20 * log_{10}\left(40 * 3.14 * 50.71 * \frac{3.55}{3}\right)\right) + f_{min}\left((0.03 * pow(5{,}1.72)), 10\right) * log_{10}(50.71) - f_{min}\left((0.044 * pow(5{,}1.72)), 14.77\right) + (0.002 * log_{10}(5) * 50.71) = 77.73\ dB$$

$$PL_{NLOS} = \max(PL_{LOS}, PL'_{NLOS}) = \max(77.73{,}92.39)$$

Pathloss = 92.39 dB (matches NetSim result)

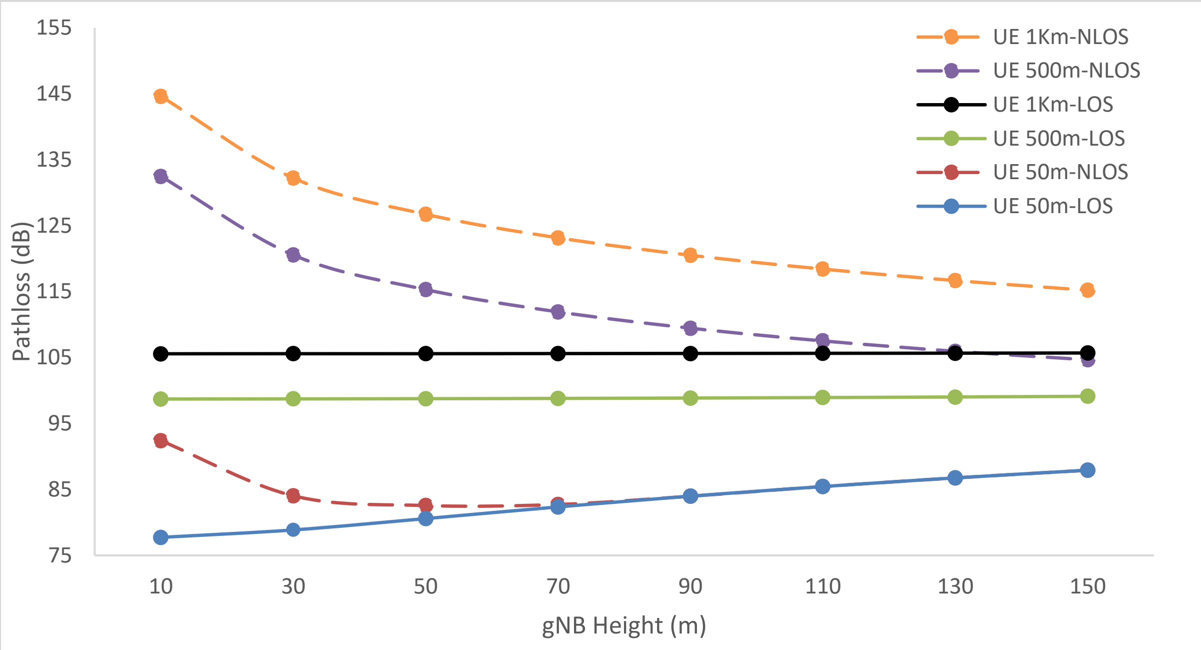


Figure 10-56: Plots of Pathloss vs. gNB height for different UE-gNB 2D Distances and UE States (LOS, NLOS)

Discussion

We explain the results in the plots above from the specifics of the pathloss formulas.

- In the LOS plots, the pathloss is flat for different gNB heights when the gNB-UE distance is high, i.e 500m and 1 km. When the gNB-UE distance is low i.e 50m, the pathloss increases with gNB height.
- Observe from the LOS pathloss formula that pathloss is proportional to $\log(D_{3d})$. D_{3d} of the 3D distance between the UE and the gNB and is defined as $d_{3D} = \sqrt{(d_{2D})^2 + (h_{BS} - h_{UT})^2}$. It is the hypotenuse of the right triangle with the base being the gNB-UE 2D distance.
    - Since the length of the hypotenuse is sensitive to the height of the triangle, when the base is small, we see the pathloss increasing with gNB height when the UE is 50m away.
    - Inversely, the length of the hypotenuse is almost insensitive to the triangle height when the base is much larger than the height. Therefore, when the UE

is far, the gNB's height does not have a noticeable impact. Pathloss is flat when the UE is 500m and 1 km away.

- Let us turn to the NLOS results.
  - The NLOS pathloss decreases with gNB height when the gNB-UE distance is high i.e 500m and 1000m.
  - When the UE is near, i.e 50m, the NLOS pathloss first decreases and then increases with gNB height.
  - The reason for this kind of variation is the NLOS pathloss formulas in which that pathloss has terms proportional to:
    - $\log(h_{BS}), \log((h_{BS})^2)$
    - $\log(d_{3d})$
    - The reciprocal of $(h_{BS})^2$
- We see that at larger distances LOS pathloss is almost flat and NLOS pathloss decreases, as gNB height increases. From the plots one sees that the optimal gNB height would be between 125m to 150m in the example discussed above.

Exercises

1. Make a separate plot with the UE distance on the X-axis and show the behavior for three different values of the gNB height. Recommend the gNB height for different cell radii. Does your recommendation make practical sense?
2. Use MATLAB or Python to plot similar curves from the standard pathloss formulas. Compare your results against the NetSim results.
3. (For the Instructor or TA) Generate *personalized* exercises where the student can be asked to
   a. Recommend the gNB height given the cell radius
   b. Recommend gNB height given the transmit power.

Find the cell radius given the gNB height, transmit power and noise figure

10.5 Performance of OFDMA SU-MIMO in 5G

The 5G cellular system utilizes OFDMA MIMO technology at the physical layer. This technology permits degrees of freedom in frequency, time, and "space" for multiplexing the data of multiple users. Let us consider the downlink direction, i.e., from the base-station (gNB) to the users (UE). The system bandwidth (e.g., 100 MHz), has several OFDM carriers, separated by a carrier spacing (e.g., 60 KHz, yielding 132 physical resource blocks (PRBs) or carriers). Each PRB has 12 consecutive sub carriers. These carriers (set of usable PRBs) have time-division framing (e.g., 0.25 ms), each frame carrying 14 symbols of which 18% on average is modeled as overheads in NetSim. When the above system is used to carry data to multiple users, it is called OFDMA. In addition, MIMO technology exploits spatial multiplexing, thereby effectively carrying multiple (spatially multiplexed) symbols for each time-symbol in the OFDM carrier. In MIMO all the spatially multiplexed symbols for an OFDM symbol can be destined to one user, in which case it is called Single User MIMO, or SU-MIMO.

In this experiment we study the performance of OFDMA along with SU-MIMO to carry downlink data to multiple UEs. Since we have SU-MIMO, multiple users are multiplexed by OFDMA, and SU-MIMO is used to obtain spatial multiplexing gain, depending on the number of antennas available at the UE. In Multi User MIMO (MU-MIMO) different spatial layers within the same resource block, can be allotted to different UEs.

10.5.1 Objective

Simulate a maximum data transmission rate[13], using OFDMA and SU-MIMO, for the following 4 cases

- 1 gNB with 8 Tx antennas transmitting to 1 UE with 8 Rx antennas using all the 132 resource blocks
- 1 gNB with 8 Tx antennas transmitting to 2 UEs each with 4 Rx antennas, each using $\frac{132}{2}$ resource blocks on average
- 1 gNB with 8 Tx antennas transmitting to 4 UEs each with 2 Rx antennas, each using $\frac{132}{4}$ resource blocks on average,
- 1 gNB with 8 Tx antennas transmitting to 8 UEs each with 1 Rx antennas, each using $\frac{132}{8}$ resource blocks on average

[13] We mean the saturation or full buffer case. There is always a packet in the gNB queue to transmit; the queue is never empty.

Repeat the above cases with the number of UEs now set to 1 for each case, while the antenna counts remain the same. Show that there is no difference in maximum capacity between a single user and a multi-user transmission, when using SU-MIMO. And finally, explain the results obtained for different number of receive antennas by using matrix theory to compute the eigen values of the associated Gram matrices.

10.5.2 Introduction

We begin with a description of the channel model. Consider a transmitter with N_t transmit antennas, and a receiver with $N_r$receive antennas. The channel can be represented by the $N_r \times N_t$ matrix $\boldsymbol{H}$ of channel gains h_{ij} representing the gain from transmit antenna j to receive antenna i. The $N_r \times 1$ received signal $\boldsymbol{y}$ is equal to

$$\boldsymbol{y} = \boldsymbol{Hx} + \boldsymbol{n}$$

The channel state information is the channel matrix $\boldsymbol{H}$ and/or its distribution.

Under rich scattering conditions the MIMO channel can be decomposed into parallel non-interfering channels. The number of such parallel streams is known as the layer count and is equal to $Min(N_t, N_r)$. These parallel channels are commonly referred to as the *eigenmodes* of the channel because the singular values of $\boldsymbol{H}$ are equal to the square root of the eigenvalues of the Wishart[14] matrix $W = H\,H^{\dagger}$ (for $N_t \geq N_r$).

Since each layer is reduced to a flat fading SISO channel, i.e., for layer $j, 1 \leq j \leq LayerCount$,

$$y_j = \sqrt{\lambda_j}\, x_j + w_j$$

where, x_j is the symbol transmitted, λ_j is the corresponding eigenvalue of the Wishart matrix obtained as in the previous section, w_j is circular symmetric complex Gaussian noise, and y_j is the complex valued baseband received symbol.

If fast fading with eigen-beamforming is enabled in NetSim's GUI, then the MIMO link is modelled by parallel SISO channels with the symbol level beamforming gain derived from the eigenvalues[15] of the Wishart matrix.

$$BeamFormingGain\ (dB) = 10\log_{10}(\lambda)$$

Three assumptions made in NetSim are:

[14] Explanation on the Wishart Matrix, $W = H\,H^{\dagger}$ and it's eigen values is provided in the experiment titled "MIMO Beamforming in 5G: A start with MISO and SIMO.

[15] Note that the eigenvectors are not required as they are only a part of the receive and transmit signal processing; NetSim only needs to work with the equivalent symbol-by-symbol flat fading SISO channels.

A1. Perfect CSIT and CSIR: The channel matrix $\boldsymbol{H}$ is assumed to be known perfectly, at the start of each frame, at the transmitter and receiver, respectively. With perfect CSIT the transmitter can adapt its transmission rate (MCS) relative to the instantaneous channel state (SNR).

A2. No channel errors.

A3. The transmit power is equally split between all layers transmitted. The justification lies in the fact that at a high SNR, (iterative) water-filling will lead to nearly equal power allocation across all subcarriers and all layers.

Note that the *LOS probability* parameter in NetSim is solely used to compute the large scale pathloss per the 3GPP 38.901 standard. This parameter is not used in the channel rank (MIMO layers) computations. The *Fading and Beam Forming* parameter is used to determine (i) the number of MIMO layers and (ii) the gains in each layer, as shown in the table below.

Parameter drop down option	No. of MIMO layers	Beamforming Gain
No fading MIMO unit gain	Min (N_t, N_r)	Unity (0 dB)
No fading MIMO array gain	Min (N_t, N_r)	Max (N_t, N_r)
Rayleigh with Eigen Beamforming	Min (N_t, N_r)	Eigen values of the Wishart Matrix

Table 10-20: Gains at different layers

10.5.3 Network simulation setup

Open NetSim and click on Experiments > 5G NR > Performance of OFDMA SU-MIMO in 5G then click on the tile in the middle panel to load the example as shown in below.

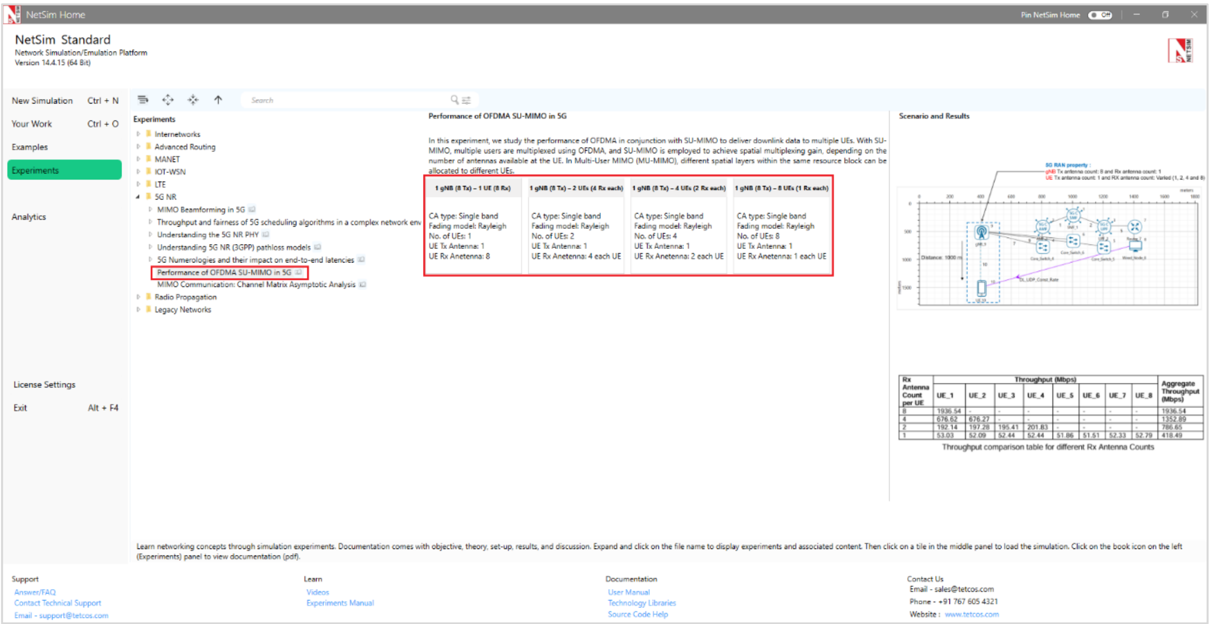


Figure 10-57: List of Experiments

10.5.4 NetSim Settings

1. The following parameters were configured in Interface 5G RAN- Physical Layer of gNB and UE:

gNB- Interface 5G_RAN Parameters	
gNB Height	10m
Tx Power	40 dBm
Duplex Mode	TDD
CA Type	SINGLE BAND
CA Configuration	n78
DL: UL Ratio	4:1
Numerology	2
Channel Bandwidth (MHz)	100
Tx Antenna Count	8
Rx Antenna Count	1
MCS Table	QAM256
CQI Table	TABLE2
Pathloss Model	3GPPTR38.901-7.4.1
Outdoor Scenario	Urban Macro
LOS NLOS Selection	User Defined
LOS Probability	1
Shadow Fading Model	None
Fast Fading Model	Rayliegh
Channel Rank/ MIMO layers	Max Rank
MIMO Beamforming Model	Eigen BF
Coherence Time (ms)	10
Additional Loss Model	None
UE Interface 5G RAN	
Tx Power	23 dBm
UE Height	1.5m
Tx Antenna Count	1
Rx Antenna Count	<varied>

Table 10-21: gNB and UE properties

2. The following parameters were configured in the wired link properties:

Wired Link Parameters	
Wired Link Speed	10 Gbps
Wired Link BER	0
Wired Link Propagation Delay	5 µs

Table 10-22: Wired link properties

3. Run simulation for 1.1s.

Case 1: 1 gNB - 8 Tx antennas, 1 UE - 8 Rx antennas

Network Scenario:

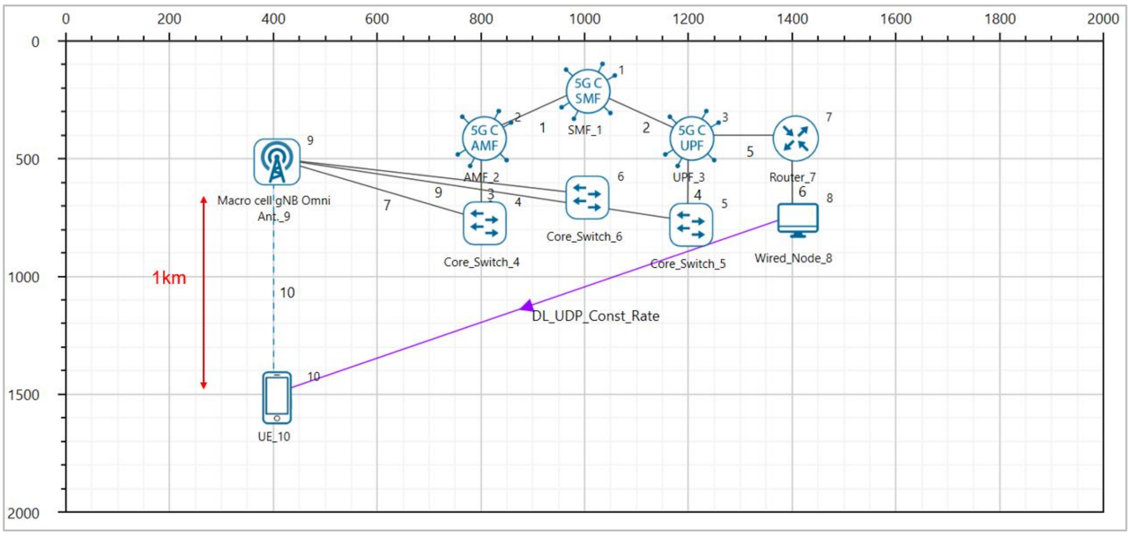


Figure 10-58: Network topology in this experiment

Additional Settings:

1. The Tx Antenna Count was set to 1 and Rx Antenna Count was set to 8 in Interface 5G RAN- Physical Layer in the UE
2. The following parameters were set in Application Properties:

Application Parameters	
Application	CBR
Packet Size	1460
Inter Packet Arrival Time (µs)	3.33
Start Time	1
Transport Protocol	UDP

Table 10-23: Application properties

Result:

Application	Throughput (Mbps)
App_1_CBR	1950.56

Table 10-24: Throughput obtained per UE

Case 2: 1 gNB- 8 Tx antennas, 2 UEs with 4 Rx antennas each

Network Scenario:

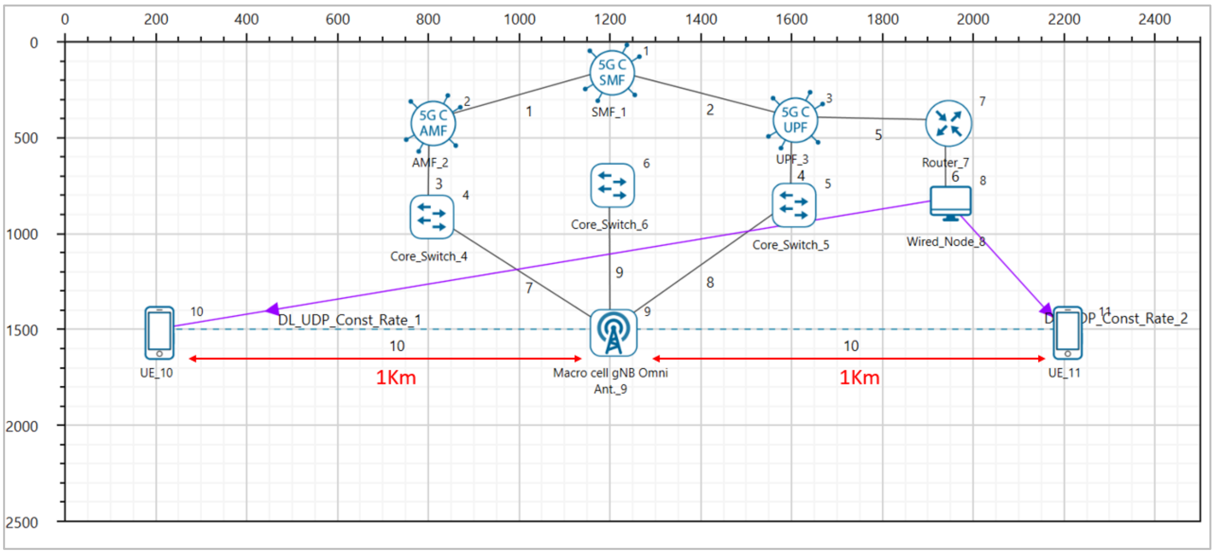


Figure 10-59: Network topology in this experiment

Additional Settings:

1. The Tx Antenna Count was set to 1 and Rx Antenna Count was set to 4 in Interface 5G RAN- Physical Layer in both the UEs.
2. The following parameters were set in Application Properties:

Application Parameters		
	Wired Node- UE_10	Wired Node- UE_11
Application	CBR	CBR
Packet Size	1460	1460
Inter Packet Arrival Time (µs)	12.97	12.97
Start Time	1	1
Transport Protocol	UDP	UDP

Table 10-25: Application properties

Result:

Throughput Obtained (Mbps)		
UE_10	UE_11	Aggregate Throughput (Mbps)
685.26	659.92	1351.03

Table 10-26: Throughput obtained per uE

Case 3: 1 gNB 8- Tx antennas, 4 UEs with 2 Rx antennas each

Network scenario:

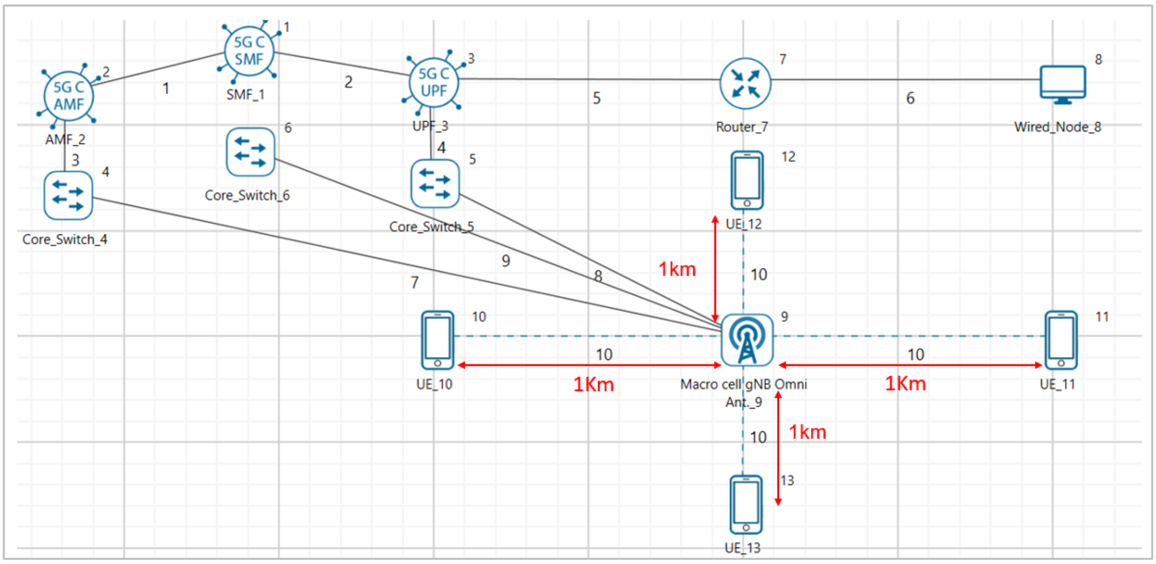


Figure 10-60: Network topology in this experiment

Additional Settings:

1. The Tx Antenna Count was set to 1 and Rx Antenna Count was set to 2 in Interface 5G RAN- Physical Layer in all the UEs.
2. The following parameters were set in Application Properties:

Application Parameters				
	Wired Node-UE_8	Wired Node-UE_9	Wired Node-UE_10	Wired Node-UE_11
Application	CBR	CBR	CBR	CBR
Packet Size	1460	1460	1460	1460
Inter Packet Arrival Time (µs)	53.09	53.09	53.09	53.09
Start Time	1	1	1	1
Transport Protocol	UDP	UDP	UDP	UDP

Table 10-27: Application properties

Result:

Throughput (Mbps)				
UE_10	UE_11	UE_12	UE_13	Aggregate Throughput (Mbps)
197.39	197.98	197.74	200.20	793.31

Table 10-28: Throughputs obtained per UE

Case 4: 1 gNB 8 tx antennas, 8 UEs with 1 Rx antennas each

Network Scenario:

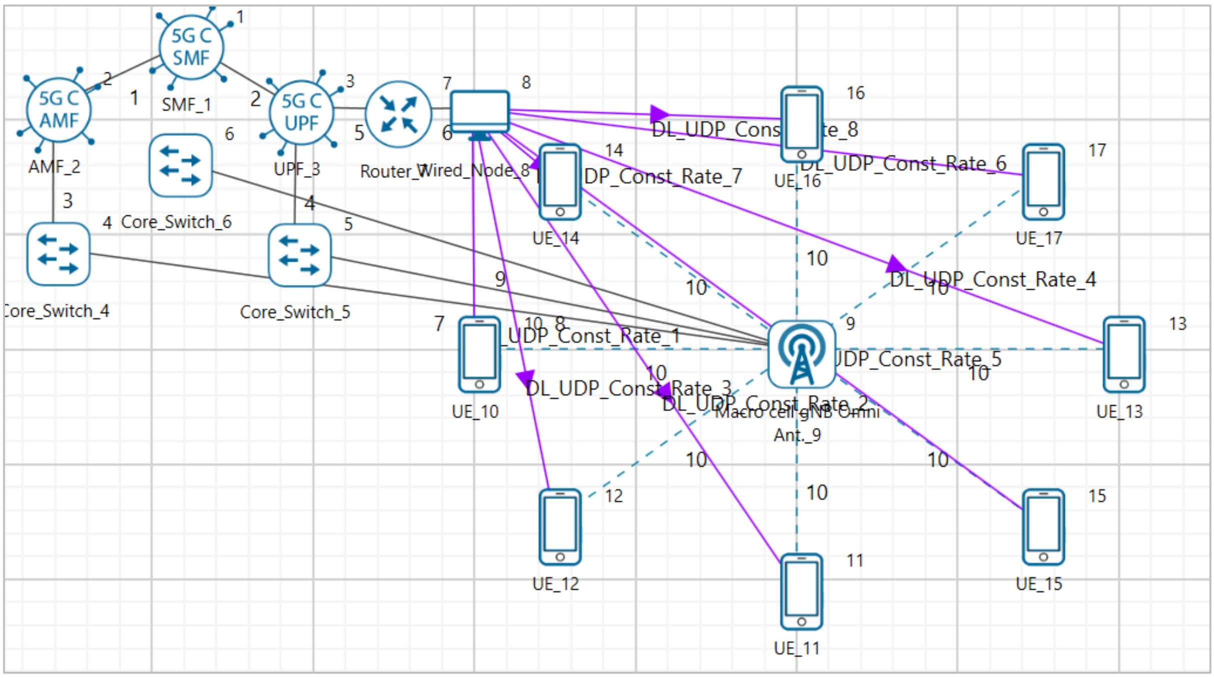


Figure 10-61: Network topology in this experiment

Additional Settings:

1. The Tx Antenna Count was set to 1 and Rx Antenna Count was set to 1 in Interface 5G RAN- Physical Layer in all the UEs.
2. The following parameters were set in Application Properties:

Application Parameters								
	Wired Node-UE_10	Wired Node-UE_11	Wired Node-UE_12	Wired Node-UE_13	Wired Node-UE_14	Wired Node-UE_15	Wired Node-UE_16	Wired Node-UE_17
Application	CBR	CBR	CBR	CBR	CBR	CBR	CBR	CBR
Packet Size	1460	1460	1460	1460	1460	1460	1460	1460
Inter Arrival Time (µs)	212.36	212.36	212.36	212.36	212.36	212.36	212.36	212.36
Start Time	1	1	1	1	1	1	1	1
Transport Protocol	UDP	UDP	UDP	UDP	UDP	UDP	UDP	UDP

Table 10-29: Application properties

Results:

Throughput Obtained (Mbps)								
UE_10	UE_11	UE_12	UE_13	UE_14	UE_15	UE_16	UE_17	Aggregate Throughput (Mbps)
51.98	52.79	52.09	52.21	53.03	50.22	51.63	51.63	415.57

Table 10-30: Throughputs obtained per UE

10.5.5 Discussion

We combine the results of the four cases and present it in the table below.

Rx Antenna Count per UE	Throughput (Mbps)								Aggregate Throughput (Mbps)
	UE_1	UE_2	UE_3	UE_4	UE_5	UE_6	UE_7	UE_8	
8	1981.39	-	-	-	-	-	-	-	1981.39
4	690.05	688.76	-	-	-	-	-	-	1378.81
2	201.94	196.34	204.4	199.02	-	-	-	-	801.7
1	51.74	51.97	52.79	52.44	51.97	51.62	51.05	52.79	416.85

Table 10-31: Throughput comparison table for different Rx Antenna Counts

We compare the aggregate throughputs (from Table 10-31) with single UE peak throughputs.

UE Rx Antenna Count per UE	Aggregate Throughput (Mbps)	Single UE Peak Throughput (Mbps)
8	1936.54 (1 UE 8 Rx Antennas)	1936.54 (1 UE 8 Rx Antennas)
4	1352.89 (2 UEs with 4 Rx Antennas)	1330.46 (1 UE 4 Rx Antennas)
2	786.65 (4 UEs with 2 Rx Antennas)	754.52 (1 UE 2 Rx Antennas)
1	418.49 (8 UEs with 1 Rx Antennas)	414.75 (1 UE 1 Rx Antennas)

Table 10-32: Throughputs obtained for different Rx Antenna counts in multi and single UE cases.

From Table 10-32 it becomes clear that the bandwidth is shared across the UEs by OFDMA. It is just like having one UE use the entire bandwidth.

Theoretical analysis:

8*8 SU-MIMO

- 2^k carriers in the OFDMA system
- S: symbol rate per carrier (assuming the same numerology for all carriers)
- L: is the large-scale pathloss, including shadowing (not including the fading). Note that, L is modelled as $c\left(\frac{d}{d_0}\right)^{-\eta}$ where η is the pathloss coefficient.

Then the expected rate to the single SU-MIMO user is given by

$$\mathbb{E}(R_1) = 2^k \times S \times \mathbb{E}\left(\sum_{j=1}^{8} log\left(1 + \frac{P_j \times L \times \lambda_j}{\sigma^2}\right)\right)$$

where P_j is the power allotted to the j^{th} equivalent channel, with $\sum_{j=1}^{8} P_j = P,$ the total transmit power. We assume equal power allocation, so that $P_j = \frac{P}{8},$ so that

$$\mathbb{E}(R_1) = 2^k \times S \times \mathbb{E}\left(\sum_{j=1}^{8} log\left(1 + \frac{P \times L \times \lambda_j}{8\sigma^2}\right)\right)$$

8*1 SU-MIMO, with 8 such UEs

- 2^{k-3} OFDMA carriers per UE

Then, for the same OFDMA symbol rate, $S,$ and large-scale pathloss, $L,$ the total expected rate for the 8 UEs is given by:

$$\mathbb{E}(R_2) = 2^{k-3} \times S \times \left(\sum_{j=1}^{8} \mathbb{E}\, log\left(1 + \frac{P \times L \times \|h_j^2\|}{\sigma^2}\right)\right) = 2^k \times S \times \mathbb{E}\log\left(1 + \frac{P \times L}{\sigma^2} \times \|h_1^2\|\right)$$

From matrix theory we know that the sum of the eigenvalues of a matrix is equal to its trace. Therefore

$$\sum_{i=1}^{8} \lambda_i = \|h_1^2\| + \|h_2^2\| \dots + \|h_8^2\|$$

We know that when $a, \lambda_1, \lambda_2 > 0$

$$1 + a(\lambda_1 + \lambda_2) \leq (1 + a\lambda_1)(1 + a\lambda_2)$$

and by monotonicity of log function, it follows that

$$log\big(1 + a(\lambda_1 + \lambda_2)\big) \leq log(1 + a\lambda_1) + log(1 + a\lambda_2)$$

Extending this inequality recursively, we get

$$log\big(1 + a(\lambda_1 + \lambda_2 \dots \lambda_8)\big) \leq log(1 + a\lambda_1) + log(1 + a\lambda_2) \dots + log\ (1 + a\lambda_8)$$

Using the above expressions, we can compare the expected total downlink throughputs for one 8x8 SU-MIMO UE, with eight 8x1 SU-MIMO UEs, as follows:

$$\mathbb{E}R_1 = 2^k \times S \times \mathbb{E}\left(\sum_{j=1}^{8} log\left(1 + \frac{P \times L \times \lambda_j}{8\sigma^2}\right)\right) \geq 2^k \times S \times \mathbb{E}\, log\left(1 + \frac{P \times L}{8\sigma^2} \times \sum_{j=1}^{8} \lambda_j \right)$$

$$= 2^k \times S \times \mathbb{E}\log\ (\ 1 + \frac{P \times L}{8\sigma^2} \times \sum_{j=1}^{8} \|h_j^2\|)$$

$$\geq 2^k \times S \times \sum_{j=1}^{8} \frac{1}{8}\mathbb{E}\log\left(1 + \frac{P \times L}{\sigma^2} \times \|h_j^2\|\right)$$

$$= 2^k \times S \times \mathbb{E}\log\left(1 + \frac{P \times L}{\sigma^2} \times \|h_1^2\|\right) = \mathbb{E}\, R_2$$

where the first inequality arises from the argument given earlier, and the second equality follows from the equality between the trace of a matrix and the sum of its eigenvalues. Physically, the first inequality captures the performance gain obtained by splitting the power over the different spatial degrees of freedom. The second inequality follows from the fact the fact that the logarithm function is concave and by an application of Jensen's inequality. We see from Table 10-31 that $\mathbb{E}R_1 = 1942.26\ Mbps,$ whereas $\mathbb{E}R_2 = 418.46\ Mbps$, showing the large effect of the two inequalities in the above argument.

10.5.6 Exercises

1. Carefully explain your observations. Also, place the UEs at different distances from the gNB and see how the throughput changes. Again, explain your observations.
2. Vary the height of UEs (as this also effects pathloss) and see how the throughput changes. Again, explain your observations.

10.6 5G Numerologies and their impact on end-to-end latencies

10.6.1 Objective

5G NR supports flexible numerology with a range of subcarrier spacings, based on scaling a baseline subcarrier spacing of 15 kHz to support diverse spectrum bands/types and deployment models. The numerology, μ, can take values from 0 to 4 and specifies the Sub-Carrier Spacing (SCS) as $15 \times 2^{\mu}$ kHz and a slot length of $\frac{1}{2^{\mu}}$ ms. With μ varying from 0 to 4, Sub- Carrier Spacing (SCS) varies from 15 to 240 kHz.

We investigate the impact of numerology on latency and throughput in two cases

- A simple case where one UE is transmitting and receiving UDP traffic from a server
- A complex 5G scenario with Sensors, Cameras and Smartphones having DL and UL, TCP and UDP flows[16].

10.6.2 Theory

In NetSim, for data channels FR1 supports $\mu = 0, 1, 2$ and FR2 supports $\mu = 2, 3$. The setting $\mu = 0$ corresponds to the LTE (4G) system configuration. In the time domain (to support backwards compatibility with LTE) the frame length in 5G NR is set to 10 ms, and each frame is composed of 10 subframes of 1 ms each. The 1 ms subframe is then divided into one or more slots in 5G, whereas LTE had exactly two slots in a subframe. The slot size is defined based on μ, and the number of slots is 2^{μ}. The number of OFDM symbols per slot is 14 for a configuration using normal cyclic prefix. For extended cyclic prefix, the number of OFDM symbols per slot is 12.

---

[16] This is adapted from (Patriciello, Lagen, Giupponi, & Bojovic, 2018)

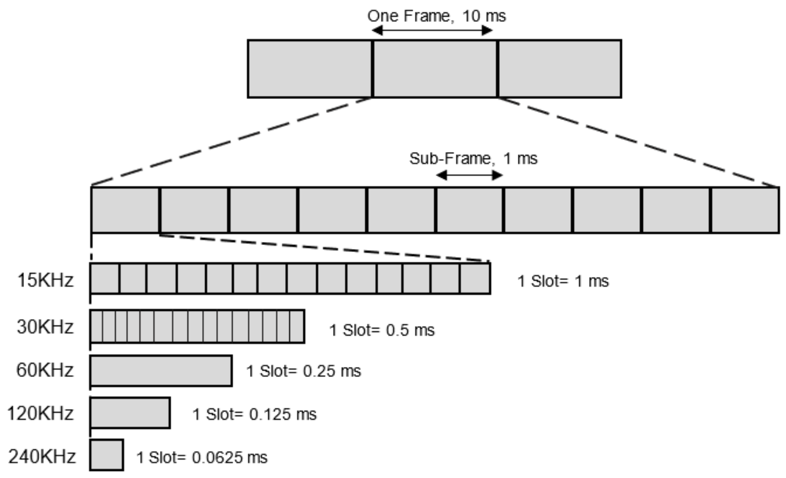


Figure 10-62: Frame, subframe and slot structure for different numerologies

For $\mu = 0$ there are 1 slot per subframe, for $\mu = 1$ there are 2 slots per subframe, for $\mu = 2$ there are 4 slots per subframe and so on. The number of slots per frame is ten times of number of slots per sub frame. Hence for $\mu = 2$, there are 40 slots/frame.

Numerology	Sub-Carrier Spacing (KHz)	OFDM Symbols per Slot	Slots per Frame	Slots per Sub-frame
0	15	14	10	1
1	30	14	20	2
2	60	14	40	4
3	120	14	80	8
4	240	14	160	16

Table 10-33: Table 1 1:Sub-carrier spacing, number of OFDM symbols per slot, slots

10.6.3 Network simulation setup

Open NetSim and click on Experiments > 5G NR > 5G Numerologies and their impact on end-to-end latencies then click on the tile in the middle panel to load the example as shown in below.

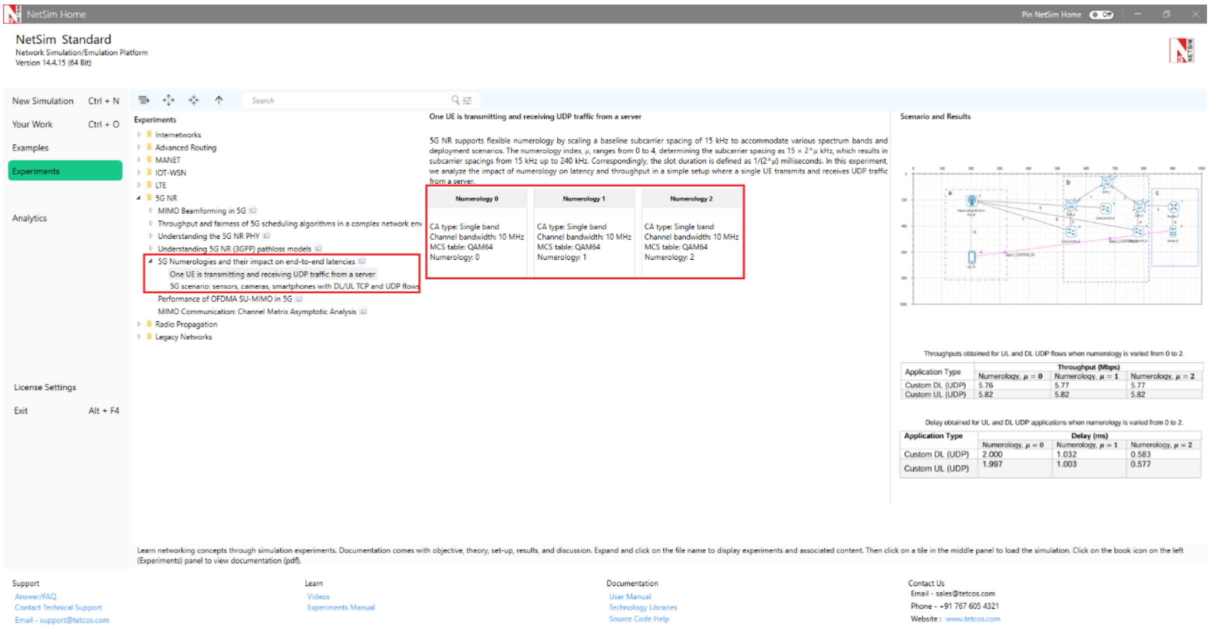


Figure 10-63: List of Experiments

10.6.4 Network Model

10.6.5 Case 1: One UE is transmitting and receiving UDP traffic from a server

This is a simple scenario whereby the UE is transmitting and receiving UDP traffic from a server as shown in Figure 10-82.

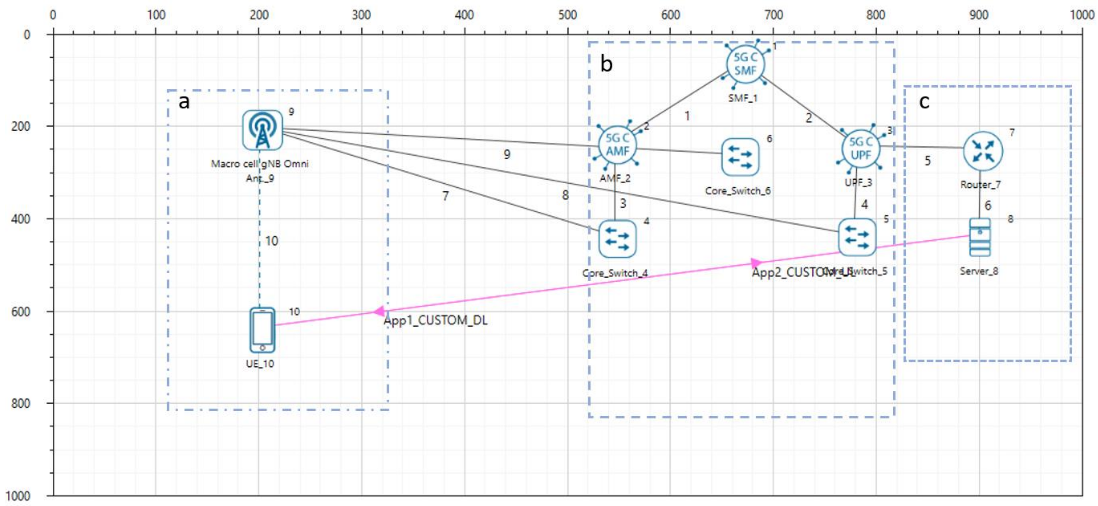


Figure 10-64: Network scenario. a) The RAN with a User Equipment b) 5G Core and C) Cloud Server. The device in the RAN has both UL and DL communication with the cloud server.

The UE connects to the gNB which connects to the 5G core. The 5G core then connects to the remote server over the cloud (represented by the router and WAN links).

Keeping all other parameters fixed, we vary the numerology μ, as 0, 1, 2 and see its impact on end-to-end latency and application (user) throughput. In terms of application data traffic, the User Equipment (UE) has two UDP flows, one Uplink and one Downlink, that goes in the UL towards a remote node on the Internet. These flows are fixed-rate flows:

10.6.6 Procedure

1. For the above scenario set the following properties:

gNB Properties -> Interface (5G_RAN)	
Pathloss Model	None
Frequency Range	FR1
CA Type	Single Band CA
CA Configuration	n78
Numerology	0,1 and 2
Channel Bandwidth	10 MHz
DL: UL Ratio	1:1
MCS Table	QAM64
CQI Table	Table 1

Table 10-34:The Physical Layer properties set in 5G RAN interface of gNB

Link Properties (All wired links)	
Uplink/ Downlink Speed (Mbps)	10000
Uplink/ Downlink BER	0
Uplink/ Downlink Propagation Delay (µs)	0

Table 10-35:Wired Link properties set in this experiment

CUSTOM UL UDP	
Generation Rate (Mbps)	5.8
Transport Protocol	UDP
Application Type	Custom
Start time (s)	1
Packet Size (Bytes)	1460
IAT Distribution	Exponential
Inter Arrival Time (µs)	2000

Table 10-36: Custom application properties for UL UDP

CUSTOM DL UDP	
Generation Rate (Mbps)	5.8
Transport Protocol	UDP
Application Type	Custom
Start time (s)	1
Packet Size (Bytes)	1460
IAT Distribution	Exponential
Inter Arrival Time (µs)	2000

Table 10-37:Custom application properties for DL UDP

2. The Tx Antenna Count was set to 2 and Rx Antenna Count was set to 2 in gNB > Interface 5G_RAN >Physical Layer.
3. The Tx Antenna Count was set to 2 and Rx Antenna Count was set to 2 in UE > Interface 5G_RAN >Physical Layer.
4. Run simulation for 10 sec. After simulation completes go to metrics window and note down throughput and delay value from application metrics.

10.6.7 Case 2: A complex 5G scenario with Sensors, Cameras and Smartphones having DL and UL, TCP and UDP flows

To model a real-world scenario, we base our simulation on the setup shown in Figure 10-65**.** The link between the gNB and the L2_Switches that represents the Core Network (CN) is made with a point-to-point 10 Gb/s link, without propagation delay. The Radio Area Network (RAN) is served by 1 gNB, in which different UEs share the connectivity. We have 25 smartphones, 6 sensors, and 3 IP cameras. The bandwidth is 100MHz and Round Robin MAC Scheduler.

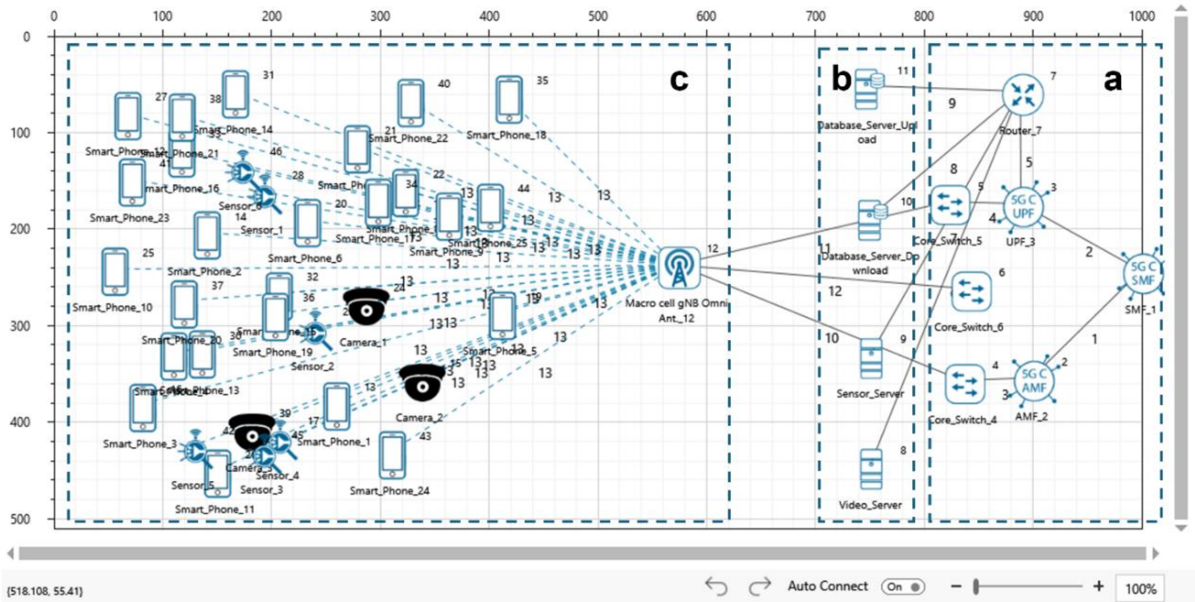


Figure 10-65: Network scenario. a) Cloud servers b) 5G Core and C) The RAN with 25 smartphones, 6 sensors and 3 cameras communicating. The devices in the RAN communicate with respective cloud servers for both Downloads and Uploads.

In terms of data traffic, the camera (video) and sensor nodes have one UDP flow each, that goes in the UL towards a remote node on the Internet. These flows are fixed-rate flows: we have a continuous transmission of 5 Mb/s for the video nodes, to simulate a 720p24 HD video, and the sensors transmit a payload of 500 bytes each 2.5 ms, that gives a rate of 1.6 Mb/s. For smartphones, we use TCP as the transmission protocol. These connect to database servers. Each phone downloads a 25MB file and uploads a 1.5MB file. These flows start at different times: the upload starts at a random time between the 4.5th and the 105th simulation seconds, while each download starts at a random time between the 1.5th and the 105th simulation seconds.

	Flows (No of devices)	Traffic Rate (Mbps)	Segment / File Size (B)	Traffic Dir.	TCP ACK Dir.
Camera (UDP)	3	5	500	UL	-
Sensor (UDP)	6	1.6	500	UL	-
Smartphone Upload (TCP)	25	-	1,500,000	UL	DL
Smartphone Download (TCP)	25	-	25,000,000	DL	UL

Table 10-38:Various parameters of the Traffic flow model for all the devices.

The numerology μ can take values from 0 to 3 and specifies an SCS of $15 \times 2^{\mu}$ kHz and a slot length of $\frac{1}{2^{\mu}}$ ms. FR1 support $\mu = 0, 1$ and 2, while FR2 supports $\mu = 2, 3$. We study the impact of different numerologies, and how they affect the end-to-end performance. The metrics measured and analyzed are a) Throughput of TCP uploads and b) Latency of the UDP uploads and downloads.

10.6.8 Procedure

1. For the above scenario set the following properties:

gNB Properties -> Interface (5G_RAN)	
Pathloss Model	None
Frequency Range	FR1
CA Type	Inter Band CA
CA Configuration	CA_2DL_2UL_n40_n41
CA1	
Numerology	0, 1 and 2
Channel Bandwidth	50 MHz
DL: UL Ratio	1:4
CA2	
Numerology	0, 1 and 2
Channel Bandwidth	50 MHz
DL: UL Ratio	1:4
MCS Table	QAM64
CQI Table	Table 1

Table 10-39: The Physical Layer properties set in 5G Ran interface of gNB

Phone UL TCP	
Application Type	FTP
Transport Protocol	TCP
Start Time (s)	$4.5 + 4(i-1) where, i = 1,2, \dots .25$
Stop Time (s)	105
File Size (B)	1,500,000
Inter Arrival Time (µs)	200 (simulation ends at 100s and hence only one file is sent)

Table 10-40: Phone applications for UL TCP

Link Properties (All wired links)	
Uplink/ Downlink Speed (Mbps)	10000
Uplink/ Downlink BER	0
Uplink/ Downlink Propagation Delay (µs)	5

Table 10-41: Wired link properties set in this experiment

Sensor UL UDP	
Generation Rate (Mbps)	1.6
Transport Protocol	UDP
Application Type	Custom
Packet Size (Bytes)	500
Inter Arrival Time (µs)	2500

Table 10-42: Sensor application properties for UL UDP

Camera UL UDP	
Generation Rate (Mbps)	5
Transport Protocol	UDP
Application Type	Custom
Packet Size (Bytes)	500
Inter Arrival Time (µs)	800

Table 10-43:Camera application properties for UL UDP

Phone DL TCP	
Application Type	FTP
Transport Protocol	TCP
Start Time (s)	$1.5 + 4(i-1) where, i = 1,2, \dots .25$
Stop Time (s)	105
File Size (B)	25,000,000
Inter Arrival Time (µs)	200(simulation ends at 100s and hence only one file is sent)

Table 10-44: Phone application properties for DL TCP

2. The Tx Antenna Count was set to 2 and Rx Antenna Count was set to 2 in gNB > Interface 5G_RAN >Physical Layer.
3. The Tx Antenna Count was set to 2 and Rx Antenna Count was set to 2 in UE > Interface 5G_RAN >Physical Layer.
4. Run simulation for 110 sec. After simulation completes go to metrics window and note down throughput and delay value from application metrics.

10.6.9 Results

Case 1:

Application Type	Throughput (Mbps)		
	Numerology, $\mu = 0$	Numerology, $\mu = 1$	Numerology, $\mu = 2$
Custom DL (UDP)	5.76	5.77	5.77
Custom UL (UDP)	5.82	5.82	5.82

Table 10-45: Throughputs obtained for UL and DL UDP flows when numerology is varied from 0 to 2.

Application Type	**Delay (ms)**		
	Numerology, $\mu = 0$	Numerology, $\mu = 1$	Numerology, $\mu = 2$
Custom DL (UDP)	2.000	1.032	0.583
Custom UL (UDP)	1.997	1.003	0.577

Table 10-46: Delay obtained for UL and DL UDP applications when numerology is varied from 0 to 2.

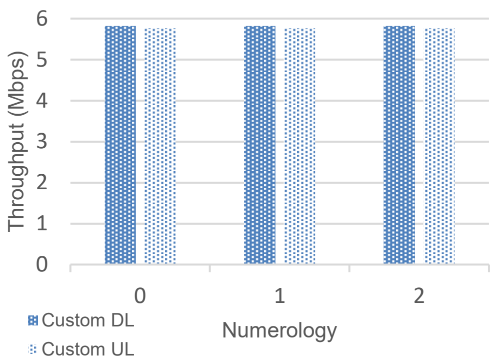


Figure 10-66: Custom DL and UL throughput vs numerology. Numerology has no impact on throughput

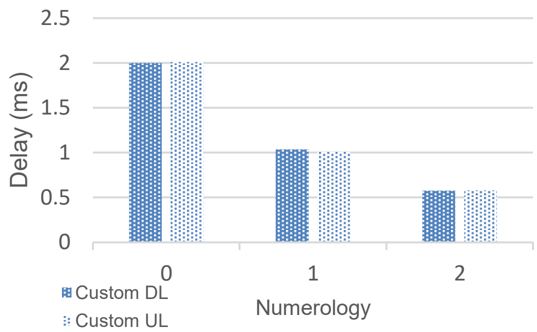


Figure 10-67: Custom DL and UL delay vs numerology. The delay for both DL and UL decreases as numerology is increased

Case 2:

Application Type	Average Throughput (Mbps)		
	Numerology, $\mu = 0$	Numerology, $\mu = 1$	Numerology, $\mu = 2$
Camera Video UL (UDP)	4.99	4.99	4.99
Sensor UL (UDP)	1.59	1.59	1.59
Smartphone UL (TCP)	86.30	172.52	311.42
Smartphone DL (TCP)	101.67	156.43	151.74

Table 10-47: Average and aggregate throughputs obtained for Camera, Sensors and Smartphones, when numerology is varied from 0 to 2.

Application Type	**Average Delay (ms)**		
	Numerology, $\mu = 0$	Numerology, $\mu = 1$	Numerology, $\mu = 2$
Camera Video UL (UDP)	1.82	0.92	0.47
Sensor UL (UDP)	2.28	1.52	0.77
Smartphone UL (TCP)	79.71	39.96	21.85
Smartphone DL (TCP)	992.32	643.28	660.70

Table 10-48: Average delay obtained for Camera, Sensors and Smartphones, when numerology is varied from 0 to 2

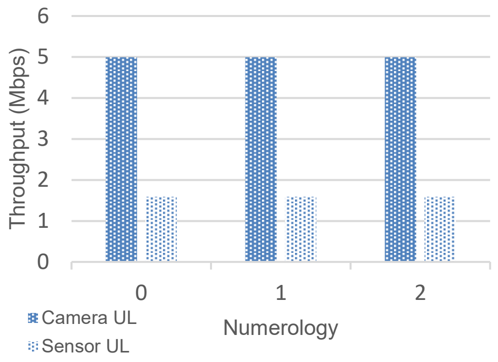


Figure 10-68: The average uplink throughputs for Cameras and Sensors remains the same as the numerology is increased. This is because the flow is UDP.

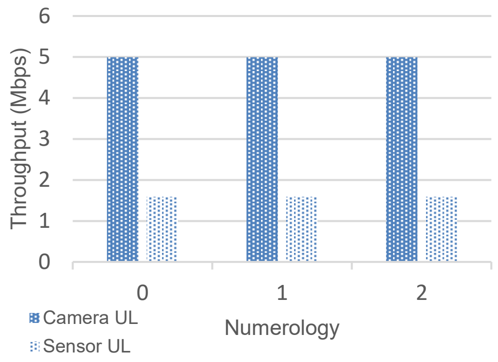


Figure 10-69: The average uplink delays for cameras and sensors decreases as the numerology is increases

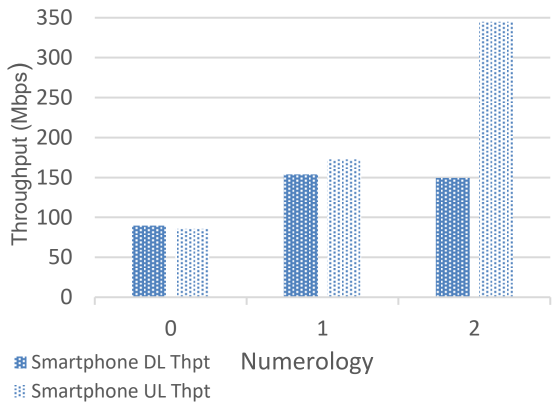


Figure 10-70: The average throughput for smartphone downlink and uplink increases with numerology. This is because the flow is TCP

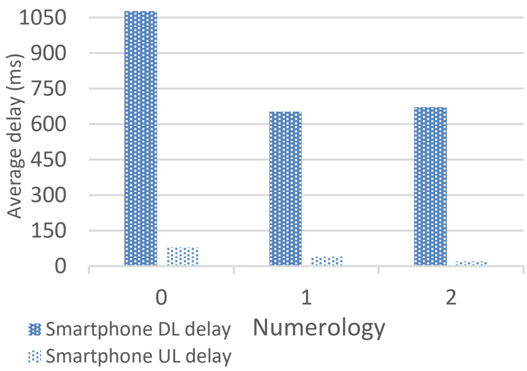


Figure 10-71: The average delay for smartphone downlink and uplink decreases with numerology. This is because the flow is TCP.

10.6.9.1 Discussion

For UDP applications, the Numerology, μ does not impact the throughput.

The TCP throughput is inversely proportional to round trip time. Therefore, for applications running over TCP the throughput increases with higher numerology since a higher Numerology leads to reduced round-trip times.

Therefore, the selection of the numerology in an NR system should be carefully made by considering the traffic patterns.

10.6.9.2 References

1. Patriciello, N., Lagen, S., Giupponi, L., & Bojovic, B. (2018). 5G New Radio Numerologies and their Impact on the End-To-End Latency. IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD).

10.7 MIMO Communication: Channel Matrix Asymptotic Analysis

10.7.1 Objective

In this experiment, properties of MIMO channel matrices in 5G wireless communications are studied in the asymptotic number of antennas. In particular, the condition number of a large MIMO matrix, which dictates the performance of spatial multiplexing and subsequently the behaviour of the eigen spectrum of large MIMO matrices, are investigated through simulations setup in NetSim v14.4.

10.7.2 Introduction

A MIMO channel matrix in wireless communications is, in general, a random matrix. Hence, all the parameters obtained out of matrix entries are essentially random variables. For e.g., the eigen values, the condition number are all random variables. In particular, the condition number of a MIMO channel matrix is of significant importance in characterizing the capacity of a MIMO channel. For example, by virtue of Jensen's inequality we have,

$$\sum_i \log\left(1+\frac{P\sigma_i}{N}\right) \leq r\log\left(1+\frac{1}{r}\sum_i \frac{P\sigma_i}{N}\right),$$

with equality holding true iff condition number of the MIMO matrix is 1 and other parameters have usual meanings, and with r being the rank of the MIMO channel matrix. Thus, for a given rank and SNR of the channel, condition number of the matrix determines how close to the upper bound the achievable rate is. In this view, the eigen spectrum and condition number distribution of large MIMO matrices are investigated. By means of theory, it will be shown that the condition number of a large MIMO matrix stabilizes near unity, indicating superior spatial multiplexing capability in the asymptotic number of antennas.

10.7.3 Network simulation setup

Open NetSim and click on Experiments > 5G NR > MIMO Communication: Channel Matrix Asymptotic Analysis then click on the tile in the middle panel to load the example as shown in below.

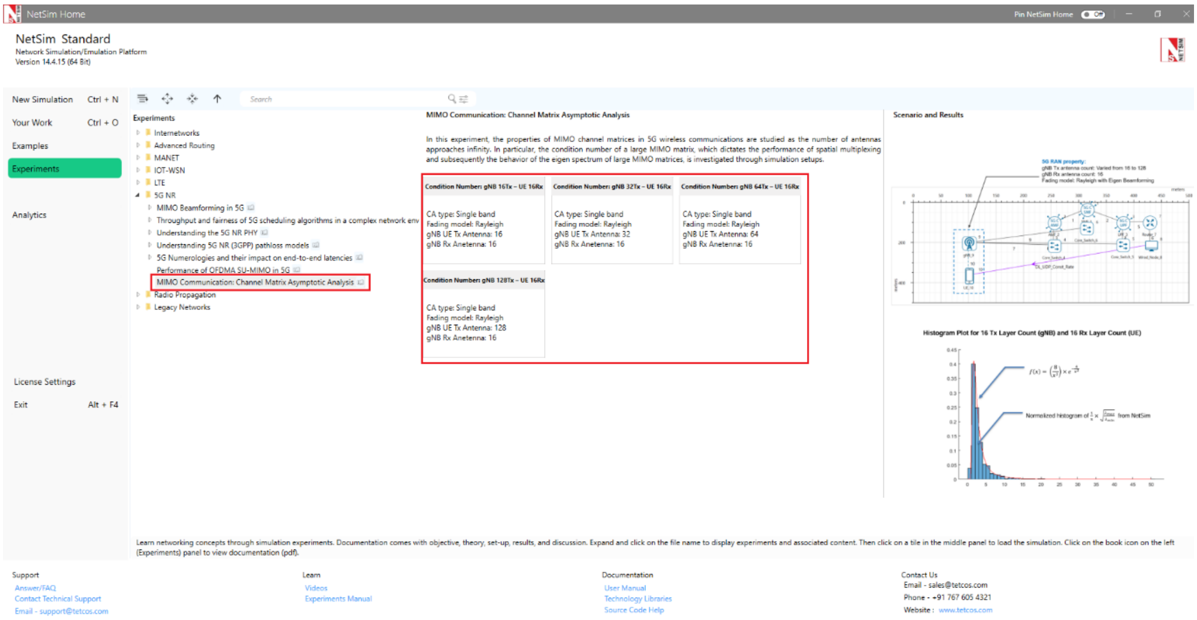


Figure 10-72: List of Experiments

10.7.4 Network Scenario

NetSim UI would display the network topology shown in the screenshot below when you open the example configuration file.

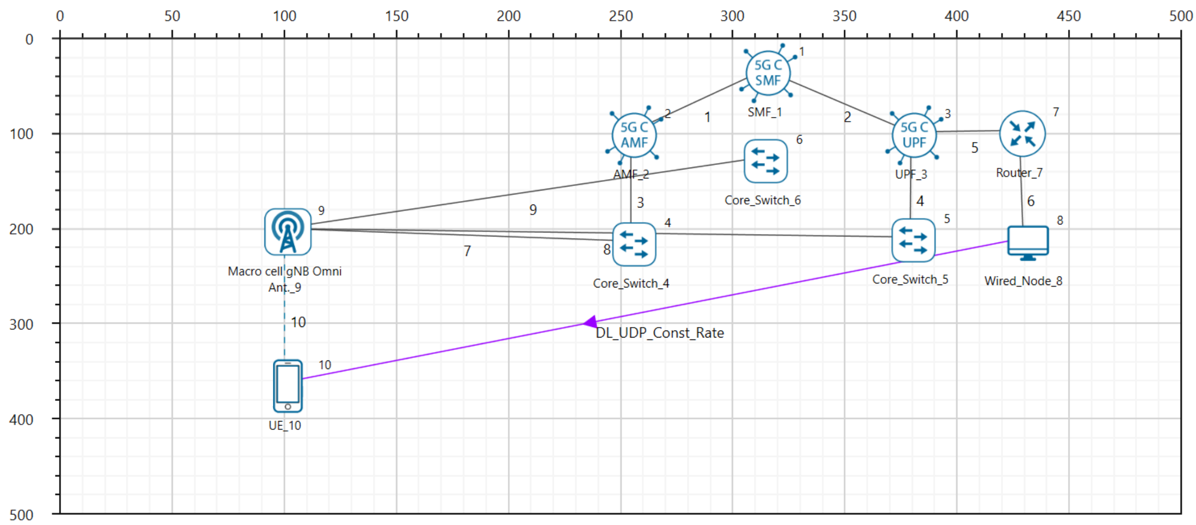


Figure 10-73: Network topology in this experiment

10.7.5 Settings

The following parameters were configured in the network setup:

1. The gNB- Interface 5G_RAN were set with the following properties:

gNB- Interface 5G_RAN Parameters	
gNB Height	10m
Tx Power	40 dBm
Duplex Mode	TDD
CA Type	SINGLE BAND

CA Configuration	n78
DL: UL Ratio	4:1
Numerology	0
Channel Bandwidth (MHz)	10
Tx Antenna Count	Varied from 16 to 128
Rx Antenna Count	16
MCS Table	QAM64LOWSE
CQI Table	TABLE3
Pathloss Model	3GPPTR38.901-7.4.1
Outdoor Scenario	Rural Macro
LOS NLOS Selection	User Defined
LOS Probability	1 (LOS)
Shadow Fading Model	3GPPTR38.901-7.4.1
Fast Fading Model	Rayliegh
Channel Rank/ MIMO layers	Max Rank
MIMO Beamforming Model	Eigen BF
Coherence Time (ms)	10
Additional Loss Model	None

Table 10-49: gNB properties

2. The UE properties were configured with the following parameters:

UE Interface 5G RAN	
Tx Power	23 dBm
UE Height	1.5m
Tx Antenna Count	16
Rx Antenna Count	16

Table 10-50: UE properties

3. A downlink CBR application was configured from wired node to UE with Transport protocol as UDP and Packet Size of 1460 Bytes and Inter Arrival time of 2000 µs and the Start Time was set to 1s[17].

[17] The application end time value of 10,000s is not changed. In NetSim the application runs for $min(AppEndTime,\ SimulationTime)$. Since the simulation is run for 10s, the application runs for only 10s.

4. Click on the plots/log tab in the right panel to enable LTENR Radio measurement as shown below

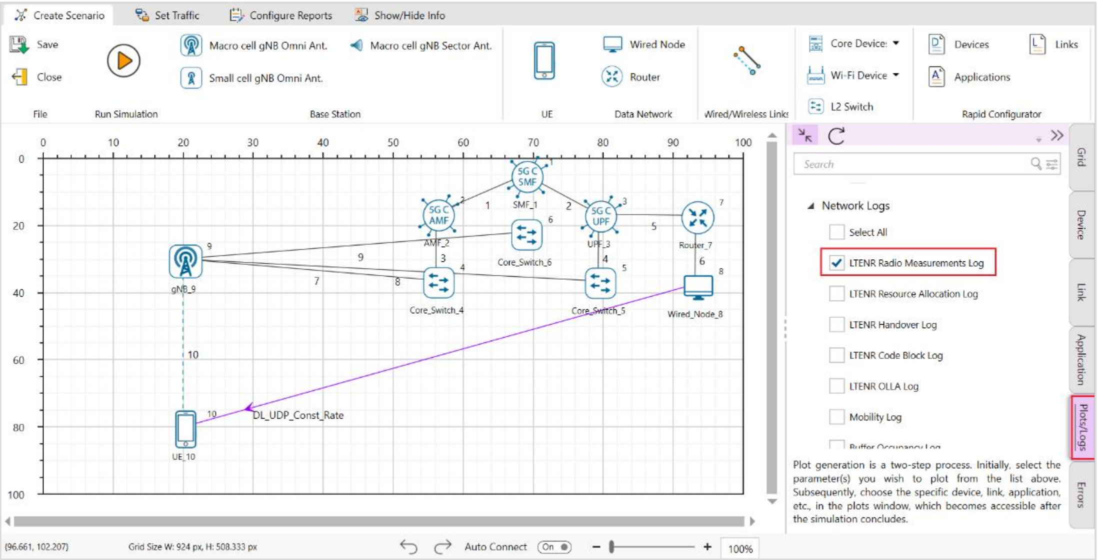


Figure 10-74: Network topology in this experiment

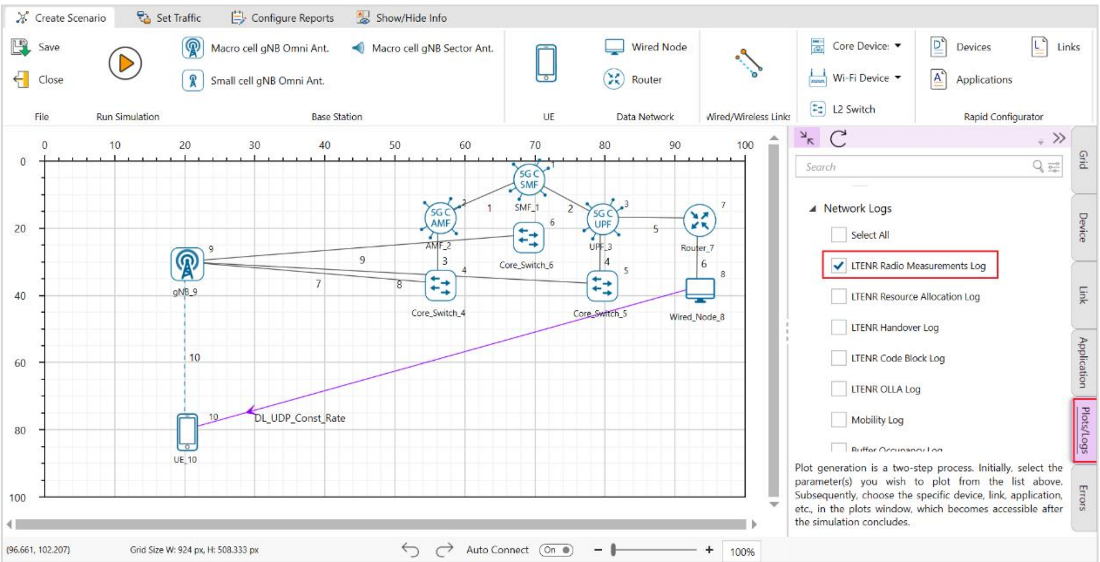


Figure 10-74: Network topology in this experiment

5. Run simulation for 10s. After the simulation, note down the average linear Beamforming Gain (eigen value) obtained for the DL application from the log file generated and then condition number can be obtained.

10.7.6 Part 1: Asymptotic Condition Number Mean

Steps to calculate the condition number through Eigen value (BeamForming Gain Linear):

1. In the results window, click on Log option in the left panel and select LTENR_Radio_Measurements_Log.csv file.

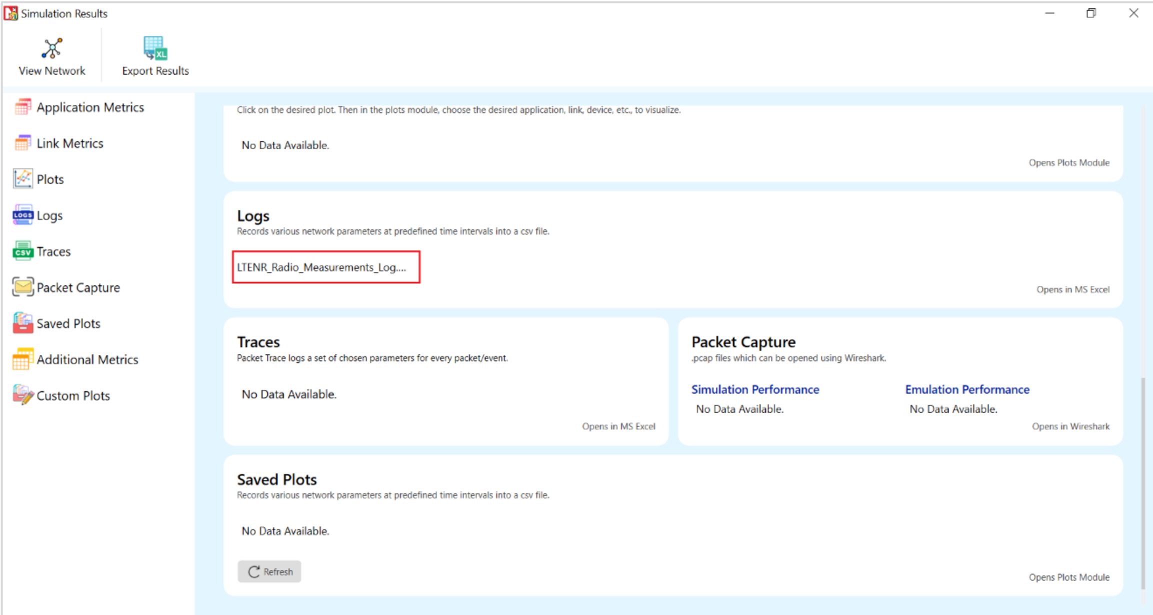


Figure 10-75: NetSim Results window showing access to log file generated

2. This will open a csv file which logs the parameters beamforming gain, over time as shown below.

Time(ms)	gNB or eNB Name	UE Name	Distance(m)	isAssociated	CC_ID	Band	Channel	Rank	Layer ID	Tx_Power(dBm)	LoS State	TotalLoss(dB)	PathLoss(dB)	ShadowFadingLoss(dB)
82	GNB_9	UE_10	162	FALSE	1	n78	SSB	N/A	N/A	40	LOS	83.081927	88.230404	-5.148477
162	GNB_9	UE_10	162	FALSE	1	n78	SSB	N/A	N/A	40	LOS	83.081927	88.230404	-5.148477
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	1	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	2	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	3	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	4	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	5	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	6	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	7	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	8	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	9	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	10	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	11	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	12	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	13	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	14	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	15	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PDSCH	16	16	27.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	1	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	2	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	3	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	4	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	5	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	6	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	7	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	8	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	9	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	10	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	11	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	12	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	13	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	14	10.9588	LOS	89.647997	88.230404	1.417593
162	GNB_9	UE_10	162	TRUE	1	n78	PUSCH	16	15	10.9588	LOS	89.647997	88.230404	1.417593

Figure 10-76: LTENR Radio Measurements log file created after simulation.

3. To change the beamforming gain from dB scale to linear, the following method is used:

$$Eigen\ Value\ (Beam\ Forming\ Gain, Linear) = 10^{\left(\frac{BeamFormingGain_{dB}}{10}\right)}$$

In a new column, enter the following function to calculate the linear Beamforming gain:

$$= POWER(10, [@[BeamFormingGain(dB)]]/10)$$

4. Now goto Insert, select Pivot table option, and then select new sheet option and click ok.

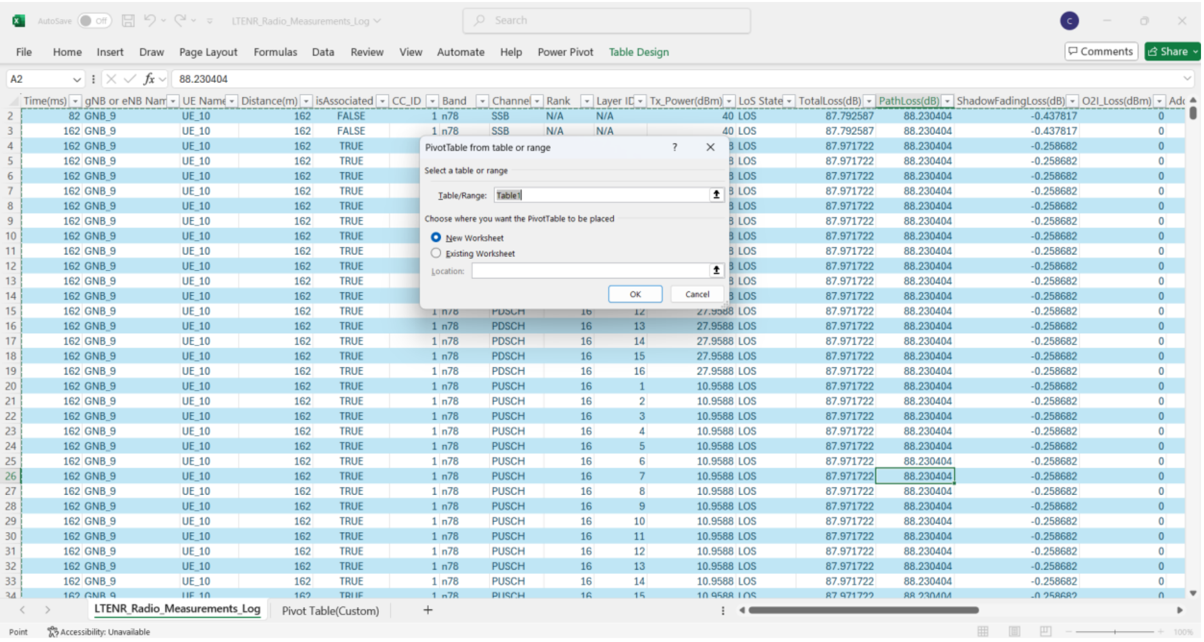


Figure 10-77: Create pivot table

5. Drag and drop the Channel and LAYER_ID field to filter block. Filter the Channel to only PDSCH since we have considered a DL application from server to UE. Similarly, drag and drop the linear beamforming gain to values field and Time to Rows field. Here, the beamforming gain values are set to the maximum by right-clicking on the Value Field Settings and changing the sum to Max.

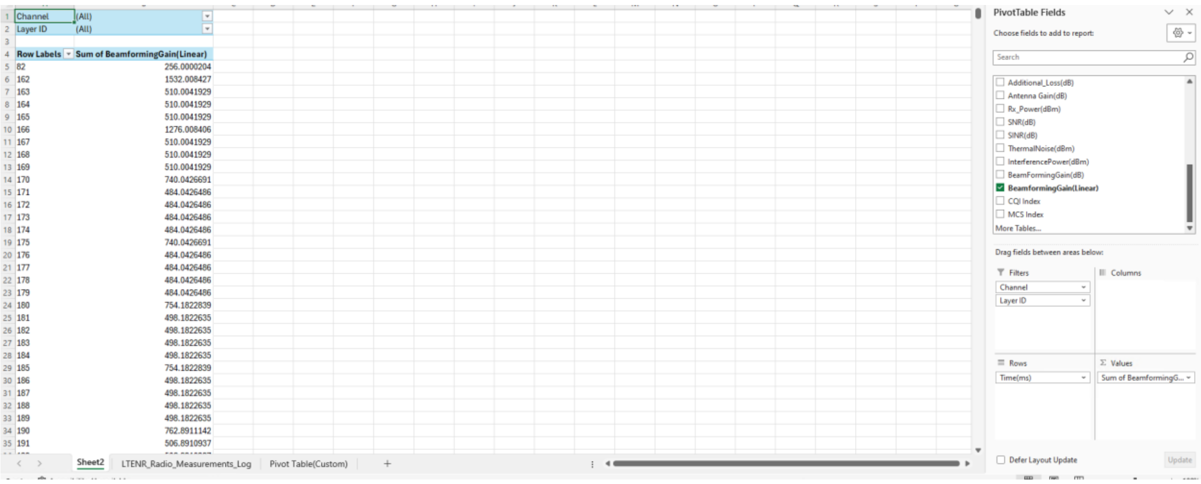


Figure 10-78: LTENR Radio Measurements Log file pivot table showing the filtering process of DL/UL column

6. Now, filter the Layer_Id to layer 1. Now, copy the beamforming values and paste the values in a new sheet with column name as Layer1.

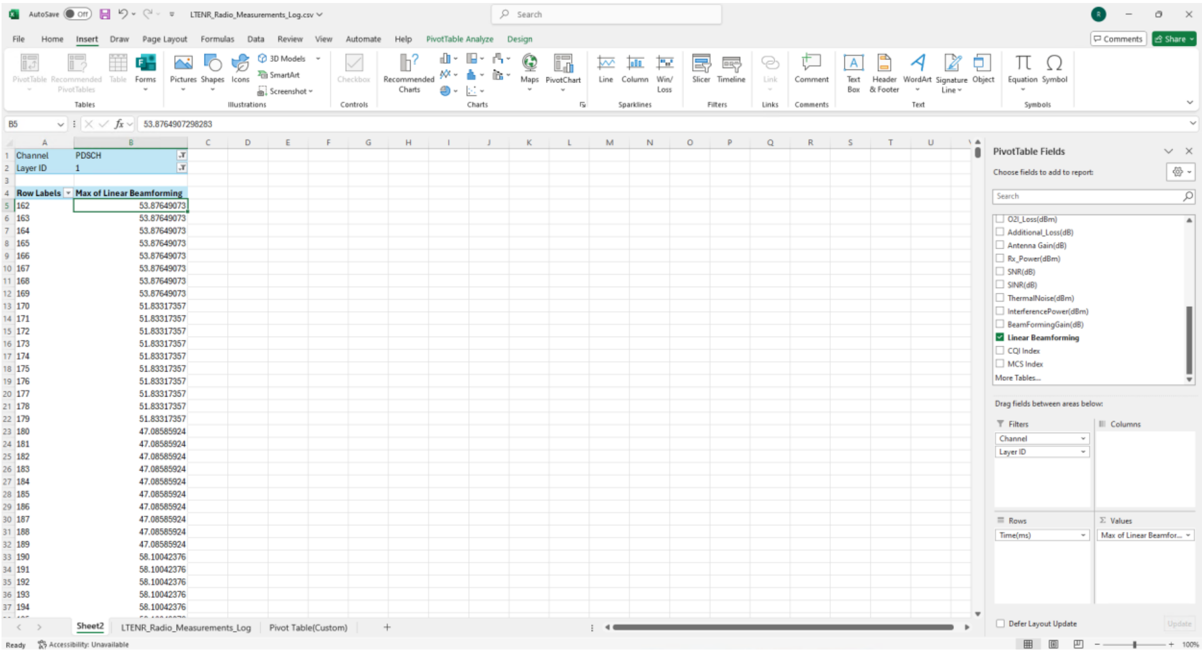


Figure 10-79: LTENR Radio Measurements Log file showing eigen value obtained for layer1.

7. Similarly, filter the LAYER_ID as 16 and copy all the eigen values and paste in new sheet with column name Layer16 as below.

1	Layer1	Layer16
2	49.12557	0.0023089
3	49.12557	0.0023089
4	49.12557	0.0023089
5	49.12557	0.0023089
6	49.12557	0.0023089
7	49.12557	0.0023089
8	49.12557	0.0023089
9	49.12557	0.0023089
10	43.2484	0.2057367
11	43.2484	0.2057367
12	43.2484	0.2057367
13	43.2484	0.2057367
14	43.2484	0.2057367
15	43.2484	0.2057367

Figure 10-80: Layer 1 and layer 16 eigen value in new table

8. Now in the next column, enter the formula, $=[@Layer1]/[@Layer16]$ to calculate EV_max/EV_min . Note that the eigenvalues of Layer 1 are λ_{max} while the eigenvalue of Layer16 is λ_{min}. Rename the column suitably.

1	Layer1	Layer16	EV_Max/EV_Min
2	49.12557	0.0023089	21276.29974
3	49.12557	0.0023089	21276.29974
4	49.12557	0.0023089	21276.29974
5	49.12557	0.0023089	21276.29974
6	49.12557	0.0023089	21276.29974
7	49.12557	0.0023089	21276.29974
8	49.12557	0.0023089	21276.29974
9	49.12557	0.0023089	21276.29974
10	43.2484	0.2057367	210.2123368
11	43.2484	0.2057367	210.2123368
12	43.2484	0.2057367	210.2123368
13	43.2484	0.2057367	210.2123368
14	43.2484	0.2057367	210.2123368
15	43.2484	0.2057367	210.2123368

Figure 10-81: Showing $\sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}})$ obtained

9. Now in the next column, enter the formula, $= SQRT([@[EV\_Max\_EV\_Min]])$. This will calculate square root of the ratio $\sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}}$ which is known as the condition number.

1	Layer1	Layer16	EV_Max/EV_Min	SQRT[EV_Mx/EV_Min]
2	49.12557	0.0023089	21276.29974	145.8639768
3	49.12557	0.0023089	21276.29974	145.8639768
4	49.12557	0.0023089	21276.29974	145.8639768
5	49.12557	0.0023089	21276.29974	145.8639768
6	49.12557	0.0023089	21276.29974	145.8639768
7	49.12557	0.0023089	21276.29974	145.8639768
8	49.12557	0.0023089	21276.29974	145.8639768
9	49.12557	0.0023089	21276.29974	145.8639768
10	43.2484	0.2057367	210.2123368	14.49870121
11	43.2484	0.2057367	210.2123368	14.49870121
12	43.2484	0.2057367	210.2123368	14.49870121
13	43.2484	0.2057367	210.2123368	14.49870121
14	43.2484	0.2057367	210.2123368	14.49870121
15	43.2484	0.2057367	210.2123368	14.49870121

Figure 10-82: Condition Number obtained

10. In a new cell, enter the formula $= AVERAGE(Table2[Condition\_Number])$ to calculate the average of condition number.

	Layer1	Layer16	EV_Max/EV_Min	SQRT[EV_Mx/EV_Min]		Average Condition Number
2	49.12557	0.0023089	21276.29974	145.8639768		46.40390893
3	49.12557	0.0023089	21276.29974	145.8639768		
4	49.12557	0.0023089	21276.29974	145.8639768		
5	49.12557	0.0023089	21276.29974	145.8639768		
6	49.12557	0.0023089	21276.29974	145.8639768		
7	49.12557	0.0023089	21276.29974	145.8639768		
8	49.12557	0.0023089	21276.29974	145.8639768		
9	49.12557	0.0023089	21276.29974	145.8639768		
10	43.2484	0.2057367	210.2123368	14.49870121		
11	43.2484	0.2057367	210.2123368	14.49870121		
12	43.2484	0.2057367	210.2123368	14.49870121		
13	43.2484	0.2057367	210.2123368	14.49870121		
14	43.2484	0.2057367	210.2123368	14.49870121		
15	43.2484	0.2057367	210.2123368	14.49870121		

Figure 10-83: Average eigen value obtained

11. Similarly, enter the formula, $= VAR.P(Table2[Condition\_Number])$ in a new cell, to calculate the variance of condition number.

	Layer1	Layer16	EV_Max/EV_Min	SQRT[EV_Mx/EV_Min]		Average Condition Number
2	49.12557	0.0023089	21276.29974	145.8639768		46.40390893
3	49.12557	0.0023089	21276.29974	145.8639768		
4	49.12557	0.0023089	21276.29974	145.8639768		
5	49.12557	0.0023089	21276.29974	145.8639768		
6	49.12557	0.0023089	21276.29974	145.8639768		Variance Condition Number
7	49.12557	0.0023089	21276.29974	145.8639768		1267.256498
8	49.12557	0.0023089	21276.29974	145.8639768		
9	49.12557	0.0023089	21276.29974	145.8639768		
10	43.2484	0.2057367	210.2123368	14.49870121		
11	43.2484	0.2057367	210.2123368	14.49870121		
12	43.2484	0.2057367	210.2123368	14.49870121		
13	43.2484	0.2057367	210.2123368	14.49870121		
14	43.2484	0.2057367	210.2123368	14.49870121		
15	43.2484	0.2057367	210.2123368	14.49870121		

Figure 10-84: Variance of eigen value obtained

12. Repeat the steps 1 to 10 with varying Tx antenna count in gNB as 32, 64, and 128. Note down the mean and variance of condition number.

10.7.7 Results

Tx, Rx	Condition Number Mean	Condition Number Variance
16, 16	46.40	1267.25
32, 16	4.40	0.230519
64, 16	2.54	0.0227125
128,16	1.88	0.0052224

Table 10-51: Showing Mean and Variance of Condition Number with varying Tx

a) Case1: gNB Tx = 32, UE Rx = 16

From theory $K = \frac{\left(1 + \sqrt{\frac{16}{32}}\right)}{1 - \sqrt{\frac{16}{32}}} = 5.82$. NetSim result = 4.419.

$$\text{Difference} = \frac{5.82 - 4.419}{4.419} = 31.70\ \%$$

b) Case2: gNB Tx = 64, UE Rx = 16

From theory $K = \frac{\left(1+\sqrt{\frac{16}{64}}\right)}{1-\sqrt{\frac{16}{64}}} = 3.0$. NetSim result = 2.552.

$$\text{Difference} = \frac{3 - 2.552}{2.552} = 17.55\ \%$$

c) Case3: gNB Tx = 128, UE Rx = 16

From theory $K = \frac{\left(1+\sqrt{\frac{16}{128}}\right)}{1-\sqrt{\frac{16}{128}}} = 2.09$. NetSim result = 1.883

$$\text{Difference} = \frac{2.09 - 1.883}{1.883} = 10.99\ \%$$

Since theory is for the asymptotic mean, we will not get an exact match. The results show the trend that as N increases simulation outputs approach theoretical predictions

10.7.8 Part 2: Asymptotic Condition Number Distribution

10.7.9 Theory

Consider a i.i.d $N_r \times N_t$ complex Gaussian random matrix $\mathbf{H}$. Define the Wishart matrix $\boldsymbol{W} = \boldsymbol{H} \times \boldsymbol{H}^{\dagger}$ with parameters $n = \min(N_t, N_r)$ and $N = \max(N_t, N_r)$, and eigenvalues as $\lambda_{max} = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{min} \geq 0$.

The condition number of $\mathbf{H}$ is defined as

$$K(\boldsymbol{H}) = \sqrt{\frac{\lambda_{max}}{\lambda_{min}}}$$

From [1], if $n = N$, and $N \to \infty$ then $\frac{K(\boldsymbol{H})}{n}$ converges in distribution to a random variable whose PDF is given by

$$f(x) = \left(\frac{8}{x^3}\right) \times e^{-\left(\frac{4}{x^2}\right)}$$

We simulate the case $N_r = N_t = 16$, since the antenna count is limited to 16, in the UEs in NetSim. Figure 10-85 is a comparison of the normalized histogram of $\frac{K}{N}$ from NetSim vs. the asymptotic pdf equation. At $N_r = N_t = 16$ itself, there seems to be a reasonable fit.

10.7.9.1 Comparison of NetSim Results with asymptotic function

Steps to plot histogram of condition number:

1. Calculate the Condition Number using the LTENR Radio Measurements Log.
2. In a new column, divide the Condition_Number by 16 using the following excel function:

$$= ([@Condition\_Number]/16)$$

3. Click on Insert-> Pivot Table, drag and drop Column1 to Rows field.
4. Copy the values in Row Labels column.
5. In MATLAB create a new file, create an array **Condition_Number_array**, paste the values to it as

```
Condition_Number_array = [c1
                          c2
                          c3
                          ….
                          cn];
```

6. Now use below MATLB code to plot the normalized histogram plot with the function

$$f(x) = \left(\frac{8}{x^3}\right) \times e^{-\frac{4}{x^2}}$$

```
hold on;

c = histogram(Condition_Number_array,'Normalization','probability');

x = 0:0.1:50;  %x varies from 0 to 50 in steps of 0.2.

y = (8./x.^3).*(exp(-4./x.^2));

plot(x,y,'r');

hold off;


% For CDF plot use below MATLB code

cdfplot(Condition_Number_array);
```

Program 1: MATLAB code for plotting condition number from NetSim and comparing against the asymptotic PDF from analysis.

10.7.9.2 Histogram Plot for 16 Tx Layer Count (gNB) and 16 Rx Layer Count (UE)

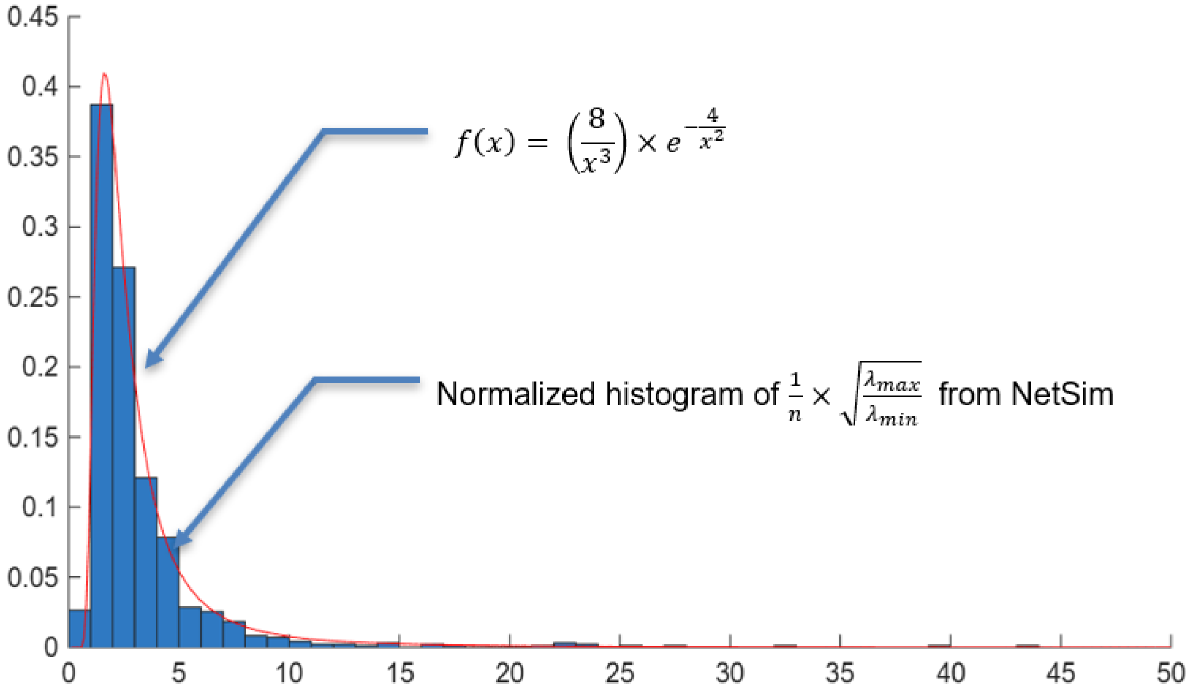


Figure 10-85: The normalized histogram $\frac{1}{n} \times \sqrt{\frac{\lambda_{max}}{\lambda_{min}}}$ for $N_t = N_r = 16$ itself fits well with the asymptotic distribution of $\frac{K(H)}{n}$

10.7.9.3 Part 3: Marchenko-Pasteur Distribution

10.7.9.4 Theory

The Marchenko Pasteur distribution for $\frac{N_r}{N_t} = y$ with $N_t \to \infty$ is

$$f(x) = \frac{1}{2\pi x y} \times \sqrt{(b-x)(x-a)}$$

where $b = \left(1+\sqrt{y}\right)^2$ and $a = \left(1-\sqrt{y}\right)^2$.

Let us consider the case where, the number of transmit antennas N_t and the number of receive antennas N_r, are related as $\frac{N_r}{N_t} = \frac{1}{8} = y$. Substituting for y we get the MP distribution as

$$f(x) = \begin{cases} \frac{4}{\pi x}\sqrt{\left(\frac{1}{2}\right) - \left(x - \frac{9}{8}\right)^2}, & \left(1 - \frac{1}{2\sqrt{2}}\right)^2 \le x \le \left(1 + \frac{1}{2\sqrt{2}}\right)^2 \\ 0, & \text{All other } x \end{cases}$$

Comparison of NetSim Results with Marchenko- Pasteur function

10.7.9.5 Steps to plot the histogram

1. Create a scenario with Tx antenna count as 128 and Rx antenna count as 16.
2. Now open the file LTENR_Radio_Measurements_Log.csv file.
3. Filter the PDSCH/PUSCH to PDSCH.

4. Now, in LTENR Radio Measurements Log file compute Eigen values (Linear Beamforming Gain) for all layers, Layer Id 1 to Layer Id 16.
5. Select the eigen values in Beamforming Gain column and copy all the values.
6. Create a new file in MATLAB. Create an array Eigen_value_array and paste the copied values from the 5G parameter log csv file as shown.

   Eigen_value_array = [ev1
                   ev2
                   …
                   evN];
7. Use below MATLAB code to plot the MP distribution function along with the normalized histogram.

   For the MP distribution function, the x varies from 0.418 to 1.832 in steps of 0.001.

```
x = 0.418:0.001:1.832;
y = (1.27324./x).*sqrt(0.5 - (x-9/8).^2);
hold on;
plot(x,y);
histogram(Eigen_value_array /128,'Normalization','pdf');
hold off;
```

Program 2: MATLAB code for plotting the MP distribution for $y = \frac{1}{8}$ and pooled eigenvalues histogram, from NetSim simulation results

10.7.9.6 Results

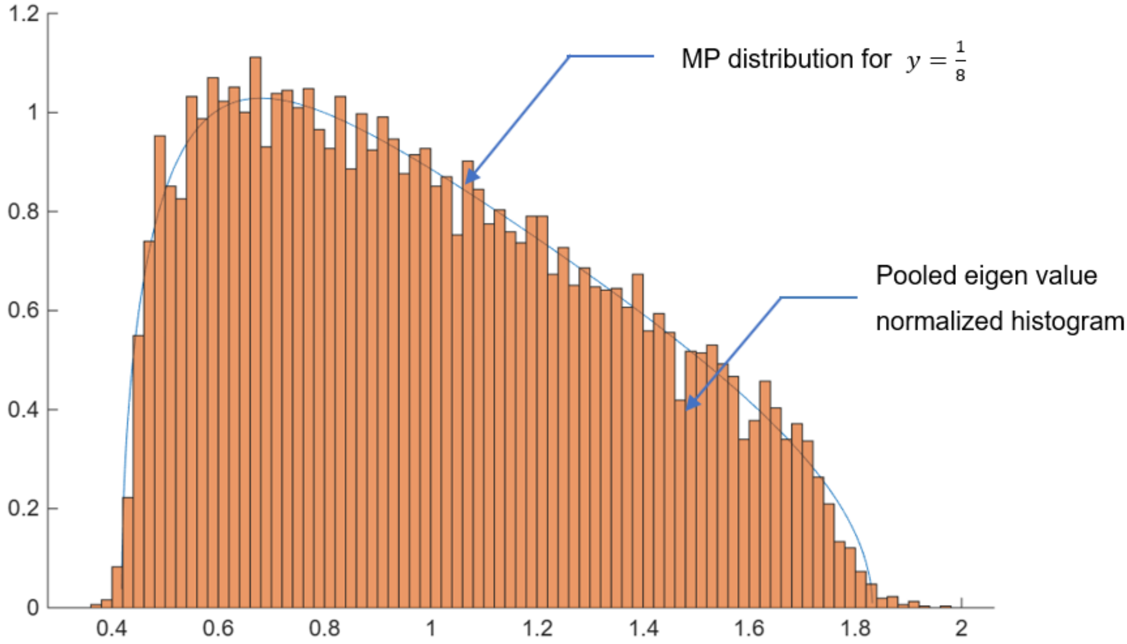


Figure 10-86: NetSim Results vs. Marchenko-Pasteur distribution for N_r=16 and N_t=128

10.7.9.7 References

1. Edelman, A. (1988). Eigenvalues and Condition Numbers of Random Matrices. SIAM Journal on Matrix Analysis and Applications, 543-560.