# NetSim®

Accelerate Network R & D

# Internetworks

A Network Simulation & Emulation Software

By

TETCOS LLP

The information contained in this document represents the current view of TETCOS LLP on the issues discussed as of the date of publication. Because TETCOS LLP must respond to changing market conditions, it should not be interpreted to be a commitment on the part of TETCOS LLP, and TETCOS LLP cannot guarantee the accuracy of any information presented after the date of publication.

This manual is for informational purposes only.

The publisher has taken care of the preparation of this document but makes no expressed or implied warranty of any kind and assumes no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information contained herein.

**Warning! DO NOT COPY**

Copyright in the whole and every part of this manual belongs to TETCOS LLP and may not be used, sold, transferred, copied or reproduced in whole or in part in any manner or in any media to any person, without the prior written consent of TETCOS LLP. If you use this manual, you do so at your own risk and on the understanding that TETCOS LLP shall not be liable for any loss or damage of any kind.

TETCOS LLP may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from TETCOS LLP, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Rev 15.1 (V), Oct 2026, TETCOS LLP. All rights reserved.

All trademarks are the property of their respective owners.

---

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

# Contents


**1 Introduction** **6**

**2 Simulation GUI** **7**

2.1 Create Scenario . . . 7

2.2 Devices specific to NetSim Internetworks Library . . . 7

2.2.1 Click and drop into environment . . . 8

2.2.2 Link Properties . . . 10

2.3 Enable Packet Trace, Event Trace & Plots (Optional) . . . 11

2.4 Enable protocol specific logs and plots . . . 11

2.5 Run Simulation . . . 12

**3 Model Features** **13**

3.1 WLAN 802.11 . . . 13

3.1.1 WLAN standards supported in NetSim . . . 14

3.1.2 The 2.4 GHz Channels . . . 14

3.1.3 The 5 GHz Channels . . . 15

3.1.4 The 5.9 GHz Channels . . . 15

3.1.5 Channel Numbering . . . 16

3.1.6 WLAN PHY Rate in NetSim . . . 17

3.1.7 SIFS, Slot Time, CW Min, and CW Max settings . . . 17

3.1.8 PHY Implementation . . . 18

3.1.9 PHY States . . . 19

3.1.10 802.11 implementation details . . . 19

3.1.11 802.11 ac/ax MAC and PHY Layer Implementation . . . 23

3.1.12 MAC Aggregation in NetSim . . . 25

3.1.13 Signal to interference and noise ratio (SINR) . . . 27

3.1.14 Error Model . . . 28

3.1.15 Transmit Power . . . 28

3.1.16 Carrier Sense . . . 28

3.1.17 Transmission Range, Carrier Sense Range, and Interference Range . . . 28

3.1.18 Carrier Sense (CS) Threshold . . . 29

3.1.19 Transmitter's choice of MCS . . . 29

3.1.20 Hidden Node Behavior . . . 29

3.1.21 IEEE 802.11 e QoS and EDCA . . . 30

3.1.22 Rate Adaptation . . . 32

3.1.23 Model Limitations . . . 32

3.1.24 Wi-Fi GUI Parameters . . . 33

3.1.25 IEEE 802.11 Results . . . 38

3.1.26 Radio measurements log file . . . 39

3.1.27 NetSim plots . . . 41

3.1.28 IEEE 802.11 Backoff Log . . . 41

3.2 Layer 2 (L2) Ethernet Switching . . . 43

3.2.1 Spanning Tree Protocol . . . 43

3.2.2 Switch Port States . . . 44

3.2.3 Model Limitations . . . 44

3.2.4 Switch: GUI Parameters . . . 44

3.3 Open Shortest Path First (OSPF v2) Routing Protocol . . . 46

3.3.1 OSPF Overview . . . 46

3.3.2 OSPF Features . . . 47
3.3.3 Excluded Features . . . 48
3.3.4 OSPF: GUI Parameters . . . 48
3.4 Transmission Control Protocol (TCP) . . . 51
3.4.1 TCP overview . . . 51
3.4.2 TCP Features . . . 51
3.4.3 Congestion Control Algorithms in TCP . . . 51
3.4.4 Limitations of TCP . . . 52
3.4.5 TCP: GUI parameters . . . 52
3.4.6 TCP Performance Metrics . . . 54
3.4.7 TCP Reference Documents . . . 55
3.5 User Datagram Protocol (UDP) . . . 55
3.5.1 UDP Overview . . . 55
3.5.2 UDP: GUI parameters . . . 56
3.5.3 UDP Performance Metrics . . . 58
3.5.4 UDP Reference Documents . . . 58
3.6 IP Protocol . . . 58
3.6.1 IP Performance Metrics . . . 58
3.7 Buffering, Queueing and Scheduling . . . 59
3.7.1 Buffers . . . 59
3.7.2 Queuing . . . 59
3.7.3 Scheduling . . . 60
3.8 Links . . . 61
3.8.1 Modeling Error in Wired Links . . . 61
3.9 IP Addressing in NetSim . . . 61
**4 Featured Examples** **61**
4.1 802.11n MIMO . . . 62
4.2 Effect of Bandwidth in Wi-Fi 802.11ac . . . 64
4.2.1 Effect of Bandwidth . . . 64
4.3 Factors affecting WLAN PHY Rate . . . 68
4.3.1 Effect of AP-STA Distance on throughput . . . 68
4.3.2 Effect of Pathloss Exponent . . . 71
4.3.3 Effect of Transmitter power . . . 73
4.4 Peak UDP and TCP throughput 802.11ac and 802.11n . . . 76
4.4.1 IEEE802.11n . . . 77
4.4.2 IEEE802.11ac . . . 79
4.5 MAC Throughput and Efficiency comparison of 802.11 Legacy vs. HT protocols . 81
4.5.1 Network Scenario . . . 82
4.5.2 Part I: Legacy . . . 82
4.5.3 Part II: High Throughput (HT) . . . 84
4.6 Configuring IP addresses, subnets and applying firewall rules based on subnets using Class B IP addresses . . . 89
4.6.1 IP Addressing . . . 89
4.6.2 IP address classes . . . 90
4.6.3 Configuring Class-B address in NetSim . . . 90
4.6.4 Subnetting . . . 91
4.6.5 Configuring Class-B subnetting . . . 91
4.6.6 Firewall rules based on subnets . . . 93

4.7 Different OSPF Control Packets . . . 95
4.8 Configuring Static Routing in NetSim . . . 98
4.8.1 Without Static Route . . . 100
4.8.2 With Static Route . . . 100
4.9 TCP Window Scaling . . . 102
4.10 An enterprise network comprising different subnets and running various applications 107
4.10.1 Network Settings . . . 108
4.10.2 Results and Observations . . . 110
4.11 Design of a nationwide university network to analyze capacity, latency, and link failures . . . 111
4.11.1 Introduction . . . 111
4.11.2 Network Settings . . . 113
4.11.3 Simulation cases and Application settings . . . 114
4.11.4 Results and Observations . . . 115
**5 Internetworks Experiments in NetSim 121**
**6 Reference Documents 122**
**7 Latest FAQs 123**

# 1 Introduction

The Internetworks library in NetSim supports various protocols across all the layers of the TCP/IP network stack. These include Ethernet, Address Resolution Protocol (ARP), Wireless LAN – 802.11 a / b / g / n / ac / ax / p and e (EDCA), Internet Protocol (IP), Transmission Control Protocol (TCP), Virtual LAN (VLAN), User Datagram Protocol (UDP), and routing protocols such as Routing Information Protocol (RIP), Open Shortest Path First (OSPF).

An internetwork is generally a collection of two or more networks (typically LANs and WLANs) which are interconnected to form a larger network. All networks in an Internetwork have a unique network address. Routers interconnect different networks.

Users can use the following devices to design Internetworks: wireless node, wired node, switch, router, and access point (AP). Wired nodes (a term for computers, servers, etc.) connect via wired link to switches or routers, and wireless nodes connect via wireless links to Access Points (APs). Multiple links terminate at a switch/router, which enables connectivity between them. Many switches/routers are present in an internetwork to connect all the end-nodes. The end-nodes provide and consume useful information via applications like data, voice, video, etc.

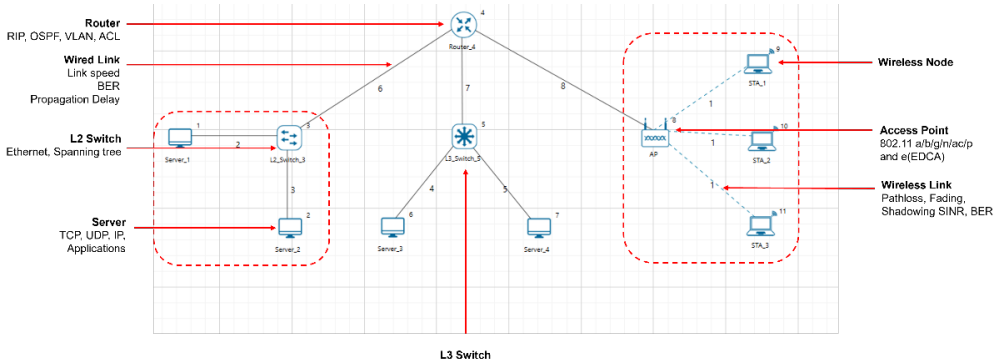



**Figure 1-1:** *A typical Internetworks scenario in NetSim.*

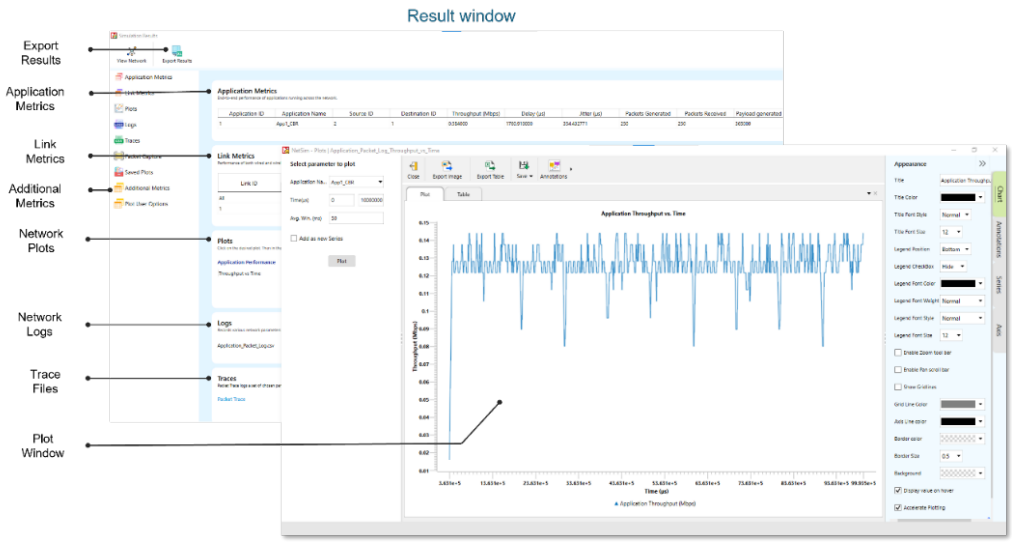



**Figure 1-2:** *The Result dashboard and the Plots window shown in NetSim after completion of a simulation.*

# 2 Simulation GUI

Open NetSim and click New Simulation ▹ Internetworks as shown in Figure 2-1.

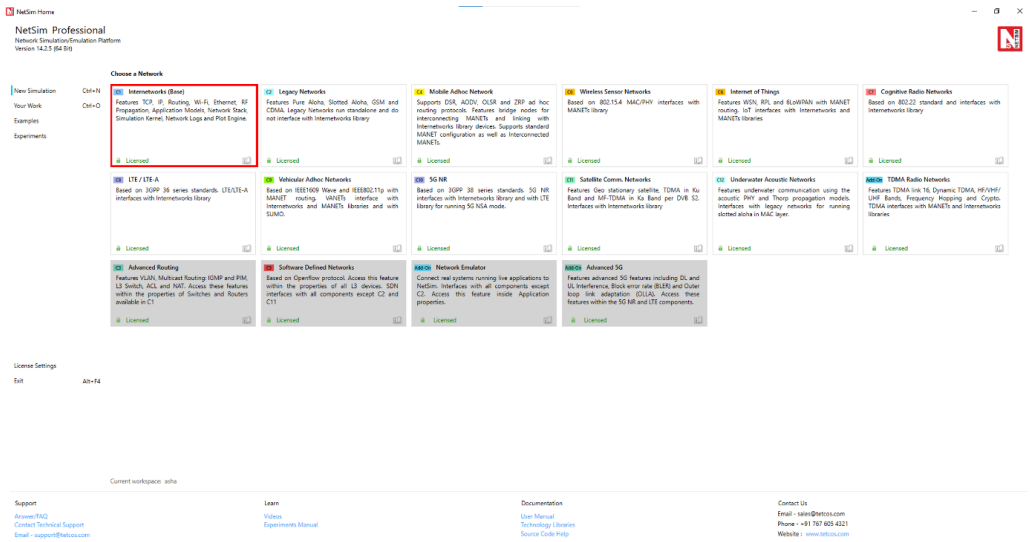



**Figure 2-1:** *NetSim Home Screen.*

## 2.1 Create Scenario

Internetworks comes with a palette of various devices like L2 Switch, L3 Switch, Router, Wired Node, Wireless Node, and AP (Access Point).

## 2.2 Devices specific to NetSim Internetworks Library

- **Wired node:** A Wired node can be an end-node or a server. It is a 5-layer device that can be connected to a switch or router. It supports only 1 Ethernet interface and has its own IP and MAC Addresses.
- **Wireless Nodes:** A Wireless node can be an end-node or a server. It is a 5-layer wireless device that can be connected to an Access Point. It supports only 1 Wireless interface and has its own IP and MAC Addresses.
- **L2 Switch:** A switch is a layer-2 device that uses the device's MAC address to make forwarding decisions. It does not have an IP address.
- **L3 Switch:** A Layer 3 switch operates at both the data link layer and the network layer. It combines the functionality of a traditional switch and a router.
- **Router:** A router is a layer-3 device and supports a maximum of 24 interfaces each of which has its own IP address.
- **Access Point:** Access Point (AP) is a layer-2 wireless device working per 802.11 Wi-Fi protocol. It can be connected to wireless nodes via wireless links and to a router or a switch via a wired link.

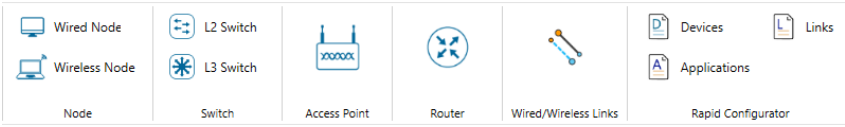



**Figure 2-2:** *Internetworks Device Palette in GUI.*

### 2.2.1 Click and drop into environment

- **Add a Wired Node or Wireless Node:** In the toolbar, click the Node ▷ Wired Node icon (or) Node ▷ Wireless Node icon, and place the device in the grid.
- **Note:** Wireless nodes can be effortlessly connected using the auto-connect feature, ensuring that users first drop the access point before adding the wireless node.
- **Add a Router:** In the toolbar, click on the Router icon and place the Router in the grid.
- **Add an L2 Switch or L3 Switch:** In the toolbar, click on Switch ▷ L2 Switch icon (or) Switch ▷ L3 Switch icon and place the device in the grid.
- **Add an Access Point:** In the toolbar, click on the Access Point icon and place the access point in the grid.
- **Connect the devices** by using Wired/Wireless Links present in the top ribbon/toolbar. Click on the first device and then click on the second device. A link will be formed between the two devices.
- **Note:** Wireless devices get auto-connected, whereas wired devices need physical connection.
- Configure an application by following these steps:
  - (i) Click on the Set Traffic tab in the top ribbon/toolbar.
  - (ii) Select any application from the list and configure the traffic between source and destination.
  - (iii) Specify other application parameters per your model.

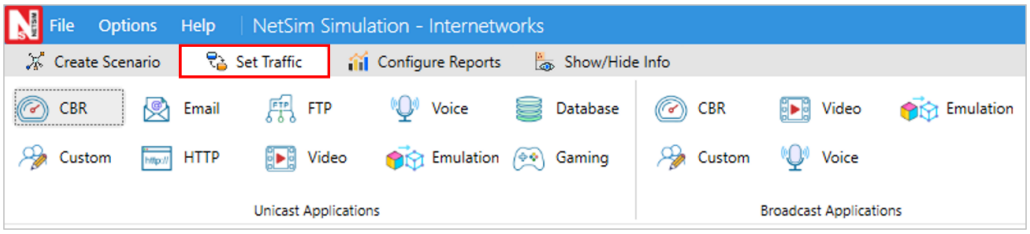



**Figure 2-3:** *Top Ribbon/Toolbar.*

- Repeat (ii) to generate multiple applications. Detailed information on application properties is available in section 6 of NetSim User Manual.
- Clicking on any device (Router, Access Point, L2 Switch, Wireless Node, Wired Node, etc.) will open a right-side property panel, allowing users to set the parameters.

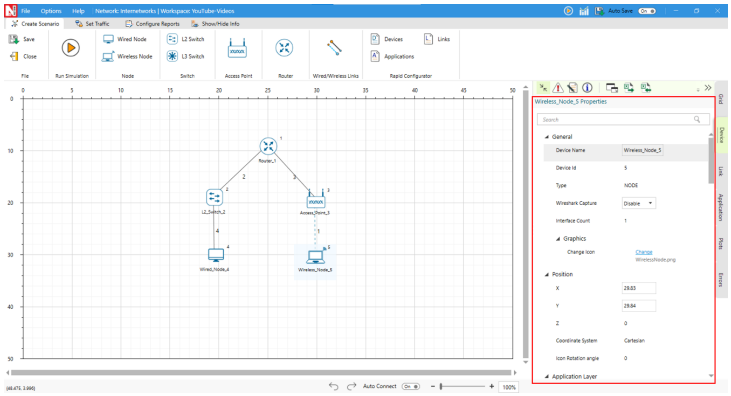


**Figure 2-4:** *Device Properties.*

- In the Wireless Interface, Physical Layer and Data Link Layer parameters are local, but in the Physical Layer, the Standard parameter is global. To set the same parameter value across all devices, ensure that you update the parameter values manually in all other devices (Access Point or Wireless Node) as the parameter change does not propagate automatically due to its local nature.

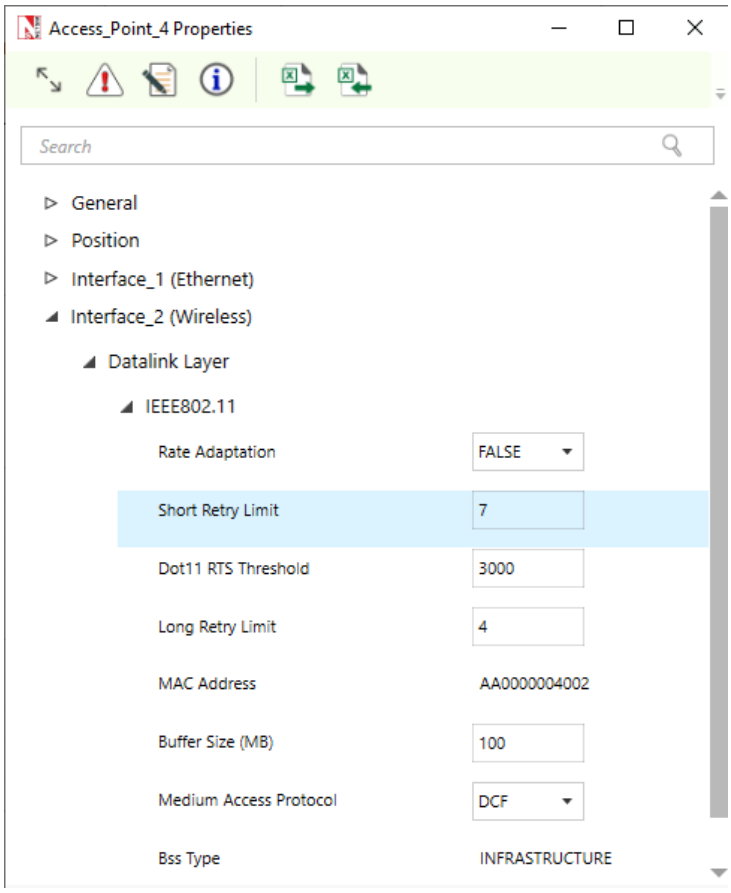



**Figure 2-5:** *MAC properties of Access Point.*

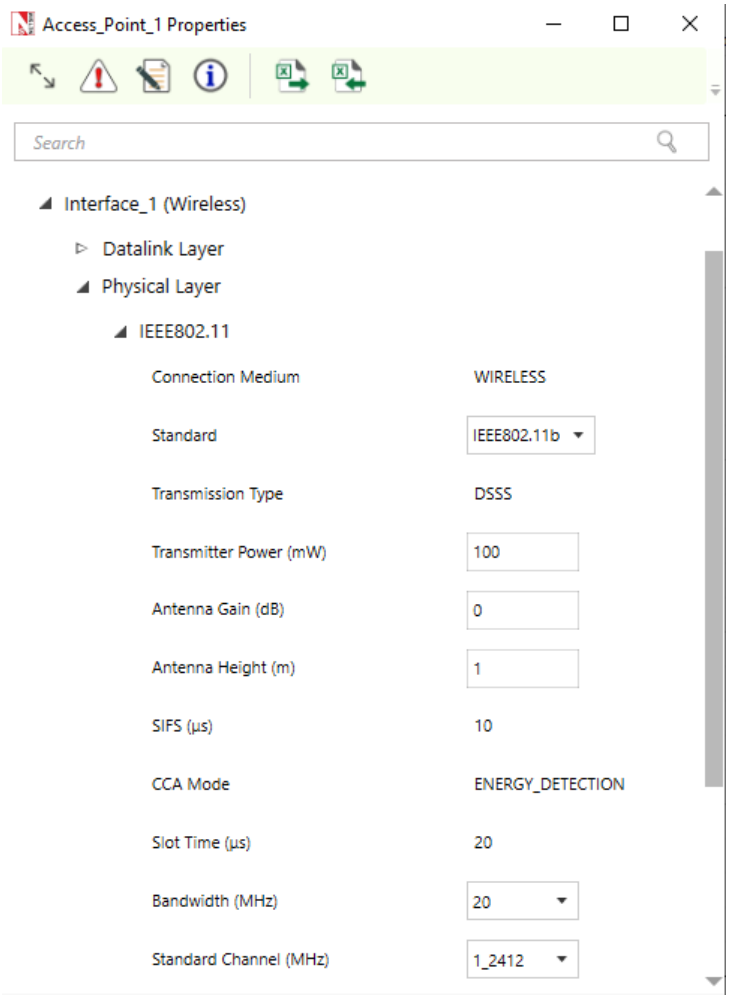



**Figure 2-6:** *PHY Layer properties of Access Point.*

### 2.2.2 Link Properties

- Clicking on the link will open a right-side property panel where users can set the link properties. Note that when simulating Internetworks, if the link propagation delay is set too high, then the applications may not see any throughput since it would take too long for OSPF to converge, and furthermore, TCP may also timeout (since max RTO is 3s).

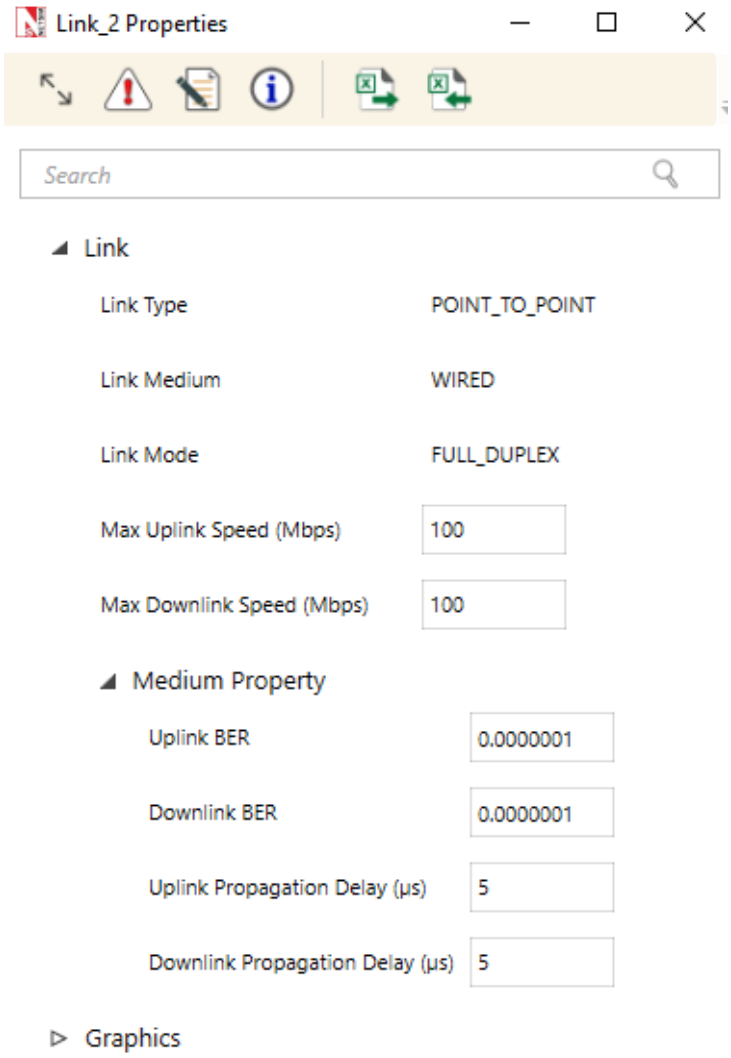



**Figure 2-7:** *Link Properties.*

## 2.3 Enable Packet Trace, Event Trace & Plots (Optional)

Click the Packet Trace / Event Trace icon in the Configure Reports option and check the Packet Trace / Event Trace check box. For detailed help about the packet and event trace, please refer to sections 8.4 and 8.5 in the User Manual.

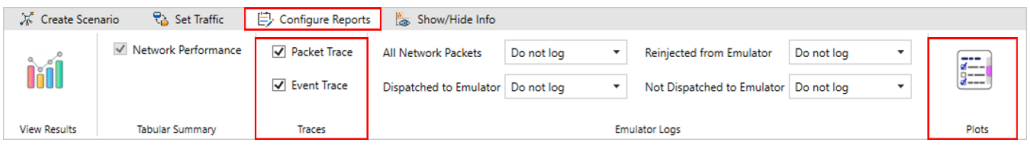



**Figure 2-8:** *Packet Trace, Event Trace & Plots options on top ribbon.*

## 2.4 Enable protocol specific logs and plots

NetSim provides protocol-specific logs for Internetworks libraries, which users can enable before running a simulation. These can be enabled by navigating to the Configure Reports section in the top ribbon, selecting Plots, choosing the desired options, and running the simulation.

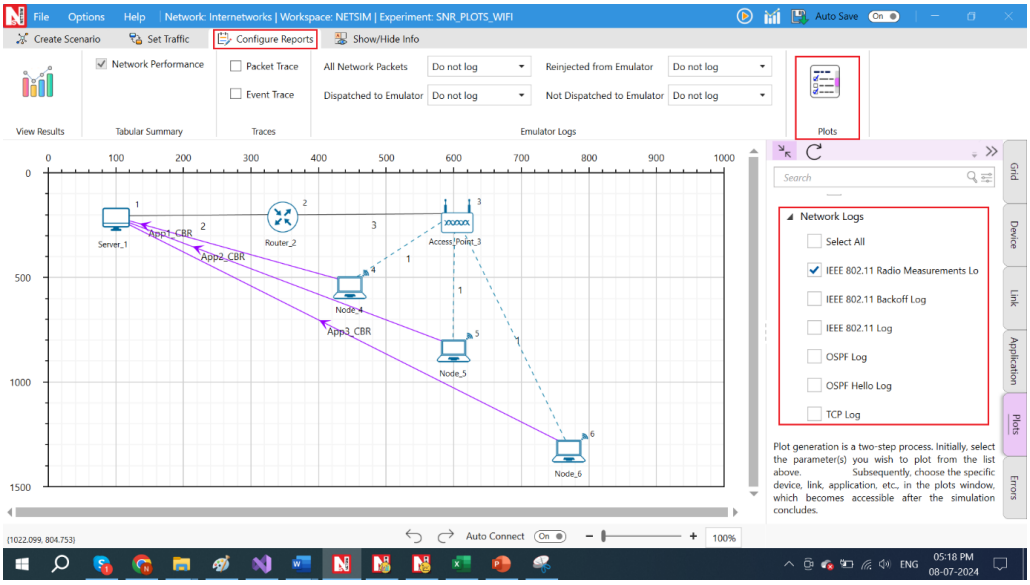



**Figure 2-9:** *Enabling the Network logs in Internetworks.*

Similarly, users can enable the plots for Wi-Fi radio measurements.

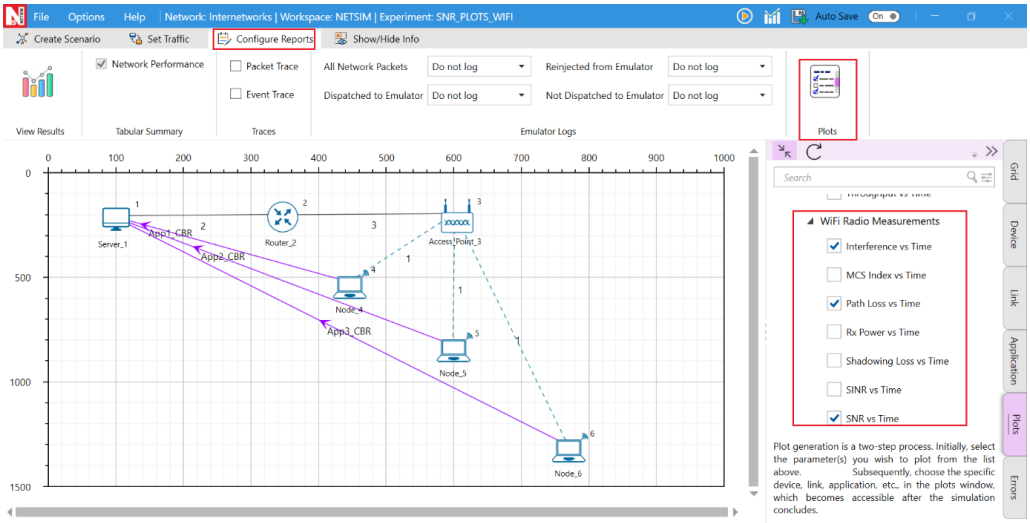



**Figure 2-10:** *Enabling the Wi-Fi Radio measurements plots in Internetworks.*

## 2.5 Run Simulation

Click on the Run Simulation icon on the top ribbon/toolbar. For detailed help, please refer to section 3.5 of the User Manual.

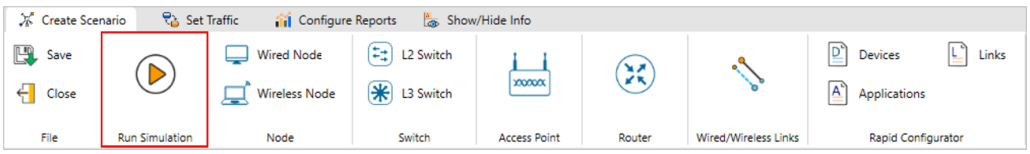



**Figure 2-11:** *Run Simulation on top ribbon.*

Set the Simulation Time and click on the Run button.

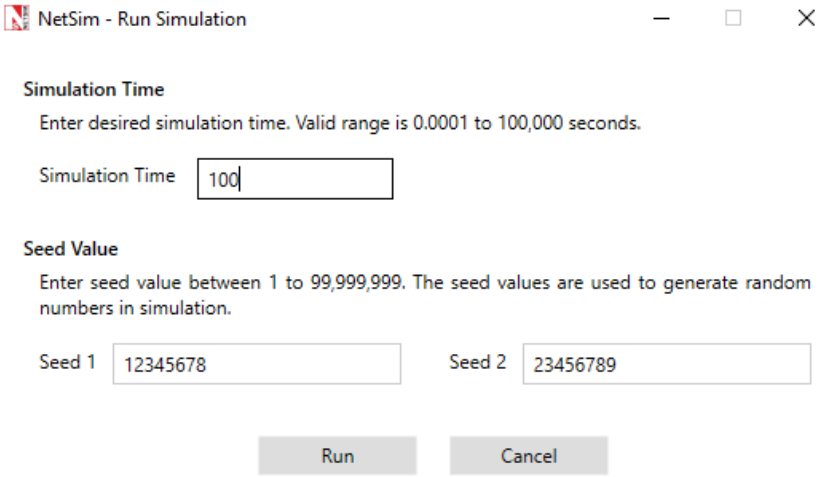



**Figure 2-12:** *Run Simulation window.*

# 3 Model Features

## 3.1 WLAN 802.11

NetSim implements the 802.11 MAC and the 802.11 PHY abstracted at a packet-level. We start with the 3 types of nodes supported in 802.11 Wi-Fi.

- **Wireless Nodes** (Internetworks) or STAs. In Internetworks, APs and Wireless nodes (STAs) are associated based on the connecting wireless link.
- **Wi-Fi Access Points** (Internetworks) or APs. Every STA in the WLAN associates with exactly one AP. Each AP, along with its associated STAs, defines a cell. Each cell operates on a specific channel.
- **Standalone Wireless nodes** (Mobile Adhoc networks).

The MAC Layer features:

- RTS/CTS/DATA/ACK frame transmissions.
- Packet queuing, aggregation, transmission, and retransmission.
- 802.11 EDCA.

The PHY layer implements:

- RF propagation (documented separately).
- Received power based on the propagation model.
- Interference and signal to interference-plus-noise (SINR) calculation.
- MCS (and in turn PHY Rate) setting based on RSS and rate adaptation algorithms.
- BER calculation and packet error modelling.

### 3.1.1 WLAN standards supported in NetSim

802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, 802.11ax, 802.11e (EDCA) and 802.11p are the WLAN standards available in NetSim. The operating frequencies and bandwidths are given in Table 3-1.

**Table 3-1:** *WLAN standards supported in NetSim.*

| WLAN standard | Frequency (GHz) | Bandwidth (MHz) |
|---|---|---|
| 802.11 a | 5 | 20 |
| 802.11 b | 2.4 | 20 |
| 802.11 g | 2.4 | 20 |
| 802.11 p | 5.9 | 5, 10, 20 |
| 802.11 n | 2.4, 5 | 20, 40 |
| 802.11 ac | 5 | 20, 40, 80, 160 |
| 802.11ax | 2.4 | 20, 40 |
| 802.11ax | 5 | 20, 40, 80, 160 |

802.11 p and WAVE are described in the VANET Technology library documentation.

### 3.1.2 The 2.4 GHz Channels

The following channel numbers are well-defined for the 2.4GHz standards:

**Table 3-2:** *2.4 GHz Wi-Fi Channels per IEEE Std 802.11g-2003, 802.11-2012 and 802.11-2024.*

| Channel Number | Center Frequency (MHz) |
|---|---|
| 1 | 2412 |
| 2 | 2417 |
| 3 | 2422 |
| 4 | 2427 |
| 5 | 2432 |
| 6 | 2437 |
| 7 | 2442 |
| 8 | 2447 |
| 9 | 2452 |
| 10 | 2457 |
| 11 | 2462 |
| 12 | 2467 |
| 13 | 2472 |
| 14 | 2484 |

Channels 1 through 14 are used in 802.11b, while channels 1 through 13 are used in 802.11g, 802.11n and 802.11ax.

### 3.1.3 The 5 GHz Channels

The following channel numbers are defined for 802.11a/n/ac/ax.

**Table 3-3:** *5GHz Wi-Fi Channels per IEEE Std 802.11a-1999, 802.11n-2009, 802.11ac-2013, and 802.11ax-2021.*

| Channel Number | Center Frequency (MHz) |
|---|---|
| 36 | 5180 |
| 40 | 5200 |
| 44 | 5220 |
| 48 | 5240 |
| 52 | 5260 |
| 56 | 5280 |
| 60 | 5300 |
| 64 | 5320 |
| 100 | 5500 |
| 104 | 5520 |
| 108 | 5540 |
| 112 | 5560 |
| 116 | 5580 |
| 120 | 5600 |
| 124 | 5620 |
| 128 | 5640 |
| 132 | 5660 |
| 136 | 5680 |
| 140 | 5700 |
| 144 | 5720 |
| 149 | 5745 |
| 153 | 5765 |
| 157 | 5785 |
| 161 | 5805 |
| 165 | 5825 |
| 169 | 5845 |
| 173 | 5865 |
| 177 | 5885 |

### 3.1.4 The 5.9 GHz Channels

**Table 3-4:** *5.9 GHz Wi-Fi Channels per IEEE Std 802.11p-2010.*

| Channel Number | Center Frequency (MHz) |
|---|---|
| 100 | 5500 |

*Continued on next page*

| Channel Number | Center Frequency (MHz) |
|---|---|
| 104 | 5520 |
| 108 | 5540 |
| 112 | 5560 |
| 116 | 5580 |
| 120 | 5600 |
| 124 | 5620 |
| 128 | 5640 |
| 132 | 5660 |
| 136 | 5680 |
| 140 | 5700 |
| 171 | 5855 |
| 172 | 5860 |
| 173 | 5865 |
| 174 | 5870 |
| 175 | 5875 |
| 176 | 5880 |
| 177 | 5885 |
| 178 | 5890 |
| 179 | 5895 |
| 180 | 5900 |
| 181 | 5905 |
| 182 | 5910 |
| 183 | 5915 |
| 184 | 5920 |

### 3.1.5 Channel Numbering

The standard method to denote 5 GHz channels is to use the 20 MHz center channel numbers, even when referring to wider channel widths (e.g., 40 MHz, 80 MHz, or 160 MHz). The following are the valid non-overlapping primary channels for 802.11n/ac/ax in NetSim:

20MHz: 36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 144, 149, 153, 157, 161, 165

40 MHz: 38, 46, 54, 62, 102, 110, 118, 126, 134, 142, 151, 159

80 MHz: 42, 58, 106, 122, 138, 155

160 MHz: 50, 114

### 3.1.6 WLAN PHY Rate in NetSim

**Table 3-5:** *WLAN PHY Rates in NetSim.*

| WLAN Standard | Frequency (GHz) | Bandwidth (MHz) | MIMO streams | PHY rate (Mbps) |
|---|---|---|---|---|
| a | 5 | 20 | N/A | 6, 9, 12, 18, 24, 36, 48, 54 |
| b | 2.4 | 20 | N/A | 1, 2, 5.5, 11 |
| g | 2.4 | 20 | N/A | 6, 9, 12, 18, 24, 36, 48, 54 |
| **n** | 2.4, 5 | 20 | 4 | Up to 288 |
| | | 40 | | Up to 600 |
| **ac** | 5 | 20 | 8 | Up to 693.3 |
| | | 40 | | Up to 1600.0 |
| | | 80 | | Up to 3466.7 |
| | | 160 | | Up to 6933.3 |
| **ax** | 2.4 | 20 | 8 | Up to 1147 |
| | | 40 | | Up to 2294 |
| | 5 | 20 | | Up to 1147 |
| | | 40 | | Up to 2294 |
| | | 80 | | Up to 4803 |
| | | 160 | | Up to 9607 |

### 3.1.7 SIFS, Slot Time, CW Min, and CW Max settings

**Table 3-6:** *DSSS PHY characteristics (IEEE-Std-802.11-2020 – Page no 2763 and 2764).*

| Sub Std. | b (20MHz) |
|---|---|
| SIFS | 10 |
| Slot Time | 20 |
| CW Min | 31 |
| CW Max | 1023 |

**Table 3-7:** *OFDM Physical Characteristics (IEEE-Std-802.11-2020 – Page no 2847).*

| Sub Std. | a | g | p (5MHz) | p (10MHz) | p (20MHz) |
|---|---|---|---|---|---|
| Bandwidth | 20MHz | 20MHz | 5MHz | 10MHz | 20MHz |
| SIFS | 16 | 16 | 64 | 32 | 16 |
| Slot Time | 9 | 9 | 21 | 13 | 9 |
| CW Min | 15 | 15 | 15 | 15 | 15 |
| CW Max | 1023 | 1023 | 1023 | 1023 | 1023 |

**Table 3-8:** *HT PHY characteristics (IEEE-Std-802.11-2020 – Page no 2952) and MIMO PHY characteristics (IEEE-Std-802.11n-2009 – Page no 335).*

| Sub Std. | n | |
|---|---|---|
| Frequency Band | 2.4GHz | 5GHz |
| SIFS | 10 | 16 |
| Slot Time | 20 | 9 |
| CW Min | 15 | 15 |
| CW Max | 1023 | 1023 |

**Table 3-9:** *Slot time in IEEE-Std-802.11-2020 – Page no 3094 and IEEE-Std-802.11ac-2013 – Page no 297.*

| Sub Std. | ac (5GHz) |
|---|---|
| SIFS | 16 |
| Slot Time | 9 |
| CW Min | 15 |
| CW Max | 1023 |

**Table 3-10:** *The PHY characteristics (IEEE-Std-802.11-2024).*

| Sub Std. | ax | |
|---|---|---|
| Frequency Band | 2.4GHz | 5GHz |
| SIFS | 10 | 16 |
| Slot Time | 20 | 9 |
| CW Min | 15 | 15 |
| CW Max | 1023 | 1023 |

### 3.1.8 PHY Implementation

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 end-systems, routers, switches, etc. It provides estimates of the statistics of application-level performance metrics such as throughput, delay, packet-loss, and statistics of network-level processes such as buffer occupancy, collision probabilities, etc.

To achieve scalable network simulation that can execute in reasonable time on desktop-level computers, in all networking technologies the details of the physical layer techniques have been abstracted up to the point that bit-error probabilities can be obtained from which packet error probabilities are obtained.

NetSim does not implement any of the digital communication functionalities of the PHY layer. For PHY layer simulation, the modulation and coding scheme, along with the transmit power, path loss, noise, and interference, determine the bit rate and the bit error rate by using well-known formulas or tables for the particular PHY layer being used. Users would need to use a PHY Layer/RF/Link Level simulator for simulating various digital communication and link level functionalities. Typically, these simulators will simulate just one transmitter-receiver pair,

rather than a network.

Generally, in NetSim, the PHY layer parameters available for the user to modify are Channel Bandwidth, Channel Centre Frequency, Transmit-power, Receiver-sensitivity, Antenna-gains, and the Modulation-and-Coding-Scheme. When simulating standard protocols, these parameters can only be chosen from a standard-defined set. NetSim also has standard models for radio pathloss; the parameters of these pathloss models can also be set.

### 3.1.9 PHY States

The PHY radio states implemented in NetSim 802.11 are RX ON IDLE, RX ON BUSY, TRX ON BUSY.

- **RX ON IDLE:** This is the default radio state.
- **RX ON BUSY:** This state is set at receiver radio when the reception of data begins. Upon completion of reception, it changes to RX ON IDLE.
- **TRX ON BUSY:** This state is set at the transmitter radio at the start of frame transmission. Upon completion of transmission, it changes to RX ON IDLE.
- A node in backoff slots can be considered as equivalent to CCA busy. In NetSim, the radio state continues to be in RX ON IDLE.
- SLEEP state is not implemented since NetSim 802.11 does not currently implement power save mode.

### 3.1.10 802.11 implementation details

Packets arriving from the Network layer are queued in an access buffer, where they are sorted according to their priority per 802.11 EDCA. An event MAC OUT with Subevent CS (Carrier Sense – CSMA) is added to check if the medium is free.

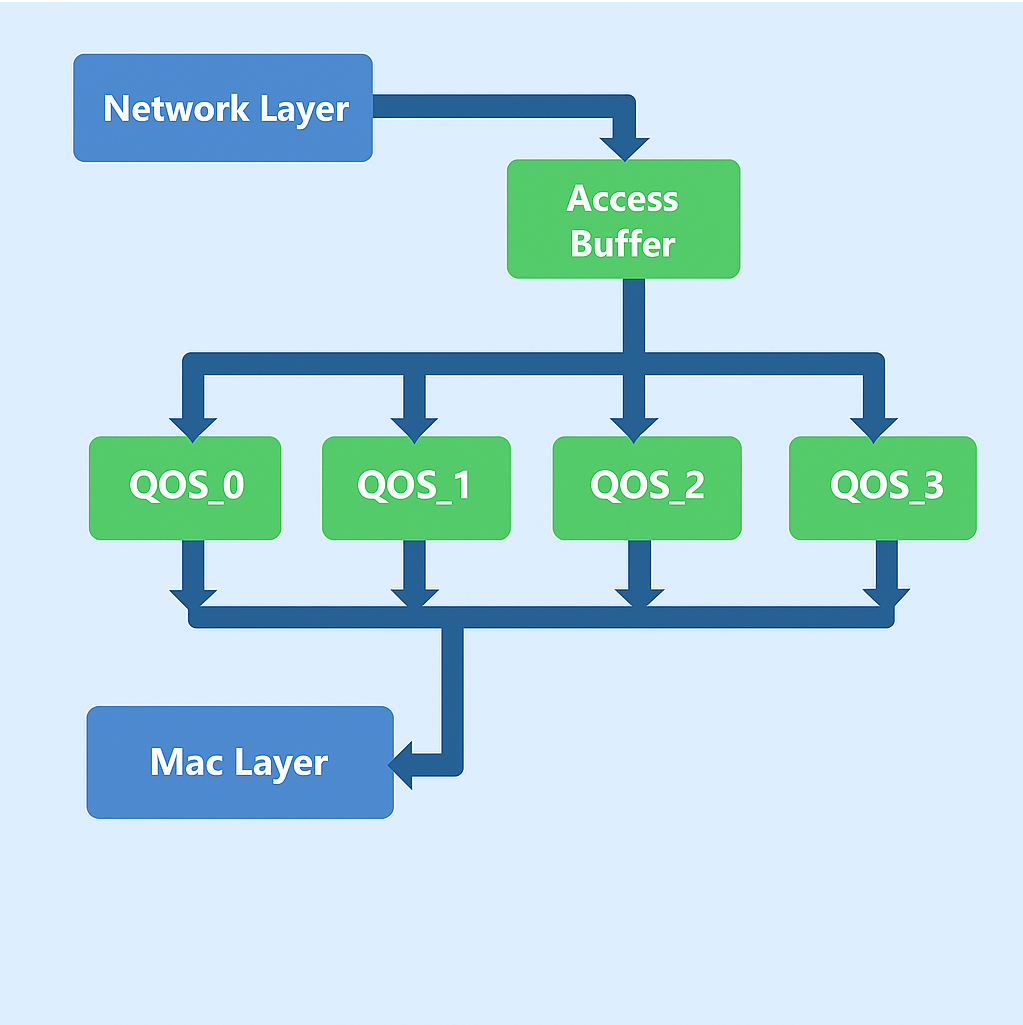



**Figure 3-1:** *Packet transmission from the Network layer to the Mac Layer and how they are queued up in an access buffer.*

During CS, if the medium is free, the NAV is checked. This occurs if the RTS/CTS mechanism is enabled, which can be done by adjusting the RTS Threshold. If the Present Time > NAV, then an Event MAC OUT with Subevent DIFS End is added at the time Present Time + DIFS time.

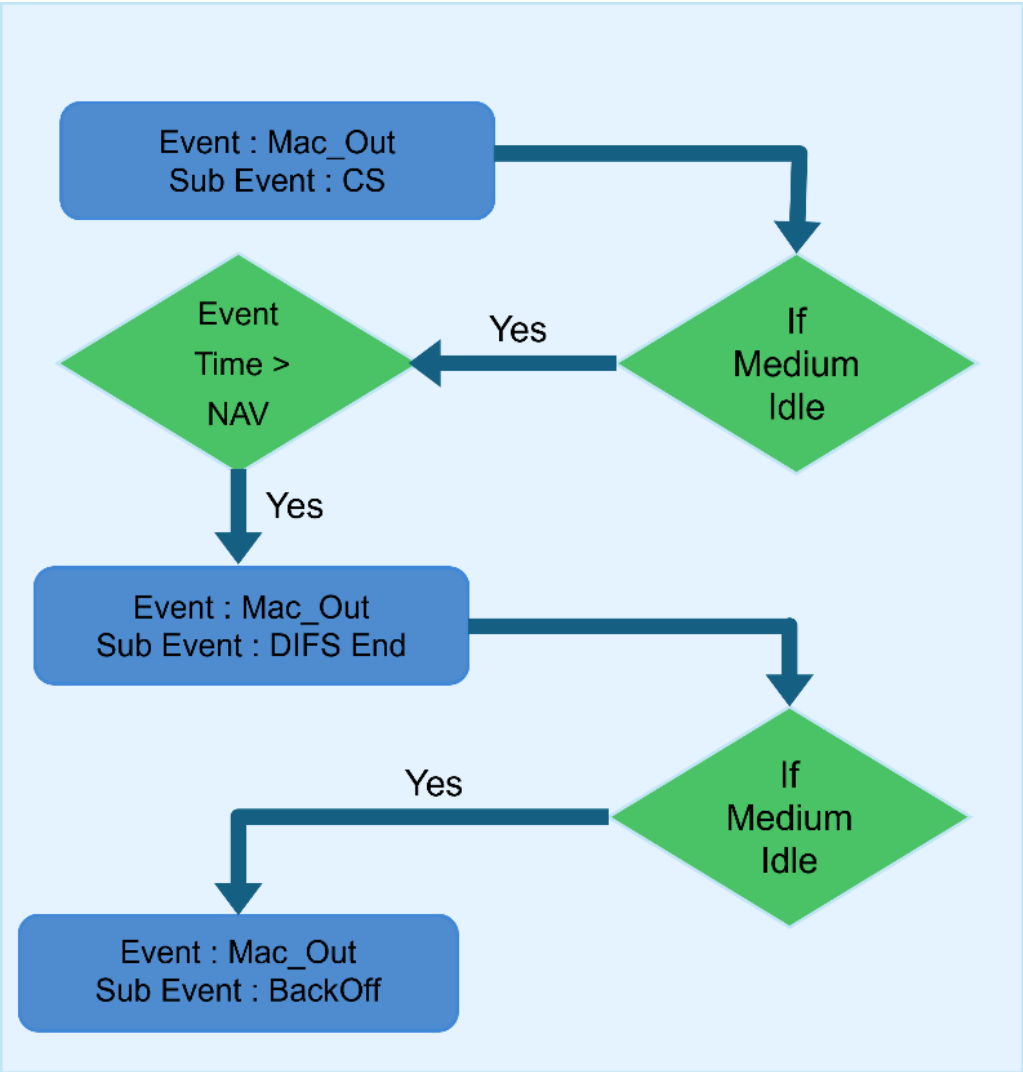



**Figure 3-2:** *Event and Subevent in Mac layer.*

The medium is checked at the end of the DIFS time period. A random backoff is then calculated based on the Contention Window (CW). An Event MAC OUT with the Subevent BackOff is added at time Present Time + BackOff Time.

Once the backoff is successful, NetSim starts the transmission process by aggregating frames from the QoS Buffer and storing them in the Retransmit Buffer. If the A-MPDU size is greater than the RTS Threshold, the RTS/CTS mechanism is enabled, which is an optional feature.

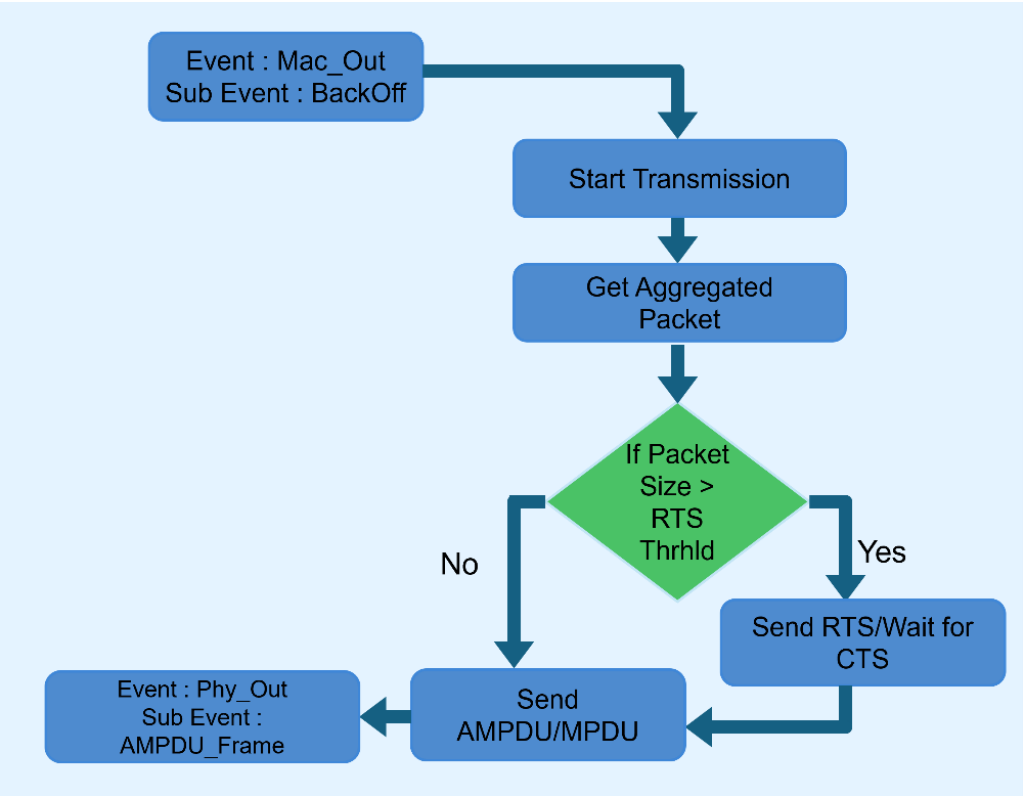



**Figure 3-3:** *Event and Subevent in Mac layer and Phy layer.*

NetSim sends the packet by calling the PHY OUT Event with Subevent AMPDU Frame. Note that the implementation of A-MPDU is in the form of a linked list.

Whenever a packet is transmitted, the medium is made busy and a Timer Event with Subevent Update Device Status is added at the transmission end time to set the medium as idle again.

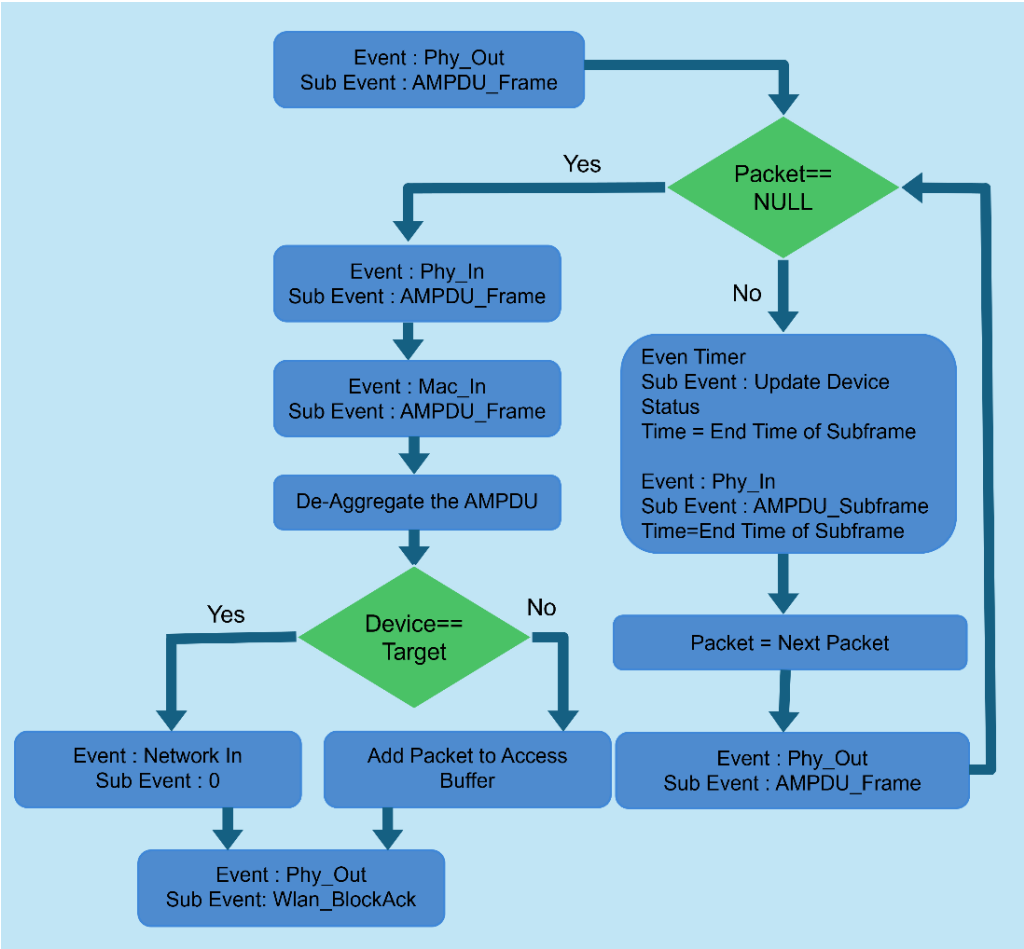



**Figure 3-4:** *Event and Subevent in Phy layer.*

Events PHY OUT Subevent AMPDU Subframe, Timer Event with the Subevent Update Device Status and Event PHY IN Subevent AMPDU Subframe are added in succession for each MPDU (Subframe of the aggregated frame). This is done for collision calculations. If two stations start transmission simultaneously, some of the Subframes may collide. Only those collided Subframes are retransmitted. The same logic is applied to errored packets. However, if the PHY header (the first packet) is errored or collided, the entire A-MPDU is resent.

At the receiver, the device de-aggregates the frame in the MAC Layer and generates a block ACK, which is sent to the transmitter. If the receiver is an intermediate node, the de-aggregated frames are added to the access buffer of the receiver in addition to the packets which arrive from the Network layer. If the receiver is the destination, then the received packets are sent to the Network layer. At the transmitter side, when the device receives the block acknowledgement, it retransmits only those packets which are errored. The rest of the packets are deleted from the retransmit buffer. This is done until all packets are transmitted successfully or a retransmit limit is reached, after which the next set of frames are aggregated to be sent.

### 3.1.11 802.11 ac/ax MAC and PHY Layer Implementation

Improvements in 802.11ac/ax compared to 802.11n:

**Table 3-11:** *Feature Comparison between 802.11ac/ax to 802.11n.*

| Feature | 802.11n | 802.11ac | 802.11ax |
|---|---|---|---|
| Spatial Streams | Up to 4 streams | Up to 8 streams | Up to 8 streams |
| MIMO | Single User MIMO | Multi-User MIMO | Multi-User MIMO |
| Channel Bandwidth | 20 and 40 MHz | 20, 40, 80 and 160 MHz (optional) | 20, 40, 80 and 160 MHz |
| Modulation | BPSK, QPSK, 16QAM, 64QAM | BPSK, QPSK, 16QAM, 64QAM, 256QAM (opt.) | BPSK, QPSK, 16QAM, 64QAM, 256QAM, 1024 QAM |
| Max Aggregated Packet Size | 65536 octets | 1048576 octets | 4,194,304 octets |

MAC layer improvements include only the increase in the number of aggregated frames from 1 to 64. The MCS index for different modulation and coding rates is as follows:

**Table 3-12:** *Different Modulation schemes and Code Rates.*

| MCS index | Modulation | Code Rate |
|---|---|---|
| 0 | BPSK | 1/2 |
| 1 | QPSK | 1/2 |
| 2 | QPSK | 3/4 |
| 3 | 16QAM | 1/2 |
| 4 | 16QAM | 3/4 |
| 5 | 64QAM | 2/3 |
| 6 | 64QAM | 3/4 |
| 7 | 64QAM | 5/6 |
| 8 | 256QAM | 3/4 |
| 9 | 256QAM | 5/6 |
| 10 (11ax only) | 1024 QAM | 3/4 |
| 11 (11ax only) | 1024 QAM | 5/6 |

Receiver sensitivity for different modulation schemes in 802.11ac/ax (for a 20MHz channel bandwidth) is as follows.

**Table 3-13:** *MCS index vs. Receiver Sensitivity (Rx-sensitivity).*

| MCS Index | Receiver Sensitivity (dBm) |
|---|---|
| 0 | −82 |
| 1 | −79 |
| 2 | −77 |
| 3 | −74 |
| 4 | −70 |
| 5 | −66 |
| 6 | −65 |
| 7 | −64 |
| 8 | −59 |
| 9 | −57 |
| 10 | −54 |
| 11 | −52 |

The Rx-sensitivity is then set per the above table in conjunction with Max Packet Error Rate

(PER) as defined in the standard.

If users wish to apply just the Rx-sensitivity (also termed as rate dependent input level), then the `calculate_rxpower_by_per()` function call in the function `fn_NetSim_IEEE802_11_HTPhy_UpdateParameter()` in the file `IEEE802_11_HT_PHY.c` (equivalent functions in `IEEE802_11_HEPhy.c`) can be commented out.

Number of subcarriers for different channel bandwidths:

**Table 3-14:** *Number of subcarriers for different channel bandwidths.*

| PHY Standard | Subcarriers | Capacity relative to 20MHz in 802.11ac |
|---|---|---|
| 802.11n/802.11ac 20MHz | Total 64, 52 Data, 4 pilot | x1.0 |
| 802.11n/802.11ac 40MHz | Total 128, 108 Data, 6 pilot | x2.1 |
| 802.11ac 80MHz | Total 256, 234 Data, 8 pilot | x4.5 |
| 802.11ac 160MHz | Total 512, 468 Data, 16 pilot | x9.0 |

**Table 3-15:** *Number of subcarriers for different channel bandwidths (11ax HE SU OFDM).*

| PHY Standard | Subcarriers | Capacity relative to 20MHz in 802.11ac |
|---|---|---|
| 20 MHz | Total 256, 234 Data, 8 pilot | x1.65 |
| 40 MHz | Total 512, 468 Data, 16 pilot | x3.3 |
| 80 MHz | Total 1024, 980 Data, 16 pilot | x6.92 |
| 160 MHz | Total 2048, 1960 Data, 32 pilot | x13.85 |

With the knowledge of MCS index and bandwidth of the channel, the data rate is set in the following manner:

1. Get the number of subcarriers that are usable (data) for the given bandwidth of the medium.
2. Get the Number of Bits per Subcarrier (NBPSC) from selected MCS.
3. Number of Coded Bits Per Symbol (NCBPS) = NBPSC × Number of Subcarriers.
4. Number of Data Bits Per Symbol (NDBPS) = NCBPS × Coding Rate.
5. Physical level Data Rate = NDBPS / Symbol Time (For 802.11n/ac, the total OFDM symbol duration is 3.6 microseconds with a 0.4 $\mu$s (short) Guard Interval and 4 microseconds with a 0.8 $\mu$s (long) Guard Interval. In 802.11ax, the total OFDM symbol durations are 13.6 $\mu$s, 14.4 $\mu$s, and 16 $\mu$s corresponding to Guard Intervals of 0.8 $\mu$s, 1.6 $\mu$s, and 3.2 $\mu$s, respectively.)

### 3.1.12 MAC Aggregation in NetSim

NetSim supports A-MPDU aggregation and does not support A-MSDU aggregation. MAC Aggregation is independent of MCS (PHY Rate) or BER. It is the PHY Rate that adapts to BER via Rate Adaptation algorithms.

In the aggregation scheme shown in Figure 3-5, several MPDUs (MAC Protocol Data Units) are aggregated into a single A-MPDU (Aggregated MPDU). The A-MPDUs are created before

transfer to the PHY. The MAC does not wait for MPDUs to aggregate. It aggregates the frames already queued to form an A-MPDU. The maximum size of an A-MPDU is 4,692,480 bytes.

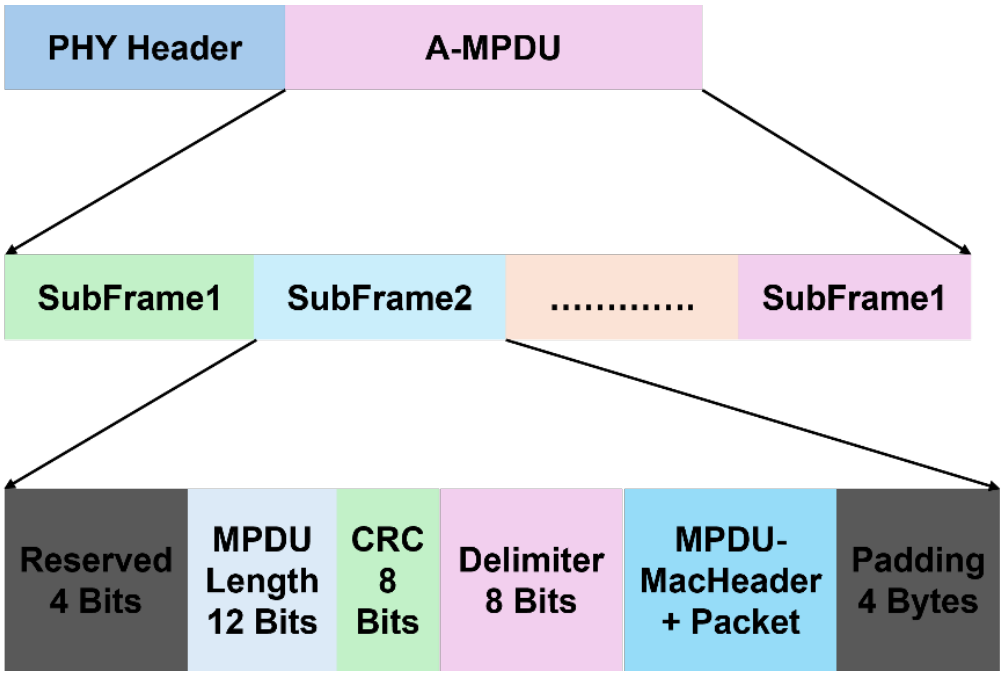



**Figure 3-5:** *Aggregation scheme.*

In 802.11n, a single block acknowledgement is sent for the entire A-MPDU. The block ack acknowledges each packet that is received. It consists of a bitmap (compressed bitmap) of 64 bits or 8 bytes. This bitmap can acknowledge up to 64 packets, 1 bit for each packet.

The value of a bitmap field is 1, if the respective packet is received without error; otherwise it is 0. Only the errored packets are present until a retry limit is reached. The number of packets in an A-MPDU is restricted to 64 since the size of block ack bitmap is 64 bits.

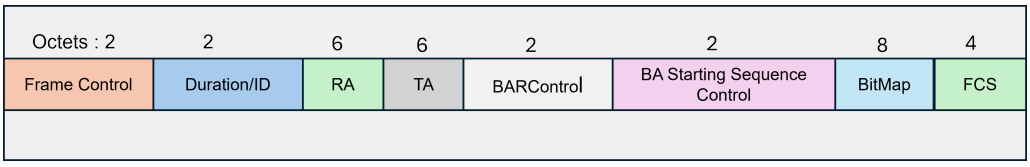



**Figure 3-6:** *Block Ack Control Packet.*

**NOTE:**

- NetSim uses the parameter Number of frames to aggregate, while the standard uses the parameter A-MPDU Length Exponent. Per standard the A-MPDU length is defined by two parameters: Max AMPDU length exponent and BLOCK ACK Bitmap. The AMPDU length in bytes is $2^{(13+\text{MaximumAMPDULengthExponent})} - 1$.
- Since NetSim does not model A-MSDU, a design decision was made to model A-MPDU based on Block ACK bitmap size (to indicate the received status of up to 64 frames) and therefore the parameter – Number of Frames to Aggregate – in the GUI.
- When EDCA is enabled, packet aggregation is done separately for each QoS class.
- NetSim ignores the padding bytes added to the MPDU.

- The MAC aggregates packets destined for the same receiver, regardless of the end destination. The receiver is to be understood as the next hop in a wireless transmission.
- The RTS threshold is compared against the total A-MPDU size.
- Aggregation functionality may not execute correctly if:

$$\text{NumberOfFramesToAggregate} \times \text{PacketSize (B)} > 4{,}692{,}480 \text{ (B)} \tag{1}$$

### 3.1.13 Signal to interference and noise ratio (SINR)

At each receiver, when the first packet is transmitted and every time the transmitter or receiver moves, NetSim calculates the received signal level from the transmitter. The received signal level would be equal to transmit power less propagation losses. Next, NetSim calculates the interference received (at the same receiver), from all the interfering transmissions. Only co-channel interference is accounted for, and adjacent channel interference is not calculated. Finally, NetSim takes the ratio of the signal level, to the sum of the total interference from other transmissions plus the thermal noise. This ratio is SINR.

Once the SINR is calculated, the BER is obtained from the SINR-BER tables for the applicable modulation scheme. This BER is then converted to Packet-Error-Rate. Packet error (Yes/No) is determined by drawing a random number in (0, 1) and comparing against PER.

The same is explained diagrammatically below.

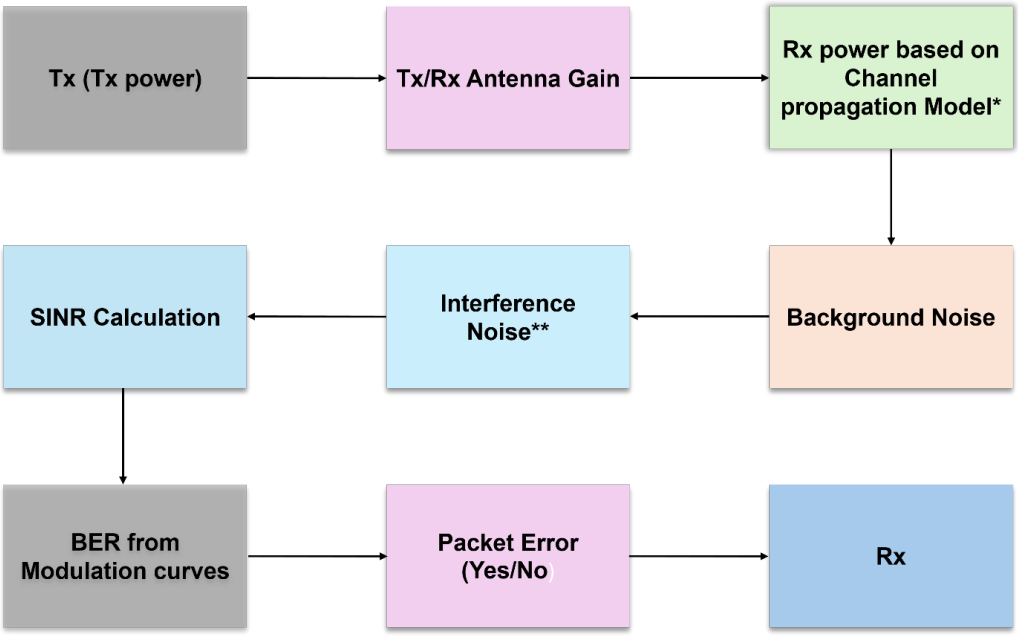



**Figure 3-7:** *Radio Tx-Rx for one transmission.*

* Propagation model covers path loss, fading and shadowing. The models are documented in a separate document named Propagation-Models.pdf.

** Interference noise due to other transmissions within the network.

### 3.1.14 Error Model

- **BER by Table:** Packet Error Probability (PEP) is derived from SINR-MCS (Signal-to-Interference-plus-Noise Ratio – Modulation and Coding Scheme) lookup tables, as referenced in the Cisco document available at https://community.cisco.com/legacyfs/online/attachments/discussion/revolution-wi-fi-mcs-to-snr-single-page.pdf. They map specific SINR values to Bit Error Rates (BER), which are then used to calculate the probability of packet errors. These tables are implemented in the `SINR-BER.c` file, which is part of the medium project. Users can edit these table entries to tailor the simulation to specific needs or scenarios.
- **BER by formula:** The Bit Error Rate (BER) is derived from SINR-BER formulas outlined in section 6 of the NetSim manual propagation-model.pdf. The packet error probability (PEP) is calculated from the BER by considering the packet length.

### 3.1.15 Transmit Power

The user can set a fixed transmit power via the GUI. Transmit power is a local variable, allowing each STA and AP to have unique transmit power settings. The transmit power can be dynamically varied by modifying the underlying 802.11 source C code.

### 3.1.16 Carrier Sense

Transmit power less propagation losses is the received power. The propagation loss is the sum (in dB scale) of pathloss, shadowing loss and fading loss. Various propagation models are available and are detailed in the Propagation model manual. Pathloss, Fading, and Shadowing can be turned on/off in GUI.

If ReceiverSensitivity (Lowest MCS) $\geq$ Receiver-Power $\geq$ ED-Threshold, the medium is set to busy. Note that CSMA/CA algorithm operates according to the medium state (busy/idle).

If Received-Power $>$ Receiver-Sensitivity (Lowest MCS) then MCS is set depending on the received power and the signal is decoded. Packet errors are decided by looking up the SINR-BER table for the given MCS.

The variables can also be modified dynamically by changing the underlying 802.11 source C code.

### 3.1.17 Transmission Range, Carrier Sense Range, and Interference Range

- **Transmission Range:** The transmission range is the range within which the receiver of a signal can decode the source's transmission correctly (when no other transmitting node's signal interferes). This is typically smaller than the carrier-sensing range of the transmitter.
- **Carrier Sense Range:** The carrier-sense range is the range within which the transmitter's signal exceeds the Carrier Sense Threshold of the receiver (or another transmitter). The receiver (or another transmitter) detects the medium to be busy and does not transmit at this time.
- **Interference Range:** The interference range (defined by the receiver) is the range within which any signal transmitted by other sources interferes with the transmission of the intended source, thereby causing a loss (marked as a collision in NetSim) at the receiver.

These three ranges are affected by the power of the transmitter. The greater the transmission power, the further a node can receive the transmission, and also the more nodes whose communication with other nodes will be affected by this transmission. The transmission range is also affected by the MCS used by the transmitter. The higher the MCS, the shorter the range, and vice versa.

### 3.1.18 Carrier Sense (CS) Threshold

In NetSim (from v13.2 onwards) the Carrier sense (CS) threshold is set equal to Control rate receive sensitivity.

$$\text{CSThreshold} = \text{ReceiverSensitivity}(\text{ControlRate}) \tag{2}$$

Users can modify the CS Threshold using the variable `CSRANGEDIFF` which is set to 0 dB in code by default. This implies a 0 dB differential between the lowest MCS (Control rate) Receive sensitivity (which determines DecodeRange) and CS Threshold (which determines CarrierSenseRange). The value of `CSRANGEDIFF` can be modified by the user in NetSim Standard or Pro versions, which ship with source code. We believe the term EDThreshold used in the literature is the same as CSThreshold.

If the interference signal power (sum of the Received-power from all other transmitters), measured at the transmitter, is greater than ED-Threshold, then the transmitter assumes the medium is busy. Carrier is sensed by the transmitter; all CS activity occurs at the transmitter, and not at the receiver.

### 3.1.19 Transmitter's choice of MCS

If the Rate Adaptation algorithm is turned off, then the transmitter chooses MCS by comparing the RSS (calculated per the equation below) against the Receiver-Sensitivity for different MCS (per the tables in the standards). The highest possible MCS is chosen. This means the MCS is not fixed but adapts to the received signal strength, even with rate adaptation turned off in the MAC layer.

NetSim exploits the AP-STA and the STA-AP channel reciprocity. Therefore, pathloss plus shadow loss is identical in both directions.

$$\text{RSSI} = \text{TxPower} - \text{Pathloss} - \text{ShadowLoss} \tag{3}$$

Note that when computing BER (from SINR) fading loss is added to this RSSI value. Thus, fading loss is not accounted for when choosing MCS, but is accounted for when computing BER.

NetSim has rate adaptation algorithms which take care of selecting the right MCS for a given SINR. In the simplest algorithm for every 20 successful transmissions, the rate (MCS) goes up 1 step, and for every 3 continuous failures, the rate goes down one step.

### 3.1.20 Hidden Node Behavior

Consider N1 and N3 transmitting to N2 where N1 and N3 are beyond Carrier sense (CS) range. N1 is said to be hidden from N3 and vice versa.

When N1 and N3 transmit, there are likely to be collisions at N2. However, collisions do not

occur all the time. The CSMA/CA algorithm exponentially increases the backoff and hence after a few collisions it is possible that one of the nodes gets a low backoff number while the other draws a very high backoff number. Thus, the node with low backoff can complete transmissions (of one and even more than one packet) while the other node (with the large backoff) is still in backoff.

When N1 transmits to N2, N3 cannot hear the transmission since N3 is beyond CS. Therefore, N3 can attempt if its backoff counts down to 0. However, when N2 sends back the WLAN-ACK, N3 will hear it since N3 is within range of N2. Therefore, in NetSim, N3 will sense the medium as busy and freeze its backoff when N2 is sending the WLAN ACK to N1.

In case of N2 to N1/N3 transmissions, then the reverse is true for the MAC-ACK from the nodes. When N2 sends a packet to N1 (or N3) it is within range of N3 (or N1), however, when N1 (or N3) sends back the MAC ACK there is a chance of collision with a data packet of N3 (or N1).

### 3.1.21 IEEE 802.11 e QoS and EDCA

Quality of Service (QoS) provides the ability to specify parameters on multiple queues for increased throughput and better performance of differentiated wireless traffic like Voice-over-IP (VoIP), other types of audio, video, and streaming media, as well as traditional IP data over the Access Point.

QoS was introduced in 802.11e and is achieved using enhanced distributed channel access functions (EDCAFs). EDCA provides differentiated priorities to transmitted traffic, using four different access categories (ACs). With EDCA, high-priority traffic has a higher chance of being sent than low-priority traffic: a station with high priority traffic waits a little less before it sends its packet, on average, than a station with low priority traffic. This differentiation is achieved through varying the channel contention parameters i.e., the amount of time a station would sense the channel to be idle, and the length of the contention window for a backoff.

In addition, EDCA provides contention-free access to the channel for a period called a Transmit Opportunity (TXOP). A TXOP is a bounded time interval during which a station can send as many frames as possible (as long as the duration of the transmissions does not extend beyond the maximum duration of the TXOP). If a frame is too large to be transmitted in a single TXOP, it should be fragmented into smaller frames. The use of TXOPs reduces the problem of low rate stations gaining an inordinate amount of channel time in the legacy 802.11 DCF MAC. A TXOP time interval of 0 means it is limited to a single MPDU.

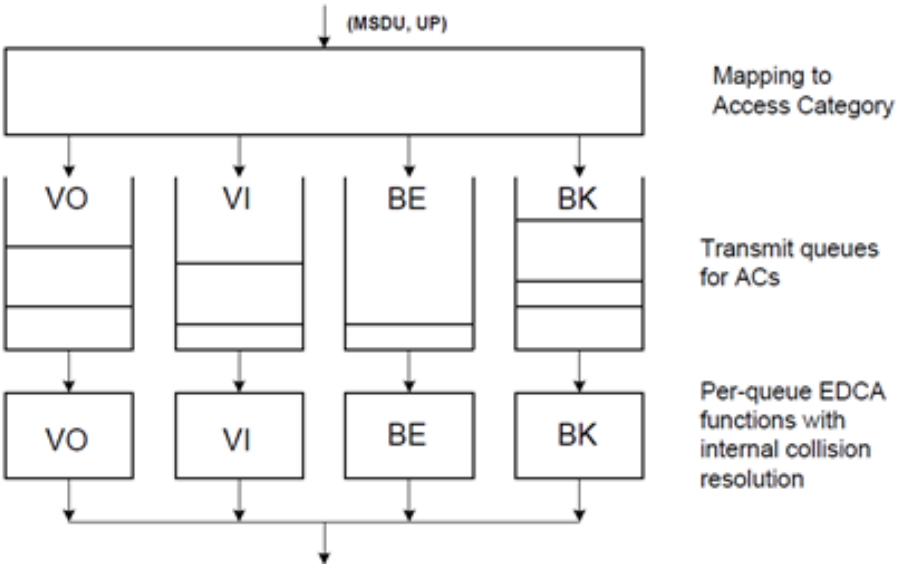



**Figure 3-8:** *Enhanced Distributed Channel Access (EDCA) in 802.11.*

NetSim categorizes application packets based on QoS class set in application properties as follows:

- **VO:** UGS and RTPS
- **VI:** NRTPS and ERTPS
- **BE:** BE and all control packets such as TCP ACKs
- **BK:** Everything else

### Default EDCA Parameters

The following tables show the default EDCA parameters. This default parameter set is per page 899, IEEE Std 802.11-2016.

**Table 3-16:** *Default EDCA access parameters for 802.11 b for both AP and STA.*

| Access Category | CW min | CW max | AIFSN | Max TXOP ($\mu$s) |
|---|---|---|---|---|
| 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 3-17:** *Default EDCA access parameters for 802.11 a / g / n / ac / ax for both AP and STA.*

| Access Category | CW min | CW max | AIFSN | Max TXOP ($\mu$s) |
|---|---|---|---|---|
| Background (AC-BK) | 15 | 1023 | 7 | 2528 |
| Best Effort (AC-BE) | 15 | 1023 | 3 | 2528 |
| Video (AC-VI) | 7 | 15 | 2 | 4096 |
| Voice (AC-VO) | 3 | 7 | 2 | 2080 |

**Table 3-18:** *Default EDCA access parameters for 802.11 p (dot11OCBActivated is true).*

| Access Category | CWmin | CWmax | AIFSN | Max TXOP ($\mu$s) |
|---|---|---|---|---|
| Background (AC-BK) | 15 | 1023 | 9 | 0 |
| Best Effort (AC-BE) | 15 | 1023 | 6 | 0 |
| Video (AC-VI) | 7 | 15 | 3 | 0 |
| Voice (AC-VO) | 3 | 7 | 2 | 0 |

**NOTE:** The EDCA parameters can be configured by changing the Physical type parameter according to the different standard, IEEE802.11b (Medium Access Protocol ▹ DSSS), IEEE802.11n (Medium Access Protocol ▹ HT), IEEE802.11ac (Medium Access Protocol ▹ VHT), IEEE 802.11ax (Medium Access Protocol ▹ HE), IEEE802.11a and g (Medium Access Protocol ▹ OFDM and OCBA ▹ FALSE), IEEE802.11p (Medium Access Protocol ▹ OFDM and OCBA ▹ TRUE).

### 3.1.22 Rate Adaptation

In NetSim (with default code), rate adaptation works as follows:

1. **FALSE:** This is similar to Receiver Based Auto Rate (RBAR) algorithm. In this, the PHY rate gets set based on the target PEP (packet error probability) for a given packet size, as given in the standard. The adaptation is termed as "FALSE" since the rate is pre-determined as per standard and there is no subsequent "adaptation".
   (a) 802.11 n/ac/ax: Target PEP = 0.1, Packet Size: 4096 B
   (b) 802.11 b: Target PEP = 0.08, Packet Size: 1024B
   (c) 802.11 a/g/p: Target PEP: 0.1, Packet size 1000B
2. **GENERIC:** This is similar to the Auto Rate FallBack (ARF) algorithm. In this algorithm:
   (a) Rate goes up one step for 20 consecutive packet successes
   (b) Rate goes down one step for 3 consecutive packet failures
3. **MINSTREL:** Per the minstrel rate adaptation algorithm implemented in Linux.

### 3.1.23 Model Limitations

1. Mobility of Wireless nodes is not available in infrastructure mode (when connected via an Access Point) and is only available in Adhoc mode. Hence mobility for wireless nodes can only be set when running MANET simulations.
2. Authentication and encryption are not supported.
3. While different APs can operate in different channels, all the Wireless nodes connected to one AP operate in the same channel.
4. No beacon generation, probing or association.
5. RTS, CTS and ACK are always transmitted at the base rate (lowest MCS).
6. Roaming, whereby a STA leaves a serving AP to associate with a target AP (usually based on RSSI/SNR), is not supported.

### 3.1.24 Wi-Fi GUI Parameters

The WLAN parameters can be accessed by clicking on an Access Point or Wireless Node, then going to the right-hand side panel and expanding Interface (Wireless) Properties ▷ Datalink and Physical Layers.

The full table of Wi-Fi GUI parameters (Table 3-19) is provided below as a long table due to its size.

**Table 3-19:** *Internetworks Config Properties.*

| Parameter | Scope | Range | Description |
|---|---|---|---|
| **Interface Wireless – Datalink Layer** | | | |
| Rate Adaptation | Cell | False / Minstrel / Generic | There is no rate adaptation, and the rate depends on the MCS selection option chosen in the PHY layer. Minstrel: Operates similar to the rate adaptation algorithm implemented in Linux. Generic: In this algorithm (i) Rate goes up one step for 20 consecutive packet successes, and (ii) Rate goes down one step after 3 consecutive packet failures. |
| Short Retry Limit | Local | 1 to 255 | Determines the maximum number of transmission attempts of a frame. The length of MPDU is less than/ equal to Dot11 RTS Threshold value, made before a failure condition is indicated. |
| Long Retry Limit | Local | 1 to 255 | Determines the maximum number of transmission attempts of a frame. The length of MPDU is greater than Dot11 RTS Threshold value, made before a failure condition is indicated. |
| Dot11 RTS Threshold | Local | 0 to 4692480 | The size of packets (or A-MPDU if applicable) above which RTS/CTS (Request to Send / Clear to Send) mechanism gets triggered. |
| MAC Address | Fixed | Auto Generated | The MAC address is a unique value associated with a network adapter. This is also known as hardware address or physical address. This is a 12-digit hexadecimal number (48 bits in length). |
| Buffer Size | Local | 1 to 100 | Buffer is the memory in a device which holds data packets temporarily. If incoming rate is higher than the outgoing rate, incoming packets are stored in the buffer. NetSim models the buffer as an egress buffer. Unit is MB. |

*Continued on next page*

| Parameter | Scope | Range | Description |
|---|---|---|---|
| Medium Access Protocol | Local | DCF / EDCAF | DCF is the process by which CSMA/CA is applied to Wi-Fi networks. DCF defines four components to ensure devices share the medium equally: Physical Carrier Sense, Virtual Carrier Sense, Random Back-off timers, and Interframe Spaces (IFS). DCF is used in non-QoS WLANs. EDCAF: QoS was introduced in 802.11e and is achieved using enhanced distributed channel access functions (EDCAFs). EDCA provides differentiated priorities to transmitted traffic, using four different access categories (ACs). With EDCA, high-priority traffic has a higher chance of being sent than low-priority traffic: a station with high priority traffic waits a little less before it sends its packet, on average, than a station with low priority traffic. |
| Physical Type | Global | DSSS / OFDM / HT / VHT / HE | DSSS: Direct Sequence Spread Spectrum. The physical type parameter is set to DSSS if the standard selected is IEEE802.11b. OFDM: Orthogonal Frequency Division Multiplexing is utilized as a digital multi-carrier modulation method. The physical type parameter is set to OFDM if the standard selected is IEEE802.11a, g and p. HT: Operates in frequency bands 2.4GHz or 5GHz band. The physical type parameter is set to HT if the standard selected is IEEE802.11n. VHT: The physical type parameter is set to VHT if the standard selected is IEEE802.11ac. HE: Operates in frequency bands 2.4 GHz or 5 GHz band. The physical type parameter is set to HE if the standard selected is IEEE802.11ax. |
| OCBA Activated | Local | True or False | This parameter determines the type of standard to be chosen for the OFDM physical type. The standard is set to IEEE802.11p if OCBA is True. The standard is set to IEEE802.11a and g if OCBA is False. |

*Continued on next page*

| Parameter | Scope | Range | Description |
|---|---|---|---|
| BSS Type | Fixed | Auto Generated | The BSS type is fixed to Infrastructure mode. The wireless device can communicate - with each other or with a wired network - through an Access Point. |
| CW min (Slots) | Local | 0 to 255 | Specifies the initial Contention Window (CW) used by an Access Point (or STA) for a particular AC for generating a random number for the back-off. |
| CW max (Slots) | Local | 0 to 65535 | At each collision the CW is doubled. $CW_{Max}$ specifies the final maximum CW values used by an Access Point (or STA) for a particular AC for generating a random number for the back-off. |
| AIFSN (Slot) | Local | 2 to 15 | Specifies the number of slots after a SIFS duration. |
| Max TXOP | Local | 0 to 65535 | Specifies the maximum number of microseconds of an EDCA TXOP for a given AC. Unit is microseconds. |
| MSDU Lifetime (TU) | Local | 0 to 500 | Specifies the maximum duration an MSDU would be retained by the MAC before it is discarded, for a given AC. MSDU Lifetime is specified in TU. |
| **Interface Wireless – Physical Layer** | | | |
| Protocol | Fixed | IEEE 802.11 | Defines the MAC and PHY specifications like IEEE802.11a/b/g/n/ac/ax/p for wireless connectivity for fixed, portable and moving stations within a local area. |
| Connection Medium | Fixed | Auto Generated | Defines how the devices are connected or linked to each other. |
| Standard | Cell | IEEE 802.11 a/b/g/n/ac/ax/p | Refers to a family of specifications developed by IEEE for WLAN technology. The IEEE standards supported in NetSim are IEEE 802.11 a, b, g, n, ac, ax and p. 802.11a provides up to 54 Mbps in 5 GHz band. 802.11b provides 11 Mbps in the 2.4 GHz bands. 802.11 g provides 54 Mbps transmission over short distances in the 2.4 GHz band. 802.11 n adds up MIMO. 802.11 ac supports wider channels and includes beamforming capabilities. 802.11 p provides support to Intelligent Transportation Systems. |

*Continued on next page*

| Parameter | Scope | Range | Description |
|---|---|---|---|
| Transmission Type | Fixed | DSSS / OFDM / HT / VHT | The transmission type parameter is DSSS if the standard selected is IEEE802.11b. OFDM if the standard selected is IEEE802.11a, g and p. HT if the standard selected is IEEE802.11n. VHT if the standard selected is IEEE802.11ac. |
| No. of Frames to Aggregate | Cell | 1 to 1024 (11ax), 1 to 64 (11ac), 1 to 64 (11n) | Number of frames aggregated to form an A-MPDU. This is fixed and cannot be dynamically varied (except by modifying the code). See Section 3.1.12 for more information. |
| Transmit Power | Local | 0 to 1000 | Transmitted signal power. Note that the transmit power is not split among the antennas. This value is applied to each antenna in a multi-antenna transmitter. Unit is mW. |
| Antenna Gain | Local | 0 to 1000 | A relative measure of an antenna's ability to direct or concentrate radio frequency energy in a particular direction or pattern. The measurement is typically measured in dBi (Decibels relative to an isotropic radiator). |
| Antenna Height | Local | 0 to 100m | It is used in the pathloss calculation in the following models: Cost231 Urban, Cost231 Hata SubUrban, Hata Urban, Hata SubUrban and Two Ray. This parameter has no effect when using any of the other pathloss models. Default:0.0 m. |
| SIFS | Fixed | Auto Generated | The time interval required by a wireless device in between receiving a frame and responding to the frame. Unit is microseconds. |
| Frequency Band | Cell | 2.4, 5 (Depends on the standard chosen) | Range of frequencies at which the device operates. The frequency band depends on the standard selected. Unit is GHz. |
| Bandwidth | Cell | 20, 40, 60, 80, 160 (Depends on the standard chosen) | The bandwidth depends on the standard and the frequency band selected. Unit is MHz. |
| CCA Mode | Fixed | Auto Generated | A mechanism to determine whether a medium is idle or not. It includes Carrier sensing and energy detection. |
| Slot Time | Fixed | Auto Generated | Time is quantized as slots in Wi-Fi. Unit is microseconds. |

*Continued on next page*

| Parameter | Scope | Range | Description |
|---|---|---|---|
| Standard Channel | Local | Depends on the standard chosen | The channel options defined in the standards. The options would also depend on the frequency band if the standard supports multiple bands. |
| CW Min | Fixed | Auto Generated | The minimum size of the Contention Window in units of slot time. The CW min is used by the MAC to calculate the backoff time for channel access during a carrier sense. |
| CW Max | Fixed | Auto Generated | The maximum size of the Contention Window in units of slot time. The CW is doubled progressively when collisions occur. |
| Transmitting Antennas | Local | 1 to 8 | The number of transmit antennas. Note that power is not split among the transmit antennas but is assigned to each antenna. (The pair of Tx and Rx antenna present only for 802.11ac and 802.11n) |
| Receiving Antennas | Local | 1 to 8 | The number of receive antennas. |
| Guard Interval | Local | 400 and 800 ns; 800, 1600 and 3200ns (11ax) | Guard Interval is intended to avoid signal loss from multipath effect. Unit is nanoseconds. |
| MCS Selection | Local | Auto Rate Fallback, Fixed | MCS selection in Wi-Fi impacts data rates and efficiency. Auto Rate Fallback adapts the MCS based on signal quality (RSSI) per the RSSI-MCS tables in the 802.11 standards. Fixed MCS locks the MCS. Default Value: Auto Rate |
| Data MCS | Local | 802.11b: 0-3, 802.11a/g/p: 0-7, 802.11n: 0-7, 802.11ac: 0-9 (MCS 9 not available for 20MHz in VHT), 802.11ax: 0-11 | Allows selection of the MCS value for different Wi-Fi standards. Determines the modulation and coding scheme. Default Value: 0. |
| Data PHY Rate (Mbps) | Local | Determined by selected Data MCS and Wi-Fi standard | Shows the physical layer data rate based on the chosen modulation and coding scheme. (MCS) |

*Continued on next page*

| Parameter | Scope | Range | Description |
|---|---|---|---|
| Error Model | Local | SINR-BER-By-Table, SINR-BER-By-Formula | Specifies how the Bit Error Rate (BER) is calculated: BER is determined based on predefined tables mapping SINR to BER. BER is calculated using mathematical formulas that account for the modulation and coding schemes used, based on the SINR value. |

### 3.1.25 IEEE 802.11 Results

IEEE 802.11 performance metrics will be displayed in the results dashboard if the network scenario simulated consisted of at least one device with WLAN protocol enabled.

**Table 3-20:** *Description of IEEE 802.11 Metrics.*

| Parameter | Description |
|---|---|
| Device Id | It represents the Id of the wireless device which supports 802.11 (WLAN) |
| Interface Id | It represents the interface IDs of the wireless nodes |
| Frame Sent | It is the number of frames sent by Access Point |
| Frame Received | It is the number of frames received by a wireless node |
| RTS Sent | It is the number of Request to send (RTS) packets sent by a Wireless Node. RTS/CTS frames are sent prior to transmission when the packet size exceeds the RTS threshold. The access point receives the RTS and responds with a CTS frame. The station must receive a CTS frame before sending the data frame. The CTS also contains a time value that alerts other stations to hold off from accessing the medium while the station initiating the RTS transmits its data. |
| RTS Received | It is the number of RTS packets received by Access Points |
| CTS Sent | It is the number of Clear to send (CTS) packets sent by Access Points |
| CTS Received | It is the number of CTS packets received by Wireless Nodes |
| Successful BackOff | It is the number of successful backoffs running at a wireless node. In the IEEE 802.11 Wireless Local Area Networks (WLANs), network nodes experiencing collisions on the shared channel need to BackOff for a random period of time, which is uniformly selected from the Contention Window (CW). BackOff is a timer which is decreased as long as the medium is sensed to be idle for a DIFS, and frozen when a transmission is detected on the medium, and resumed when the channel is detected as idle again for a DIFS interval. |
| Failed BackOff | It is the number of failed backoffs at wireless node |

### 3.1.26 Radio measurements log file

NetSim IEEE 802.11 Radio measurements csv log file records pathloss, shadowing loss, fading loss, transmitted power, received power, SNR, Interference Power, SINR, BER, NSS, MCS. This log can be enabled by clicking on configure reports in top ribbon ▷ Plots ▷ Select IEEE 802.11 Radio Measurements log under Network Logs.

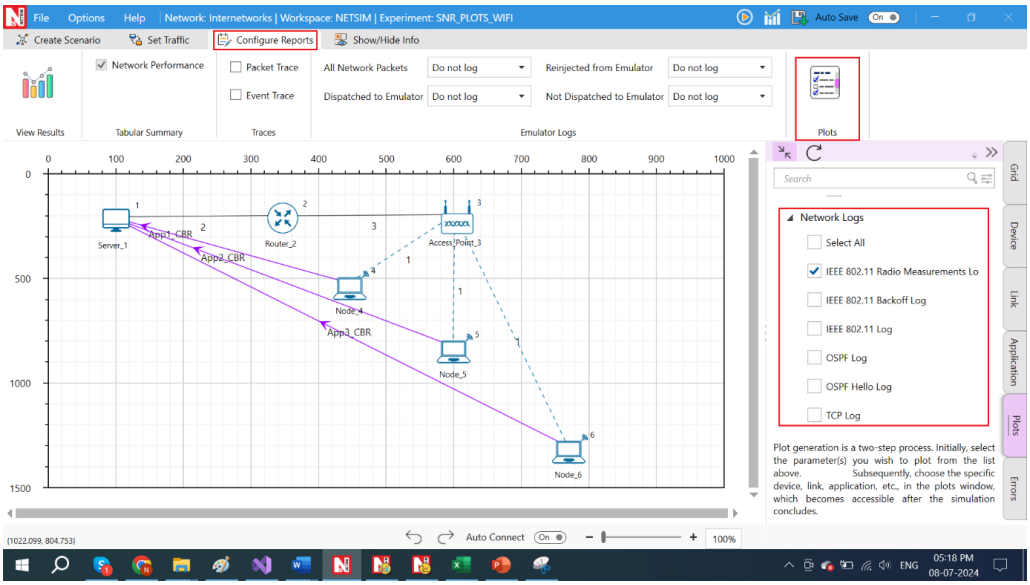



**Figure 3-9:** *Enabling IEEE 802.11 Radio Measurement log.*

The IEEE 802.11 Radio Measurement log.csv file will contain the following information:

- Time in Milliseconds
- Transmitter Name
- Receiver Name
- Distance between the Transmitter and the Receiver in meters
- Packet ID
- Packet Type
- Control Packet Type
- Transmitter Power in dBm
- Total Loss in dB
- Pathloss in dB
- Shadowing Loss in dB
- Fading Loss in dB
- Received Power in dBm
- Interference in dBm
- Tx Antenna Gain (dB)

- Rx Antenna Gain (dB)
- SNR in dB
- SINR in dB
- BER
- NSS
- MCS

The log file can be accessed from the Simulations Results Window under the Logs option.

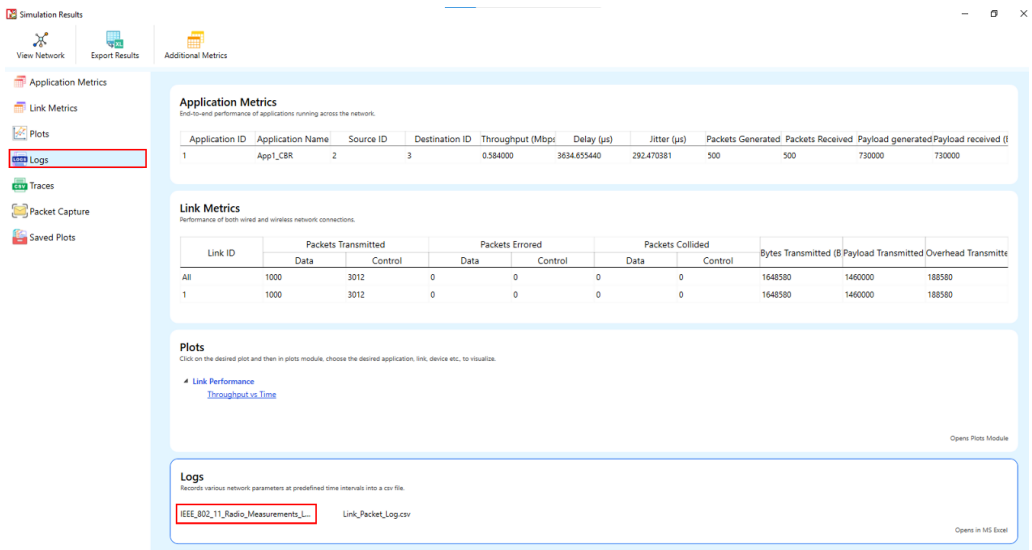



**Figure 3-10:** *IEEE 802.11 Radio Measurement log.csv file highlighted in the Results window.*

| Time(ms | Transmi | Receive | Distanc | Packet I | Packet T | Control | Tx_Pow | Path Los | Shadow | Fading L | Total Lo | Tx Ante | Rx Ante | Rx_Pow | Interfer | SNR(dB | SINR(dE | NSS | BER | MCS Inc |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.36401 | WIRELESS_ | ACCESS_P( | 159.9207 | 0 | Control_Pa | TCP_SYN | 20 | 84.16771 | 1.596765 | 0 | 85.76447 | 0 | 0 | -65.7645 | -1000 | 35.05318 | 35.05318 | 0 | 0 | 3 |
| 0.67802 | ACCESS_P( | WIRELESS_ | 159.9207 | 0 | Control_Pa | WLAN_ACK | 20 | 84.16771 | -12.4661 | 0 | 71.70165 | 0 | 0 | -51.7016 | -1000 | 49.11601 | 49.11601 | 0 | 0 | 0 |
| 1.18213 | ACCESS_P( | WIRELESS_ | 157.4915 | 0 | Control_Pa | TCP_SYN | 20 | 84.03476 | -10.7208 | 0 | 73.31394 | 0 | 0 | -53.3139 | -1000 | 47.50371 | 47.50371 | 0 | 0 | 3 |
| 1.49614 | WIRELESS_ | ACCESS_P( | 157.4915 | 0 | Control_Pa | WLAN_ACK | 20 | 84.03476 | -4.51921 | 0 | 79.51555 | 0 | 0 | -59.5155 | -1000 | 41.30211 | 41.30211 | 0 | 0 | 0 |
| 2.22025 | WIRELESS_ | ACCESS_P( | 157.4915 | 0 | Control_Pa | TCP_SYNAC | 20 | 84.03476 | -4.51921 | -4.49118 | 75.02437 | 0 | 0 | -55.0244 | -1000 | 45.79329 | 45.79329 | 0 | 0 | 3 |
| 2.53426 | ACCESS_P( | WIRELESS_ | 157.4915 | 0 | Control_Pa | WLAN_ACK | 20 | 84.03476 | -10.7208 | -0.88244 | 72.4315 | 0 | 0 | -52.4315 | -1000 | 48.38615 | 48.38615 | 0 | 0 | 0 |
| 3.03837 | ACCESS_P( | WIRELESS_ | 159.9207 | 0 | Control_Pa | TCP_SYNAC | 20 | 84.16771 | -12.4661 | -12.6359 | 59.06573 | 0 | 0 | -39.0657 | -1000 | 61.75193 | 61.75193 | 0 | 0 | 3 |
| 3.35238 | WIRELESS_ | ACCESS_P( | 159.9207 | 0 | Control_Pa | WLAN_ACK | 20 | 84.16771 | 1.596765 | -9.38104 | 76.38344 | 0 | 0 | -56.3834 | -1000 | 44.43422 | 44.43422 | 0 | 0 | 0 |
| 3.93349 | WIRELESS_ | ACCESS_P( | 159.9207 | 0 | Control_Pa | TCP_ACK | 20 | 84.16771 | 1.596765 | 0.131989 | 85.89646 | 0 | 0 | -65.8965 | -1000 | 34.9212 | 34.9212 | 0 | 0 | 3 |
| 4.2475 | ACCESS_P( | WIRELESS_ | 159.9207 | 0 | Control_Pa | WLAN_ACK | 20 | 84.16771 | -12.4661 | -2.9512 | 68.75045 | 0 | 0 | -48.7504 | -1000 | 52.06721 | 52.06721 | 0 | 0 | 0 |
| 4.66861 | ACCESS_P( | WIRELESS_ | 157.4915 | 0 | Control_Pa | TCP_ACK | 20 | 84.03476 | -10.7208 | -0.49198 | 72.82196 | 0 | 0 | -52.822 | -1000 | 47.99569 | 47.99569 | 0 | 0 | 3 |
| 4.98262 | WIRELESS_ | ACCESS_P( | 157.4915 | 0 | Control_Pa | WLAN_ACK | 20 | 84.03476 | -4.51921 | -8.99106 | 70.52448 | 0 | 0 | -50.5245 | -1000 | 50.29317 | 50.29317 | 0 | 0 | 0 |
| 6.52473 | WIRELESS_ | ACCESS_P( | 159.9207 | 1 | CBR | App1_CBR | 20 | 84.16771 | 1.596765 | 5.614471 | 91.37894 | 0 | 0 | -71.3789 | -1000 | 29.43871 | 29.43871 | 0 | 0 | 3 |
| 6.83874 | ACCESS_P( | WIRELESS_ | 159.9207 | 0 | Control_Pa | WLAN_ACK | 20 | 84.16771 | -12.4661 | 4.447078 | 76.14873 | 0 | 0 | -56.1487 | -1000 | 44.66893 | 44.66893 | 0 | 0 | 0 |
| 8.24085 | ACCESS_P( | WIRELESS_ | 157.4915 | 1 | CBR | App1_CBR | 20 | 84.03476 | -10.7208 | 1.168658 | 74.4826 | 0 | 0 | -54.4826 | -1000 | 46.33506 | 46.33506 | 0 | 0 | 3 |
| 8.55486 | WIRELESS_ | ACCESS_P( | 157.4915 | 0 | Control_Pa | WLAN_ACK | 20 | 84.03476 | -4.51921 | -3.55414 | 75.96141 | 0 | 0 | -55.9614 | -1000 | 44.85625 | 44.85625 | 0 | 0 | 0 |
| 8.97597 | WIRELESS_ | ACCESS_P( | 157.4915 | 0 | Control_Pa | TCP_ACK | 20 | 84.03476 | -4.51921 | 1.837273 | 81.35282 | 0 | 0 | -61.3528 | -1000 | 39.46484 | 39.46484 | 0 | 0 | 3 |
| 9.28998 | ACCESS_P( | WIRELESS_ | 157.4915 | 0 | Control_Pa | WLAN_ACK | 20 | 84.03476 | -10.7208 | -3.64217 | 69.67177 | 0 | 0 | -49.6718 | -1000 | 51.14589 | 51.14589 | 0 | 0 | 0 |
| 10.13109 | ACCESS_P( | WIRELESS_ | 159.9207 | 0 | Control_Pa | TCP_ACK | 20 | 84.16771 | -12.4661 | -8.49088 | 63.21077 | 0 | 0 | -43.2108 | -1000 | 57.60689 | 57.60689 | 0 | 0 | 3 |
| 10.4451 | WIRELESS_ | ACCESS_P( | 159.9207 | 0 | Control_Pa | WLAN_ACK | 20 | 84.16771 | 1.596765 | 2.243039 | 88.00751 | 0 | 0 | -68.0075 | -1000 | 32.81015 | 32.81015 | 0 | 0 | 0 |
| 21.70201 | WIRELESS_ | ACCESS_P( | 159.9207 | 2 | CBR | App1_CBR | 20 | 84.16771 | 1.596765 | -1.75432 | 84.01015 | 0 | 0 | -64.0101 | -1000 | 36.80751 | 36.80751 | 0 | 0 | 3 |
| 22.01602 | ACCESS_P( | WIRELESS_ | 159.9207 | 0 | Control_Pa | WLAN_ACK | 20 | 84.16771 | -12.4661 | 3.631315 | 75.33296 | 0 | 0 | -55.333 | -1000 | 45.48469 | 45.48469 | 0 | 0 | 0 |

**Figure 3-11:** *IEEE 802.11 Radio Measurement log.csv file.*

Implementation details and Assumptions:

- The log is written during each packet received at the physical layer (PHY_IN). Hence, the number of entries will be based on the number of packets received, by all nodes or specific nodes based on values present in the DEVICE ID LIST array.
- The MCS column displays the MCS index from the Phy parameters table in case of IEEE 802.11 n ac and ax. However, it displays the index value from the Phy parameters table

as nIndex-1 in case of DSSS based standards such as b and OFDM based standards such as a, g and p.

- The NSS value is currently set to the minimum of transmit and receive antenna counts in the same device. For example, if Transmitting Antennas is set to 2 and receiving Antennas is set to 8 then NSS is set to 2.

### 3.1.27 NetSim plots

NetSim also provides Wi-Fi radio measurement plots. Enabling the plots is explained in Section 2.4. Here is the plot showing the SNR vs. time for a scenario involving three wireless nodes connected to a single AP, with the wireless nodes placed at different distances.

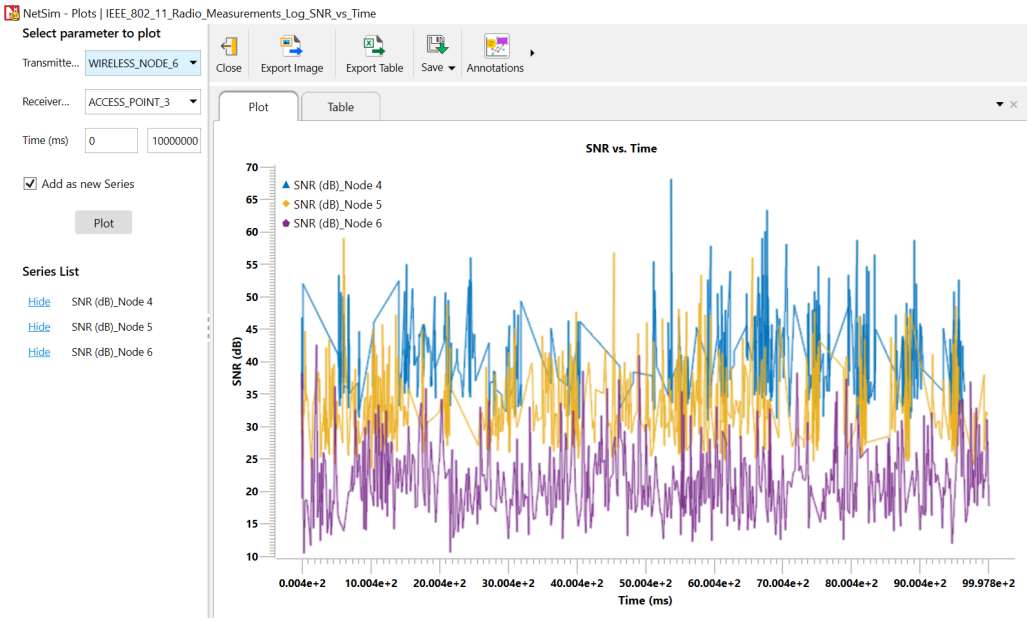



**Figure 3-12:** *Plot of SNR vs Time for Internetworks Wi-Fi scenario.*

### 3.1.28 IEEE 802.11 Backoff Log

NetSim 802.11 Backoff log file records details such as the Device name, Timestamp, Packet ID, BackoffTime, Contention Window Size and Retry Limit. This log can be used to understand the medium access mechanism in IEEE 802.11 Protocols.

This log file can be enabled in NetSim GUI by clicking on Plots tab ▷ Network Logs ▷ IEEE 802.11 Backoff Log.

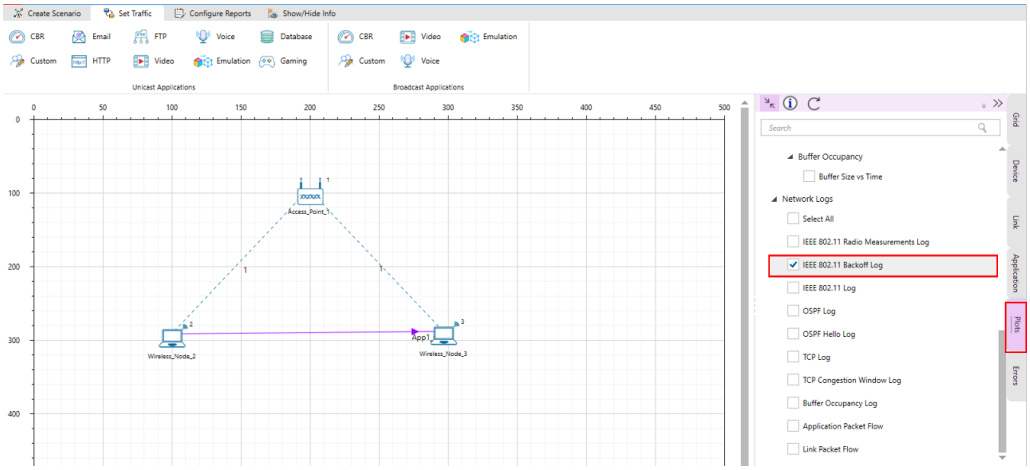



**Figure 3-13:** *Enabling IEEE 802.11 Backoff log.*

The IEEE802_11_Backoff_Log.csv file will contain the following information:

- Device Name
- Timestamp
- Packet ID
- BackOffTime
- Contention Window Size
- Retry Limit

The log file can be accessed from the Simulations Results Window under the Logs option in the left pane.

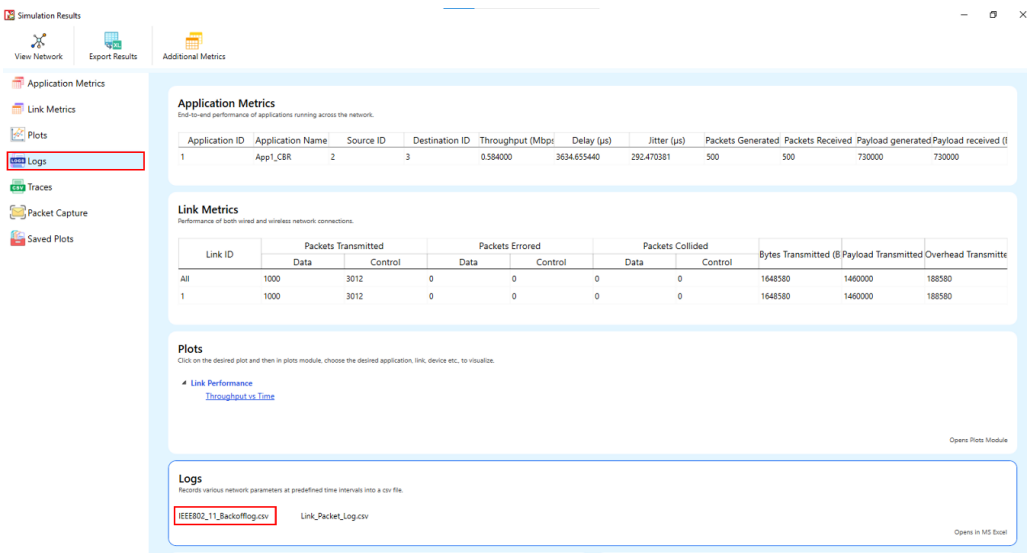



**Figure 3-14:** *IEEE802_11_Backoff_Log.csv file highlighted in the Results window.*

| Device Name | CurrentTime | PacketId | BackOffTime | CW | RetryCount |
|---|---|---|---|---|---|
| WIRELESS_NODE | 50 | 0 | 3 | 31 | 0 |
| ACCESS_POINT_ | 728.12 | 0 | 10 | 31 | 0 |
| WIRELESS_NODE | 1546.24 | 0 | 21 | 31 | 0 |
| ACCESS_POINT_ | 2584.36 | 0 | 10 | 31 | 0 |
| WIRELESS_NODE | 3402.48 | 0 | 14 | 31 | 0 |
| WIRELESS_NODE | 4297.6 | 1 | 15 | 31 | 0 |
| ACCESS_POINT_ | 4297.6 | 0 | 6 | 31 | 0 |
| WIRELESS_NODE | 5032.72 | 1 | 9 | 31 | 0 |
| ACCESS_POINT_ | 6888.84 | 1 | 2 | 31 | 0 |
| WIRELESS_NODE | 8604.96 | 0 | 6 | 31 | 0 |
| ACCESS_POINT_ | 9340.08 | 0 | 27 | 31 | 0 |
| WIRELESS_NODE | 20050 | 2 | 17 | 31 | 0 |
| ACCESS_POINT_ | 22066.12 | 2 | 13 | 31 | 0 |
| WIRELESS_NODE | 24002.24 | 0 | 23 | 31 | 0 |
| ACCESS_POINT_ | 25077.36 | 0 | 21 | 31 | 0 |
| WIRELESS_NODE | 40050 | 3 | 11 | 31 | 0 |
| ACCESS_POINT_ | 41946.12 | 3 | 4 | 31 | 0 |
| WIRELESS_NODE | 43702.24 | 0 | 24 | 31 | 0 |
| ACCESS_POINT_ | 44797.36 | 0 | 21 | 31 | 0 |
| WIRELESS_NODE | 60050 | 4 | 3 | 31 | 0 |
| ACCESS_POINT_ | 61786.12 | 4 | 14 | 31 | 0 |
| WIRELESS_NODE | 63742.24 | 0 | 12 | 31 | 0 |
| ACCESS_POINT_ | 64597.36 | 0 | 7 | 31 | 0 |
| WIRELESS_NODE | 80050 | 5 | 18 | 31 | 0 |

**Figure 3-15:** *IEEE802_11_Backoff_Log.csv file.*

## 3.2 Layer 2 (L2) Ethernet Switching

Layer 2 Switches have a MAC Address table that contains a MAC address and port number. Switches follow this simple algorithm for forwarding packets:

1. When a frame is received, the switch compares the SOURCE MAC address to the MAC address table. If the SOURCE is unknown, the switch adds it to the table along with the port number the packet was received on. In this way, the switch learns the MAC address and port of every transmitting device.
2. The switch then compares the DESTINATION MAC address with the table. If there is an entry, the switch forwards the frame out the associated port. If there is no entry, the switch sends the packet out all its ports, except the port that the frame was received on. This is termed as Flooding.
3. It does not learn the destination MAC until it receives a frame from that device.

### 3.2.1 Spanning Tree Protocol

NetSim Ethernet switches implement Spanning Tree Protocol to build a loop-free logical topology. This is always enabled and cannot be disabled.

### 3.2.2 Switch Port States

All switch ports in switches can be in one of the following states:

- **Blocking:** A port that would cause a switching loop if it were active. No user data is sent or received over a blocking port.
- **Listening:** The switch processes BPDUs and awaits possible new information that would cause it to return to the blocking state. It does not populate the MAC address table and it does not forward frames.
- **Learning:** While the port does not yet forward frames, it does learn source addresses from frames received and adds them to the filtering database (switching database). It populates the MAC address table but does not forward frames.
- **Forwarding:** A port receiving and sending data in Ethernet frames, normal operation.

It is recommended that the application start time be set to a value that is greater than the time it takes for the Spanning Tree Protocol to complete (of the order of 100s of milliseconds).

### 3.2.3 Model Limitations

1. The Spanning Tree Protocol is only run at the beginning of simulation. If a link fails, the Spanning Tree Protocol is not re-run.
2. If applications are started prior to completion of spanning tree protocol, then the MAC table created is not updated per the Spanning Tree Protocol.
3. Jumbo Frames are not supported in NetSim Ethernet Protocol.

### 3.2.4 Switch: GUI Parameters

Switch properties can be set by right clicking switch ▷ Properties ▷ Interface_1(ETHERNET).

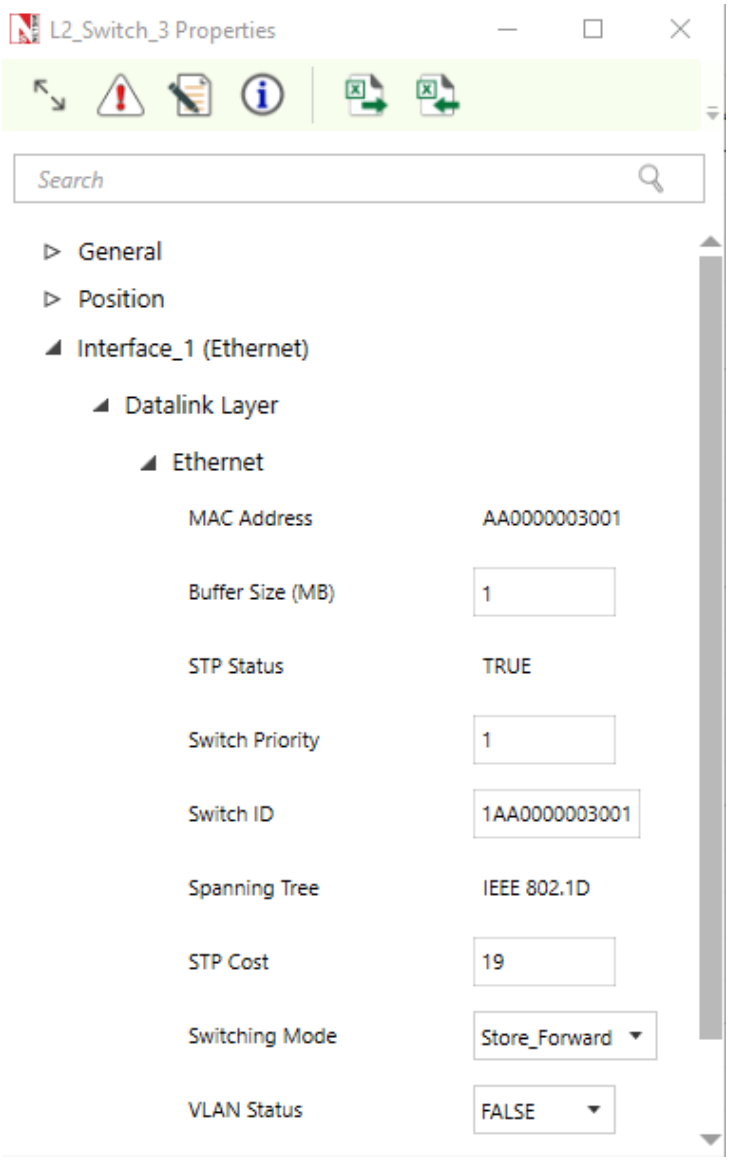



**Figure 3-16:** *Data Link Layer Properties of a Switch.*

The properties that can be set are:

**Table 3-21:** *Description of Datalink layer properties of switch parameter.*

| Parameter | Type | Range | Description |
|---|---|---|---|
| MAC ADDRESS | Fixed | Auto generated | The MAC address is a unique value associated with a network adapter. This is also known as hardware address or physical address. This is a 12-digit hexadecimal number (48 bits in length). |
| Buffer Size (MB) | Local | 1–5 | Buffer is the memory in a device which holds data packets temporarily. If the transmitting port is busy, incoming packets are stored in the buffer. NetSim models the buffer as an egress buffer and the range is 1 MB to 5MB per port of the switch. |

| Parameter | Type | Range | Description |
|---|---|---|---|
| STP Status | Fixed | TRUE | Spanning Tree Protocol is set to "True" in the Switches by default. |
| Switch Priority | Local | 1–61440 | This is the priority that can be assigned to the Switch. Priority is involved in deciding the root bridge for STP. |
| Switch ID | Local | – | Each switch has a unique ID for spanning tree calculation. The ID is derived by combining the priority and MAC address. Since a switch has a MAC address for each port, the least of the MAC addresses of the connected ports is taken while forming the unique ID. |
| Spanning Tree | Fixed | IEEE802.1D | The Spanning Tree Protocol (STP) ensures a loop-free topology for any bridged Ethernet local area network. The basic function of STP is to prevent bridge loops and the broadcast radiation that results from them. STP is standardized as IEEE 802.1D. As the name suggests, it creates a spanning tree within a network of connected layer-2 bridges (typically Ethernet switches) and disables those links that are not part of the spanning tree, leaving a single active path between any two network nodes. |
| STP Cost | Local | 0–1000 | Cost used by the switch to calculate spanning tree. The cost assigned to each port is based on its data rate. |
| Switching Mode | Local | Store Forward, Cut Through | Store and Forward: Forwarding takes place only after receipt of complete frame. This technique buffers the incoming frame and checks for errors. If no error is found it forwards the frame to the outgoing port, otherwise it discards the frame. Cut through: Switch forwards the incoming frames to its appropriate outgoing port immediately after receipt of destination address of the frame. |
| VLAN Status* | Local | TRUE, FALSE | To enable/disable VLAN |

*Requires license for Component 3 Advanced Routing and Switching.

## 3.3 Open Shortest Path First (OSPF v2) Routing Protocol

### 3.3.1 OSPF Overview

OSPF is a link-state routing protocol. It is designed to be run internally to a single Autonomous System. Each OSPF router maintains an identical database describing the Autonomous System's topology. From this database, a routing table is calculated by constructing a shortest-path tree.

OSPF routes IP packets based solely on the destination IP address found in the IP packet header. IP packets are routed "as is" – they are not encapsulated in any further protocol headers as they transit the Autonomous System. OSPF is a dynamic routing protocol. In NetSim, OSPF can detect topological changes in the AS (such as router interface failures) and calculate new loop-free routes after a period of convergence.

Each router maintains a database describing the Autonomous System's topology. This database is referred to as the link-state database. Each participating router has an identical database. Each individual piece of this database is a particular router's local state (e.g., the router's usable interfaces and reachable neighbors). The router distributes its local state throughout the Autonomous System by flooding.

All routers run the exact same algorithm, in parallel. From the link-state database, each router constructs a tree of shortest paths with itself as root. This shortest-path tree gives the route to each destination in the Autonomous System. The cost of a route is described by a single dimensionless metric.

### 3.3.2 OSPF Features

1. OSPF Messages – Hello, DD, LS Request, LS Update, LS Ack
2. Router LSA
3. The Neighbor Data structure features the following:
    - Link state request list
    - DB summary list
    - Link state re-transmission list
    - Link state send list
    - Link state retransmission timer
    - Inactivity timer
4. Routing table
5. Shortest path tree
6. The Interface data structure features:
    - Neighbor router list
    - Flood timer
    - Update LS list
    - Network LS timer
    - Delayed ack list
7. The Protocol data structure features:
    - Interface list
    - Area list
    - Max age removal timer
    - SPF timer
    - Routing table

8. The Area Data structure features:
    - Associated interface list
    - Router LSA list
    - Network LSA list
    - Router summary LSA list
    - Network summary LSA list
    - Max age list
    - Router LS timer
    - Shortest path list
9. The following can be logged during simulation:
    - Hello log
    - SPF log
    - Common log
    - Debug logs – LSDB, RXList, RLSA, RCVLSU, LSULIST, Route

### 3.3.3 Excluded Features

The following features in OSPF have not been implemented: Multiple Areas, Network LSA, Router summary LSA, Network summary LSA, Authentication, Equal cost multipath, External AS, External routing information, Interface type – Broadcast, NBMA, Virtual, Point to multipoint.

### 3.3.4 OSPF: GUI Parameters

OSPF properties can be set by right clicking on Router ▷ Properties ▷ Application layer.

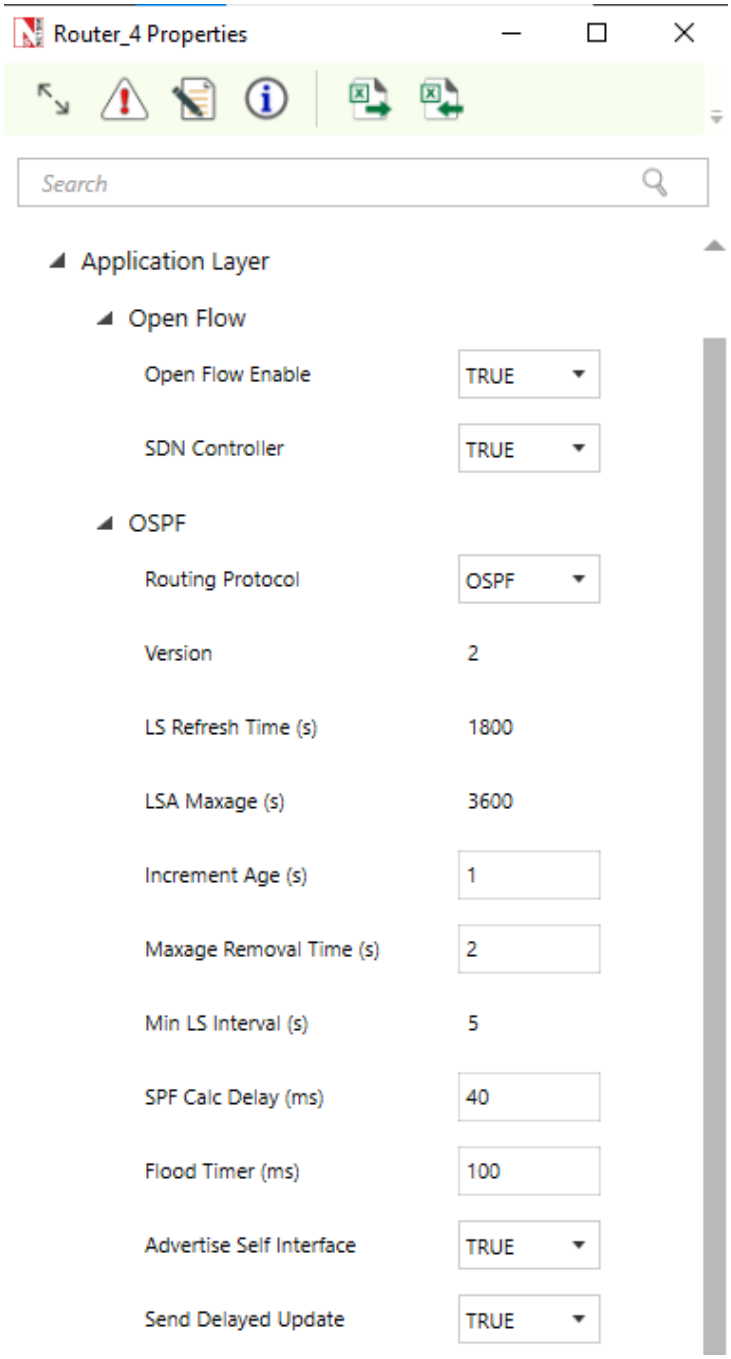



**Figure 3-17:** *Routing protocol properties of router.*

The properties that can be set are:

**Table 3-22:** *Description of Application layer Routing protocol properties.*

| Parameter | Type | Range | Description |
|---|---|---|---|
| Version | Global | Fixed | OSPF Version 2 as per RFC 2328 for IPv4. |

*Continued on next page*

| Parameter | Type | Range | Description |
|---|---|---|---|
| LSRefresh Time (s) | Global | Fixed | The maximum time between distinct originations of any particular Link State Advertisement (LSA). If the link state age field of one of the router's self-originated LSAs reaches the value LSRefreshTime, a new instance of the LSA is originated, even though the contents of the LSA (apart from the LSA header) will be the same. The value of LSRefreshTime is set to 30 minutes. |
| LSA Maxage (s) | Global | Fixed | The maximum age that an LSA can attain. When an LSA's LS age field reaches MaxAge, it is reflooded in an attempt to flush the LSA from the routing domain. LSAs of age MaxAge are not used in the routing table calculation. The default value of MaxAge is set to 1 hour or 3600s. |
| Increment Age (s) | Global | 0–100 | This is an internal variable of NetSim used for simulation purposes. This value decides how often to increase the age of the LSA in the OSPF LSA Lists. A small value will cause frequent updates and provide higher accuracy but may slow down simulation, and vice versa for a large value. |
| Maxage removal Time (s) | Global | 0–9999 | This variable decides the time when the LSA is removed from the MaxAgeLSA List. |
| Min LS Interval (s) | Global | Fixed | The minimum time between distinct originations of any particular LSA. The value of MinLSInterval is set to 5 seconds. |
| SPF Calc Delay (ms) | Global | 0–9999 | If SPF calculation is triggered, then the router will wait for this duration before starting the calculation. This can be used for the router to take multiple updates into account. |
| Flood Timer (ms) | Global | 0–9999 | The amount of time to wait before initializing the flood procedure. A random number between 0 to the set value will be chosen. The flood timer on/off is per the ISENDDELAYUPDATE variable setting. |
| Advertise Self Interface | Global | True/False | This is reserved for future use. As of NetSim v12, this should always be true. This will be used when a point-to-multipoint link is connected to the interface, and when such links are connected this should be set to false. |
| Send Delayed Update | Global | True/False | This variable can be set to true to delay sending the LSU. If set to true, then the delay would be per the flooding timer. Else the update is set immediately. |

*Global – Changes in all devices of similar type. Local – Only changes in current device.

## 3.4 Transmission Control Protocol (TCP)

### 3.4.1 TCP overview

TCP is a connection-oriented, end-to-end reliable protocol designed to fit into a layered hierarchy of protocols which support multi-network applications. The TCP provides for reliable communication between host computers and connected computer communication networks. Very few assumptions are made as to the reliability of the communication protocols below the TCP layer. TCP assumes it can obtain a simple, potentially unreliable datagram service from the lower-level protocols. In principle, the TCP should be able to operate above a wide spectrum of communication systems ranging from wired to wireless to mobile communication.

The TCP fits into a layered protocol architecture just above a basic Internet Protocol which provides a way for the TCP to send and receive variable-length segments of information enclosed in IP packets. The IP packet provides a means for addressing source and destination TCPs in different networks. The IP protocol also deals with any fragmentation or reassembly of the TCP segments required to achieve transport and delivery through multiple networks and interconnecting gateways.

**Table 3-23:** *Protocol Layering.*

| Layer |
|---|
| Application |
| TCP |
| IP |
| MAC |
| PHY |

### 3.4.2 TCP Features

The following features are implemented in TCP:

1. Three-way handshake (open/close)
2. Sequence Numbers
3. Slow start and congestion avoidance
4. Fast Retransmit/Fast Recovery
5. Selective Acknowledgement

### 3.4.3 Congestion Control Algorithms in TCP

`The following congestion control algorithms are supported in NetSim.`

1. Old Tahoe
2. Tahoe
3. Reno

4. New Reno
5. BIC
6. CUBIC

### 3.4.4 Limitations of TCP

1. Send and Receive buffers are infinite.

### 3.4.5 TCP: GUI parameters

The TCP parameters can be accessed by right clicking on a node and selecting Properties ▷ Transport Layer.

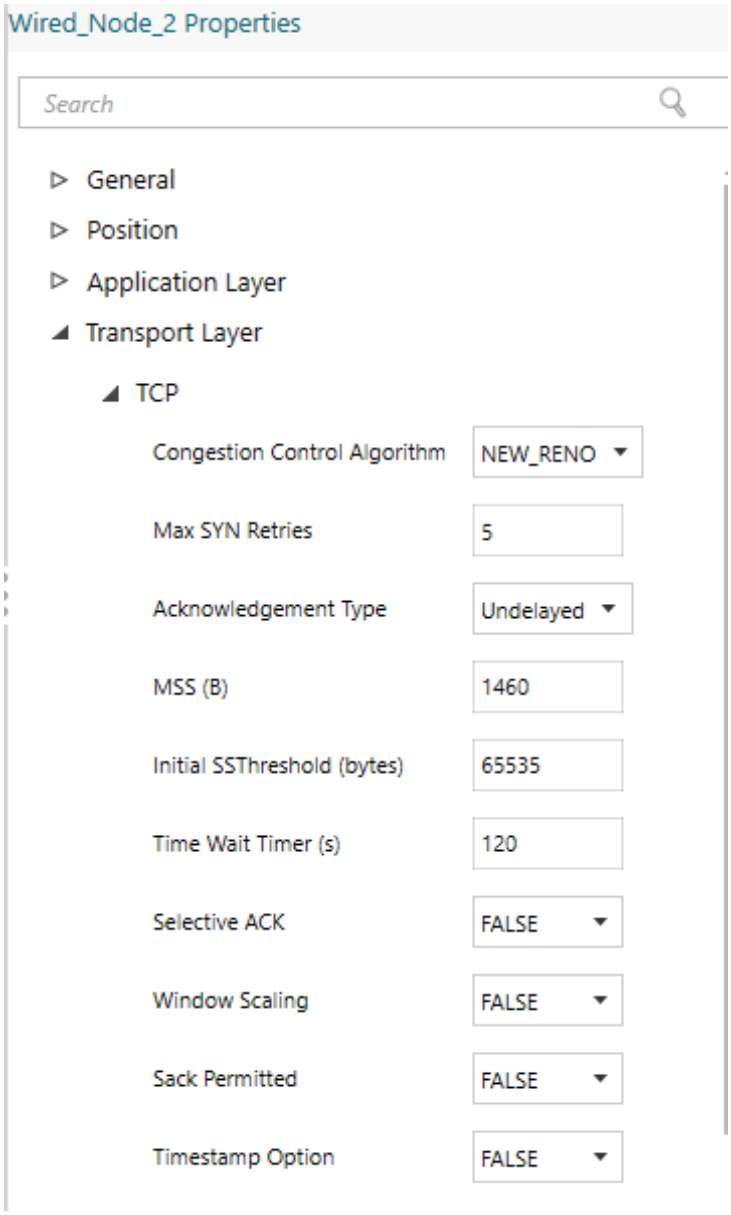



**Figure 3-18:** *Transport layer protocol properties of wired node.*

The properties that can be set are:

**Table 3-24:** *Description of Transport layer protocol properties.*

| Parameter | Type | Range | Description |
|---|---|---|---|
| Congestion Control Algorithm | Local | Old Tahoe, , Reno ,New Reno, Bic, Cubic | Congestion control algorithm is used to control the network congestion. Old Tahoe is the combination of slow start and congestion avoidance algorithm. The Fast-retransmit algorithm operating with Old Tahoe is known as the Tahoe. This algorithm works based on duplicate ack. When it receives three duplicate ack, which is the indication of segment loss, that segment will be retransmitted immediately without waiting for timeout. Reno implements fast recovery in case of three duplicate acknowledgements. New Reno improves retransmission during the fast-recovery phase of TCP Reno. BIC algorithm tries to find the maximum where to keep the window at for a long period of time, by using a binary search algorithm. CUBIC is an implementation of TCP with an optimized congestion control algorithm for high bandwidth networks with high latency. |
| Max SYN Retries | Local | 1–10 | Maximum number of TCP SYN ACK packets that can be retransmitted. The value should be in the range of 1 to 10. |
| Acknowledgement Type | Local | Delayed, Undelayed | If set to delayed, ACK response will be delayed improving network performance. If set to Un delayed, ACK will be sent immediately without delay. |
| MSS (bytes) | Local | 64–1460 | The maximum amount of data that a single message may contain. The MSS is the maximum data size and does not include the size of the header. MSS = MTU – (Network and Transport layer protocol headers). |
| Initial SSThreshold (bytes) | Local | 5840–65535 | The server-initial-ss-threshold should be in the range between 5840 and 65535 bytes. |
| Time Wait Timer (s) | Local | 30–240 | The Time wait timer default value is 120 seconds. The purpose of TIME-WAIT is to prevent delayed packets from one connection being accepted by a later connection. |

*Continued on next page*

| Parameter | Type | Range | Description |
|---|---|---|---|
| Selective ACK | Local | TRUE, FALSE | In Selective Acknowledgement (SACK) mechanism, the receiving TCP sends back SACK packets to the sender informing the sender of data that has been received. The sender can then retransmit only the missing data segments. |
| Window Scaling | Local | TRUE, FALSE | The TCP window scaling option is to increase the receive window size allowed in Transmission Control Protocol above its former maximum value of 65,535 bytes. |
| Sack Permitted | Local | TRUE, FALSE | The SACK-permitted option is offered to the remote end during TCP setup as an option to an opening SYN packet. The SACK option permits selective acknowledgment of permitted data. |
| Timestamp Option | Local | TRUE, FALSE | TCP is a symmetric protocol, allowing data to be sent at any time in either direction. Therefore, timestamp echoing may occur in either direction. For simplicity and symmetry, we specify that timestamps always be sent and echoed in both directions. For efficiency, we combine the timestamp and timestamp reply fields into a single TCP Timestamps Option. |

### 3.4.6 TCP Performance Metrics

The TCP Metrics table will be available in the Simulation Results dashboard if TCP is enabled in at least one device in the network. It provides the following information specific to TCP.

**Table 3-25:** *Parameter description of TCP Metrics table.*

| Parameter | Description |
|---|---|
| Source | It displays the name with ID of the source device which generates TCP packets |
| Destination | It displays the name with ID of the destination device which receives TCP packets |
| Local Address | It displays the local IP address with port number of the device present in source column |
| Remote Address | It represents the remote IP address with port number for the source and destination |
| Syn Sent | It is the number of syn packets sent by the source |
| Syn-Ack Sent | It is the number of syn ack packets sent by the destination |
| Segment Sent | It is the number of segments sent by a source |
| Segment Received | It is the number of segments received by a destination |
| Segment Retransmitted | It is the number of segments retransmitted by the source |

| Parameter | Description |
|---|---|
| Ack Sent | It is the number of acknowledgements sent by a source to destination in response to TCP syn ack and the number of acks sent by destination to source in response to the successful reception of data packet |
| Ack Received | It is the number of acknowledgements received by source in response to data packets and the number of acks received by destination in response to syn ack packet |
| Duplicate segment received | It is the number of duplicate segments received by destination |
| Out of order segment received | It is the number of out of order packets received by destination |
| Duplicate ack received | It is the number of duplicate acknowledgements received by source |
| Times RTO expired | It is the number of times RTO timer expired at source |

### 3.4.7 TCP Reference Documents

1. RFC 793: TRANSMISSION CONTROL PROTOCOL
2. RFC 1122: Requirements for Internet Hosts – Communication Layers
3. RFC 5681: TCP Congestion Control
4. RFC 3390: Increasing TCP's Initial Window
5. RFC 6298: Computing TCP's Retransmission Timer
6. RFC 2018: TCP Selective Acknowledgment Options
7. RFC 6582: The NewReno Modification to TCP's Fast Recovery Algorithm
8. RFC 6675: A Conservative Loss Recovery Algorithm Based on SACK for TCP
9. RFC 7323: TCP Extensions for High Performance
10. https://web.archive.org/web/20160505194415/http://netsrv.csc.ncsu.edu/export/cubic_a_new_tcp_2008.pdf
11. https://research.csc.ncsu.edu/netsrv/sites/default/files/bitcp.pdf
12. https://web.archive.org/web/20160528233754/http://netsrv.csc.ncsu.edu/export/hystart_techreport_2008.pdf

## 3.5 User Datagram Protocol (UDP)

### 3.5.1 UDP Overview

UDP (User Datagram Protocol) is a communication protocol that offers a limited amount of service when messages are exchanged between computers in a network that uses the Internet Protocol (IP). UDP uses the Internet Protocol to get a data unit (called a datagram) from one computer to another.

This protocol is transaction oriented, and delivery and duplicate protection are not guaranteed. Applications requiring reliable delivery of streams of data should use the Transmission Control Protocol (TCP).

### 3.5.2 UDP: GUI parameters

The UDP protocol can be set for an application by clicking on the Applications Transport Protocol option.

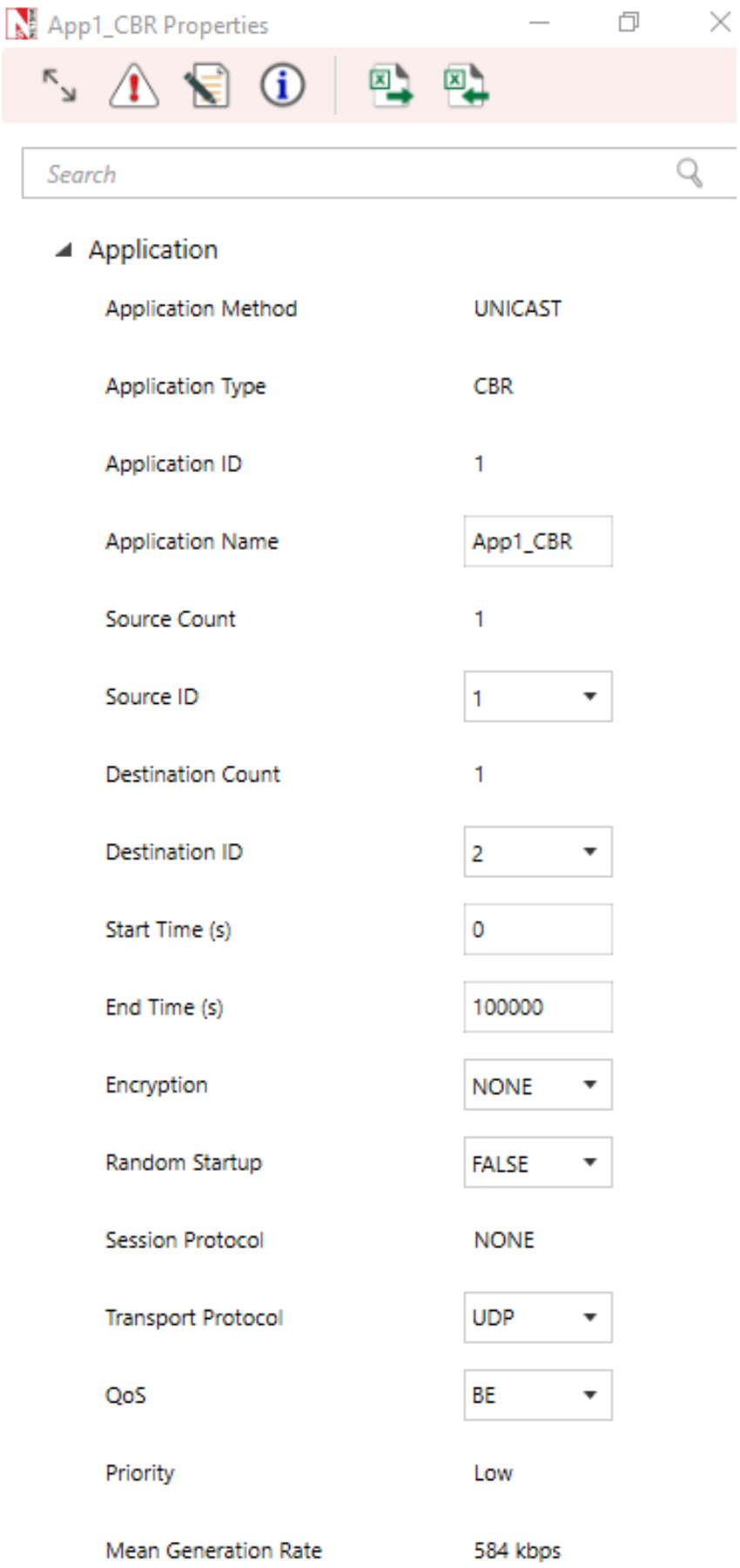



**Figure 3-19:** *Application configuration window.*

### 3.5.3 UDP Performance Metrics

The UDP Metrics table will be available in the Simulation Results dashboard if UDP is enabled in at least one device in the network. It provides the following information specific to UDP; see Table 3-26.

**Table 3-26:** *Parameter description of UDP Metrics table.*

| Parameter | Description |
|---|---|
| Device Id | It is the Id of a device in which UDP is enabled |
| Local Address | It represents the IP address with port number of the local device (either source or destination) |
| Foreign Address | It represents the IP address with port number of the remote device (either source or destination) |
| Datagram sent | It is the total number of datagrams sent from the source |
| Datagram received | It is the total number of datagrams received at the destination |

### 3.5.4 UDP Reference Documents

- RFC 768: User Datagram Protocol

## 3.6 IP Protocol

### 3.6.1 IP Performance Metrics

The IP Metrics table will be available in the Simulation Results dashboard if IP is enabled in at least one device in the network. It provides the following information specific to IP protocol:

**Table 3-27:** *Parameter description of IP Metrics table.*

| Parameter | Description |
|---|---|
| Device Id | It displays the Id of the Layer 3 devices |
| Packet sent | It is the number of packets sent by a source, intermediate devices (Router or L3 switch) |
| Packet forwarded | It is the number of packets forwarded by intermediate devices (Router or L3 switch) |
| Packet received | It is the number of data packets received by destination, intermediate devices (routing packets (OSPF, RIP etc.) received by Routers) |
| Packet discarded | It is the number of data packets that are discarded after their TTL value expires. |
| TTL expired | Time-to-live (TTL) is a value in an Internet Protocol (IP) packet that tells a network router whether or not the packet has been in the network too long and should be discarded |
| Firewall blocked | It is the number of packets blocked by firewall at routers |

## 3.7 Buffering, Queueing and Scheduling

### 3.7.1 Buffers

Devices and their Interfaces with buffers that support queuing and scheduling algorithms are:

1. Router (WAN – Network Layer)
2. EPC (WAN – Network Layer)
3. 6LOWPAN (WAN – Network Layer)
4. Satellite Gateway (WAN – Network Layer)

Queuing and scheduling in NetSim works as follows:

1. The scheduler schedules packet transmission from the head-of-queue per the scheduling algorithm. FIFO algorithm uses a single queue while Priority, RR and WFQ use 4 queues (1 queue for each priority).
2. The buffer size is a user input. This buffer is not split among the various queues. At any point in time the cumulative size of all queues is the buffer fill.
3. The way in which the individual queues are filled up, is per the queuing algorithm selected (implemented in version 12.1).

The buffer is an egress buffer. The buffer size in Megabytes (MB), for each interface mentioned above is a user input. The options: 8, 16, 32, 64, 128, 256, 512, 1024, 2048 and 4096 MB.

### 3.7.2 Queuing

**Drop Tail:** The queue is filled up till the buffer capacity. When the queue is full if any packet arrives, it is dropped. The buffer size is a user input.

**Random Early Detection (RED):**

1. The queue is filled up till the average queue size is equal to minimum threshold, without dropping any packet.
2. Random packets are dropped when the average queue size is between minimum threshold and maximum threshold. The number of packets being dropped depends on the Max Probability value.
3. All packets are dropped when the average queue size is above maximum threshold.

User Inputs: Maximum threshold, minimum threshold and maximum probability.

$$Avg = \frac{t_n}{t_{n+1}}(Avg - x_n) + x_n \tag{4}$$

where $Avg$ = Average Queue Size (initially 0), $t_n$ = Time when $n$th packet was added to the queue, $t_{n+1}$ = Current time when the $(n+1)$th packet is added, $x_n$ = Size of $n$th packet (B).

Packets are dropped if:

$$\text{No of Dropped Packets} > \frac{\text{Rand}(0,1)}{p} \quad \text{where } p = C_1 \times Avg + C_2$$

$$C_1 = \frac{\text{Max Probability}}{(\text{Max Threshold} - \text{Min Threshold})} \tag{5}$$

$$C_2 = \frac{\text{Max Probability}}{(\text{Max Threshold} - \text{Min Threshold})} \times \text{Min Threshold} \tag{6}$$

**Weighted Random Early Detection (WRED):**

Please refer to RED explained earlier. This is modified as follows:

1. There are different Max and Min threshold values for each type of priority, i.e. High, Medium, Normal, Low (the RED algorithm had only one set of Max and Min Threshold).
2. For the given threshold values of the packets, Random Early Detection (RED) algorithm is applied.

**Reference Documents:**

1. Sally Floyd, Van Jacobson (1993). Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking.

**Queue Size:** The queue depth can be obtained from the Event Trace or by modifying the protocol source code. To obtain it from the event trace, an MS Excel script would need to be written to filter by node, and at different points of time, add the number of APP-OUT events and subtract the number of TRANSPORT-OUT events. Note that deeper issues such as segmentation etc. will need to be handled appropriately based on the way the application and transport layer interact.

### 3.7.3 Scheduling

**First In First Out (FIFO):** Packets are scheduled according to their arrival time in the queue. Hence, first packet in queue is scheduled first.

**Priority:** NetSim supports 4 priority queues namely High, Medium, Normal and Low. With this scheduling, first all packets in the High priority queue are served, and then those in Medium, then in Normal, and finally those packets in the low priority queue. Note that this could lead to situations where only higher priority packets are served and lower priority packets are never served.

**Round Robin:** Packets from all the 4 priorities are served in circular order. When packets arrive, they are stored in the corresponding priority list.

**Weighted Fair Queuing (WFQ):** When packets arrive, they are stored in the corresponding list according to priority. Packets are served in order of maximum weight of the priority list. In NetSim WFQ is approximated as:

Weight = (Number of packets in Queue) × Priority where

$$\text{Priority} = 1, 2, 3 \text{ or } 4 \tag{7}$$

1 – Low priority, 2 – Normal, 3 – Medium, 4 – High.

**Early Deadline First (EDF):** Packets are added in the queue as they arrive. While dequeuing, the packets with earliest deadline are served first. The packets which have exceeded deadline are dropped.

$$\text{Deadline} = \text{Max Latency} - \text{Packet Creation Time} \tag{8}$$

Max Latency with respect to quality of service (QoS) of the packet is a user input.

## 3.8 Links

### 3.8.1 Modeling Error in Wired Links

The error rates in NetSim wired links are based on a standard error measurement unit called BER or Bit Error Rate. BER represents the ratio of errored bits to total bits.

The BER value can be set by the user. A typical value of BER, say $1 \times 10^{-6}$, which equals 0.000001, means that 1 bit is in error for every one-million bits transmitted. It is important to note that Bit Error Rate is not equal to Packet error rate (PER).

$$PER = 1 - (1 - BER)^L \quad \text{where } L \text{ is the packet length in bits} \tag{9}$$

For BER values less than 0.001, this is mathematically approximated in NetSim as:

$$PER = BER \times L \tag{10}$$

## 3.9 IP Addressing in NetSim

When you create a network using the GUI, NetSim will automatically configure the IP address to the devices in the scenario. Consider the following scenarios:

If you create a network with two wired nodes and L2_Switch, the IP addresses are assigned as 192.168.0.2 and 192.168.0.3 for the two wired nodes. The default subnet mask is assigned to be 255.255.0.0. It can be edited to 255.0.0.0 (Class A) or 255.255.255.0 (Class C) subnet masks. Both the nodes are in the same network (192.168.0.0).

Similarly, if you create a network with a router and two wired nodes, the IP addresses are assigned as 192.168.0.1 and 192.169.0.1 for the two wired nodes. The subnet mask is the default as in the above case, i.e., 255.255.0.0. The IP address of the router is 192.168.0.1 and 192.169.0.1 respectively for the two interfaces. Both the nodes are in different networks (192.168.0.0 and 192.169.0.0) in this case.

The same logic is extended as the number of devices increases.

# 4 Featured Examples

Sample configuration files for all networks are available in the Examples Menu in NetSim Home Screen. These files provide examples on how NetSim can be used – the parameters that can be changed and the typical effect it has on performance.

## 4.1 802.11n MIMO

Open NetSim, Select Examples ▷ Internetworks ▷ Wi-Fi ▷ 802.11n-MIMO then click on the tile in the middle panel to load the example.

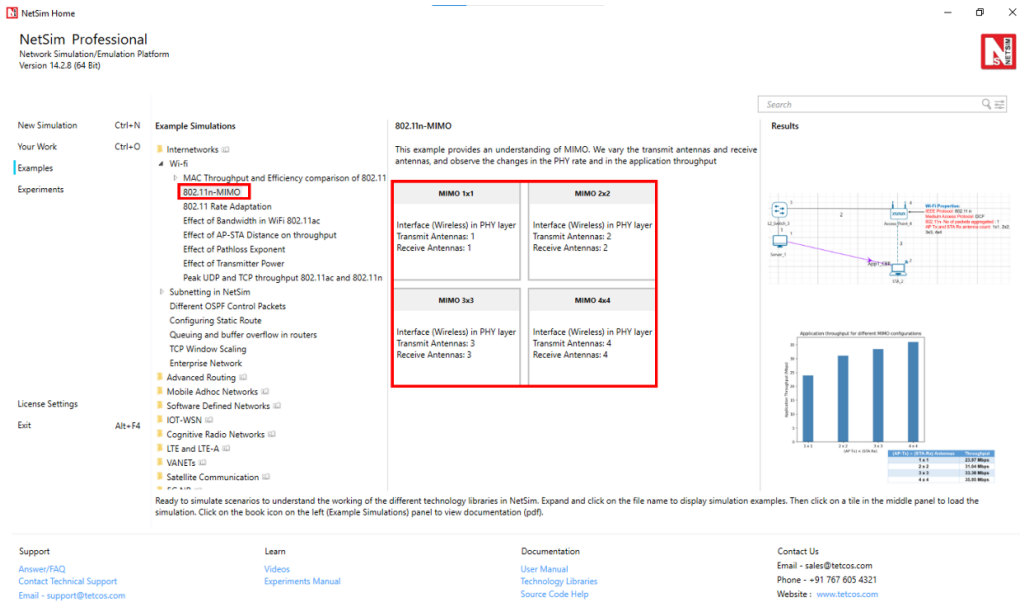



**Figure 4-1:** *List of scenarios for the example of 802.11n-MIMO.*

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file for 802.11n-MIMO.

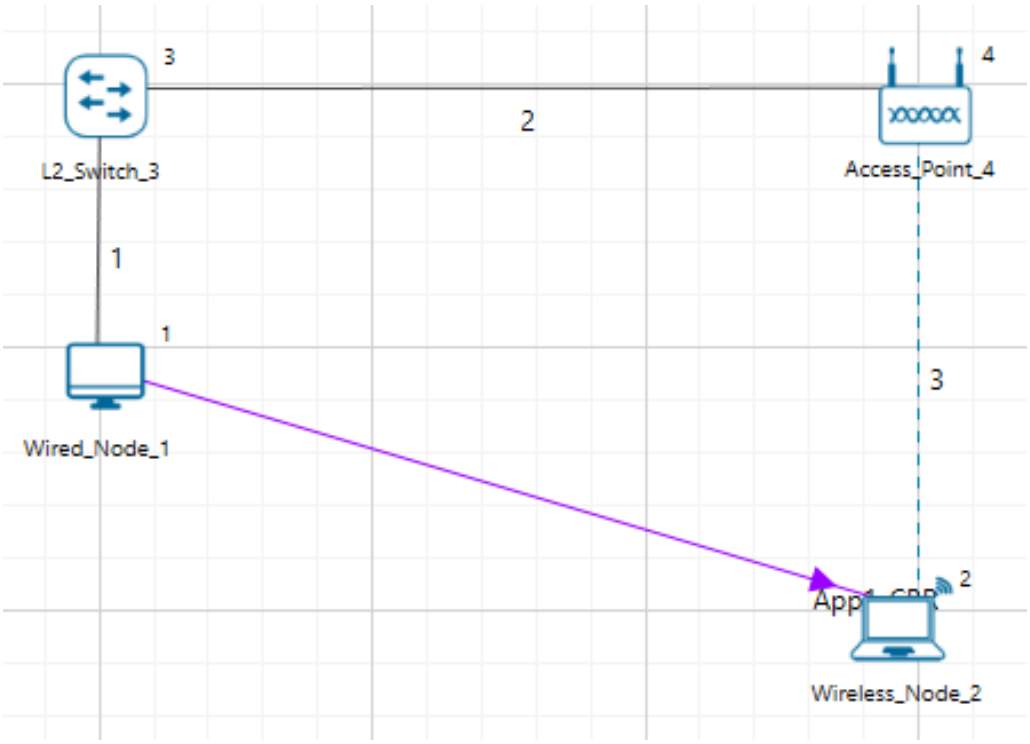



**Figure 4-2:** *Network setup for studying the 802.11n-MIMO.*

### Network Settings

1. Set grid length as 60m × 30m from the grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Distance between AP and Wireless node is 20m.
3. Set DCF as the medium access layer protocol under datalink layer properties of access point and wireless node (STA). 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 below steps.
4. In Interface (Wireless) properties of physical layer, WLAN Standard is set to 802.11n and No. of Tx and Rx Antennas is set to 1 in both Access Point and STA.
5. Click on wireless link and expand property panel on the right and set the Channel Characteristics: Pathloss, Path Loss Model: Log Distance, and Path Loss Exponent: 3 (Wireless Link Properties).
6. Configure CBR application between server and STA by clicking on the set traffic tab from the ribbon at the top. Click on created application and expand the application property panel on the right and set transport protocol to UDP, packet size to 1460 B and Inter arrival time as 233 $\mu$s.
7. Simulate for 10 sec and check the throughput.
8. Go back to the scenario and increase the Number of Tx and Rx Antenna $1 \times 1$, $2 \times 2$, $3 \times 3$, $4 \times 4$ respectively and check the throughput in the results window.
9. Results and Discussion:

**Table 4-1:** *Number of Tx and Rx Antenna vs. Throughput.*

| Number of Tx and Rx Antenna | Throughput |
|---|---|
| $1 \times 1$ | 23.97 Mbps |
| $2 \times 2$ | 31.04 Mbps |
| $3 \times 3$ | 33.38 Mbps |
| $4 \times 4$ | 35.95 Mbps |

MIMO is a method for multiplying the capacity of a radio link using multiple transmit and receive antennas. Increasing the Transmitting Antennas and Receiving Antennas increases PHY Data rate (link capacity) and hence leads to an increase in application throughput.

**Plot**

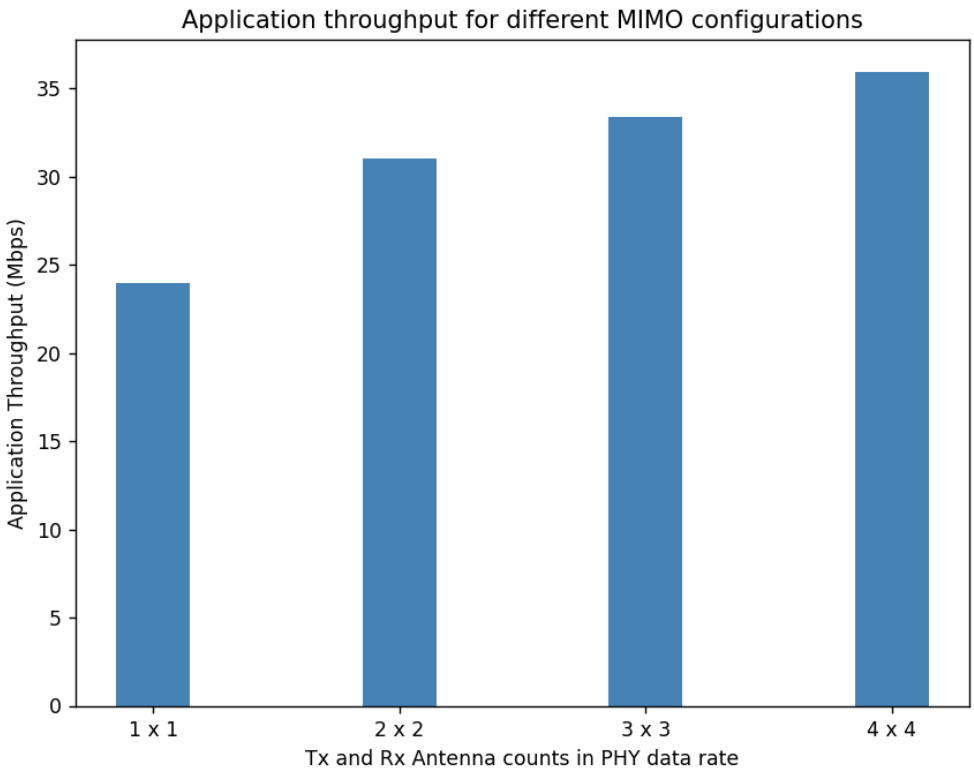



**Figure 4-3:** *Plot of Tx and Rx Antenna counts vs Throughput.*

## 4.2 Effect of Bandwidth in Wi-Fi 802.11ac

### 4.2.1 Effect of Bandwidth

Open NetSim and Select Examples ▹ Internetworks ▹ Wi-Fi ▹ Effect of bandwidth in Wi-Fi 802.11ac then click on the tile in the middle panel to load the example.

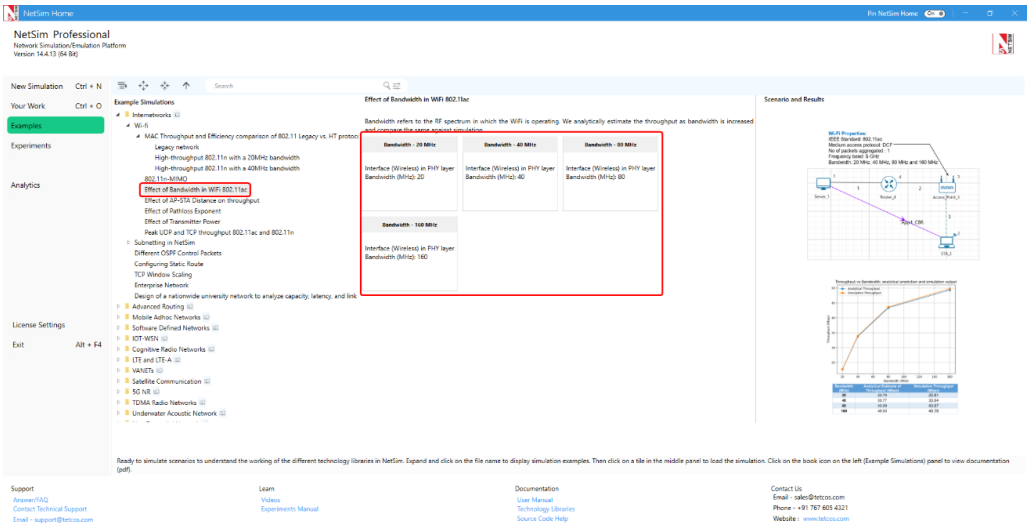



**Figure 4-4:** *List of scenarios for the example of effect of bandwidth in Wi-Fi 802.11ac.*

Figure 4-4 shows the four bandwidth cases available in this example, namely 20 MHz, 40 MHz, 80 MHz and 160 MHz.

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file as shown in Figure 4-5.

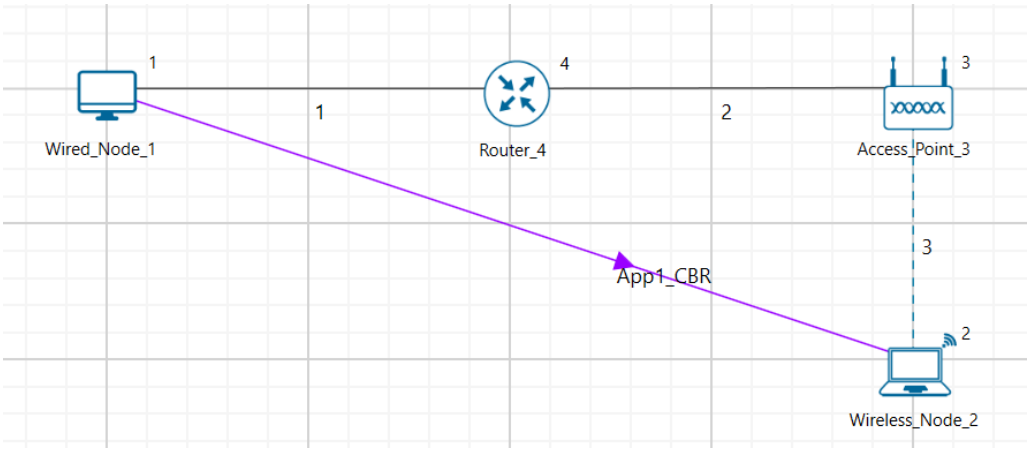



**Figure 4-5:** *Network setup for studying the effect of bandwidth in Wi-Fi 802.11ac.*

### Network Settings

1. Set grid length to 60m × 30m from the grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Set 802.11ac standard and Bandwidth to 20MHz under Wireless Interface ▷ Physical Layer properties of the access point and wireless node. 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 below steps.
3. Set DCF as the medium access layer protocol under Wireless Interface ▷ Datalink layer properties of access point and wireless node.
4. Set transmitter power as 40mW under Wireless Interface ▷ Transmitter Power properties of the access point and wireless node.
5. Click on wireless link and open property panel on the right and set the Channel Characteristics: No pathloss in wireless link properties.
6. Similarly, set Bit Error rate and Propagation delay to zero under wired link properties.
7. Configure CBR application between server and STA by clicking on the set traffic tab from the ribbon at the top. Click on created application and expand the application property panel on the right and set transport protocol to UDP, packet size to 1460 B and Inter arrival time as 116 $\mu$s (Generation rate is 100.6 Mbps).
8. Generation rate is set to 100 Mbps in application properties (Packet Size = 1460 Bytes, Interarrival time = 116.8 microseconds). Generation rate can be calculated by using the formula below:

$$\text{Generation Rate (Mbps)} = \text{Packet Size (Bytes)} \times \frac{8}{\text{Interarrival time } (\mu\text{s})} \tag{11}$$

$$= 1460 \times 8/116 \approx 100 \text{ Mbps}$$

9. Run simulation for 10s and see application throughput in the results window.
10. Go back to the scenario and increase the Bandwidth 20 to 40, 80, 160 respectively and check the throughput in the results window.

### Analytical Model

The average time to transmit a packet consists of:

- DIFS
- Backoff duration
- Data packet transmission time
- SIFS
- MAC ACK transmission time

The timing diagram is as shown below:

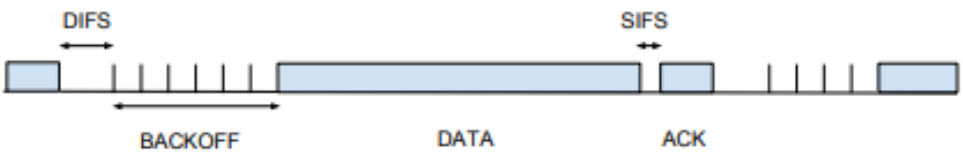



**Figure 4-6:** *Timing diagram for WLAN.*

The Average throughput can be calculated by using the formula below:

$$\text{Average Throughput (Mbps)} = \frac{\text{Application Payload (Bytes)}}{\text{Average Time per Packet } (\mu s)} \tag{12}$$

$$\text{Average time per packet } (\mu s) = \text{DIFS} + \text{Avg Backoff} + \text{Packet Tx Time} + \text{SIFS} + \text{ACK Tx Time} \tag{13}$$

$$\text{Packet Tx Time } (\mu s) = \text{Preamble time} + \frac{\text{MPDU Size}}{\text{Data rate}} \tag{14}$$

$$\text{Avg Backoff } (\mu s) = \frac{\text{CWmin}}{2} \times \text{Slot Time} \tag{15}$$

$$\text{ACK Tx Time } (\mu s) = \text{Preamble time} + \frac{\text{ACK Packet size}}{\text{ACK data rate}} \tag{16}$$

$$\text{DIFS } (\mu s) = \text{SIFS} + 2 \times \text{Slot Time} \tag{17}$$

Where:

$$\text{Application payload} = 1460 \text{ Bytes}$$
$$\text{Average time per packet} = 34 + 67.5 + 185.36 + 16 + 212.88 = 513.74\ \mu\text{s}$$
$$\text{SIFS} = 16\ \mu\text{s}, \quad \text{Slot time} = 9\ \mu\text{s}, \quad \text{CWmin} = 15 \text{ slots for 802.11ac}$$
$$\text{DIFS} = 16 + 2 \times 9 = 34\ \mu\text{s}, \quad \text{Avg Backoff} = 67.5\ \mu\text{s}$$
$$\text{Packet Tx Time} = 44 + (1532 \times 8/86.7) = 185.36\ \mu\text{s}$$
$$\text{Preamble time} = 44\ \mu\text{s for 802.11ac}$$
$$\text{MPDU Size} = 1460 + 8 + 20 + 44 = 1532 \text{ Bytes}$$
$$\text{ACK Tx Time} = 44 + (152 \times 8/7.2) = 212.88\ \mu\text{s}$$
$$\text{Average throughput} = 1460 \times 8/513.74 = 22.7 \text{ Mbps}$$

Similarly calculate throughput theoretically for other samples by changing bandwidth and compare with simulation throughput. Users can get the data rate by using the formula given below:

$$\text{PHY data rate (802.11ac)} = \frac{\text{PHY_layer_payload} \times 8}{(\text{phy end time} - \text{phy arrival time} - 44)} \tag{18}$$

### Results and Discussion

**Table 4-2:** *Result comparison of different bandwidth vs. Analytical Estimate of Throughput and Simulation Throughput.*

| Bandwidth (MHz) | Analytical Estimate (Mbps) | Simulation (Mbps) |
|---|---|---|
| 20 | 22.70 | 22.81 |
| 40 | 33.77 | 33.94 |
| 80 | 43.39 | 43.67 |
| 160 | 49.35 | 49.78 |

One can observe that there is an increase in throughput as we increase the bandwidth from 20MHz to 160MHz.

### Plot

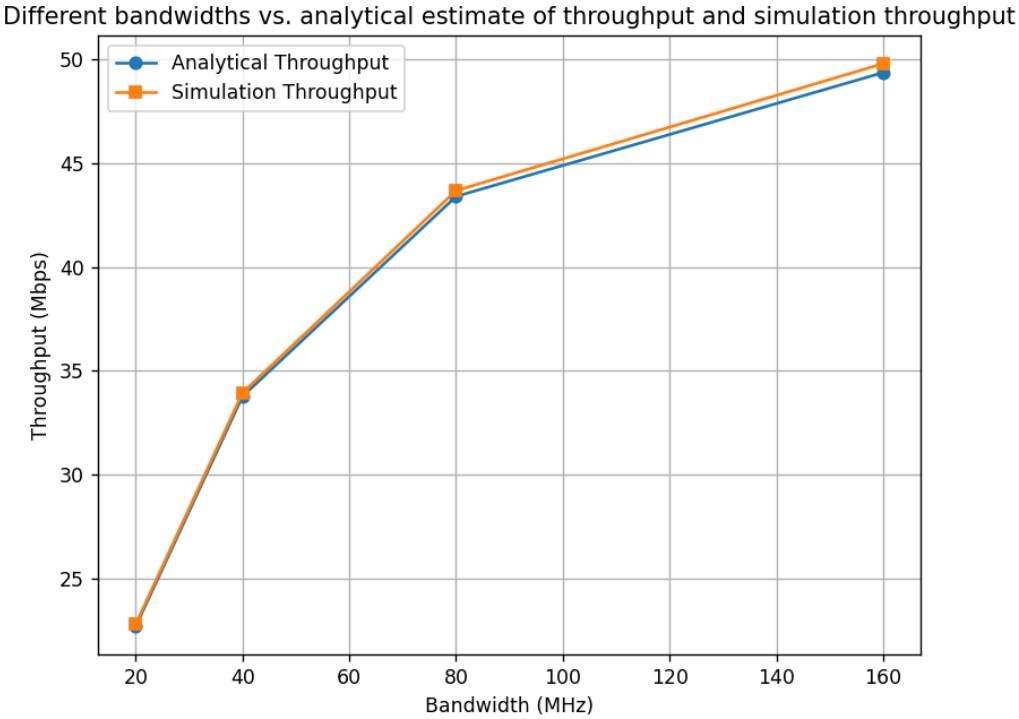



**Figure 4-7:** *Plot of Bandwidth vs Throughput.*

## 4.3 Factors affecting WLAN PHY Rate

The examples explained in this section focus on the factors which affect the PHY Rate/Link Throughput of 802.11 based networks:

- Transmitter power (More Tx power leads to higher throughput)
- Channel Path loss (Higher path loss exponent leads to lower throughput)
- Receiver sensitivity (Lower Rx sensitivity leads to higher throughput)
- Distance (Higher distance between nodes leads to lower throughput)

### 4.3.1 Effect of AP-STA Distance on throughput

Open NetSim and Select Examples ▷ Internetworks ▷ Wi-Fi ▷ Effect of AP STA Distance on throughput then click on the tile in the middle panel to load the example as shown in Figure 4-8.

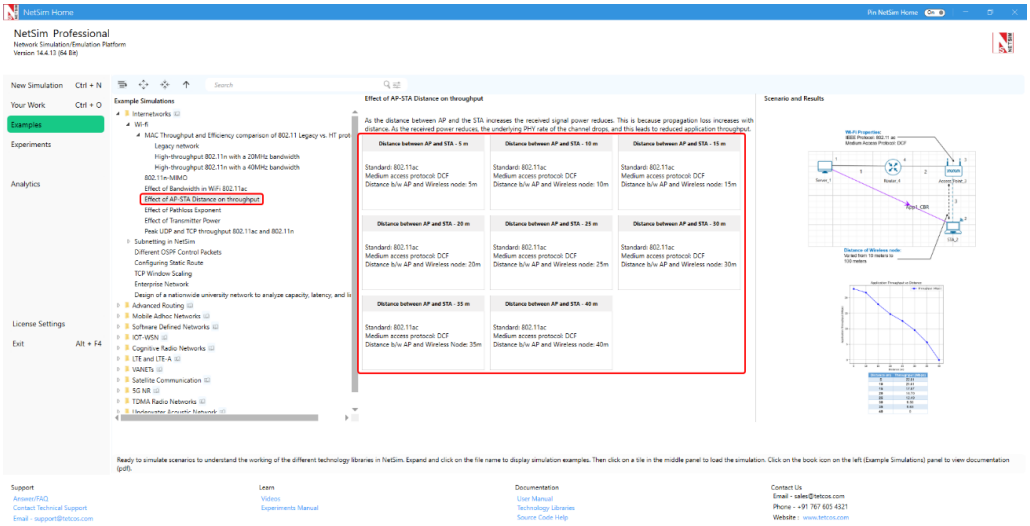



**Figure 4-8:** *List of scenarios for the example effect of AP-STA distance on throughput.*

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file see Figure 4-9.

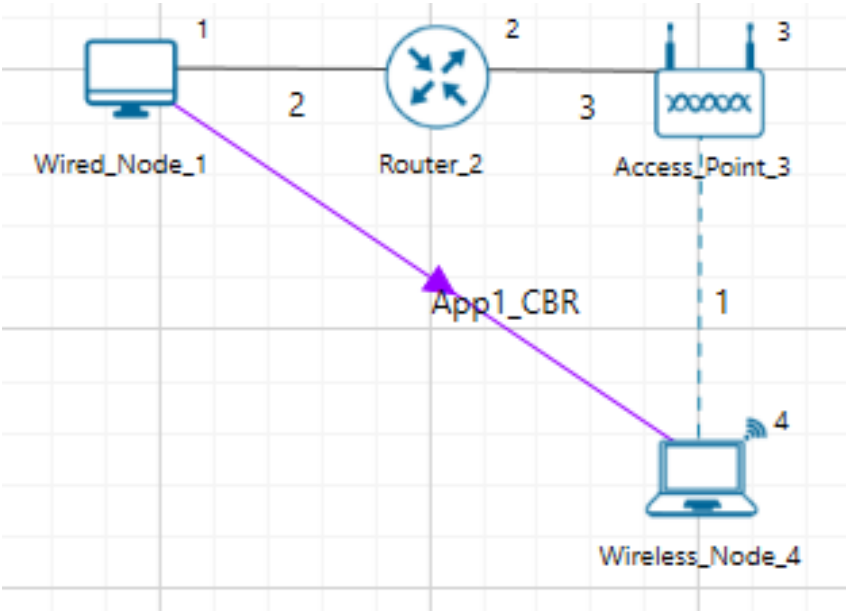



**Figure 4-9:** *Network setup for studying the Effect of AP-STA Distance on throughput.*

As the distance between two devices increases the received signal power reduces as propagation loss increases with distance. As the received power reduces, the underlying PHY rate of the channel drops.

### Network Settings

1. Set grid length to 120m × 60m from the grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Distance between Access Point and the Wireless Node is set to 5m.

3. Set DCF as the medium access layer protocol under datalink layer properties of access point and wireless node. 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 below steps.

4. WLAN Standard is set to 802.11ac and No. of Tx and Rx Antenna is set to 1 in access point and No. of Tx is 1 and Rx Antenna is set to 1 in wireless node (Right-Click Access Point or Wireless Node ▷ Properties ▷ Interface Wireless ▷ Transmitting Antennas and Receiving Antennas) and Bandwidth is set to 20 MHz in both Access-point and wireless-node. Transmitter Power set to 100mW in both Access-point and wireless-node.

5. Wired Link speed is set to 1Gbps and propagation delay to 10 $\mu$s in wired links.

6. Channel Characteristics: Pathloss, Path loss model: Log distance, Path loss exponent: 3.5.

7. Configure an application between any two nodes by selecting a CBR application from the Set Traffic tab in the ribbon on the top. Click on created application and expand the application property panel on the right and set transport protocol to UDP, packet size to 1460 B and Inter arrival time to 116.8 $\mu$s.

8. Application Generation Rate: 100 Mbps (Packet Size: 1460, Inter Arrival Time: 116.8 $\mu$s).

9. Run the simulation for 10s.

10. Go back to the scenario and increase the distance as per result table and run simulation for 10s.

### Results

**Table 4-3:** *Result comparison of different distance vs. throughput.*

| Distance (m) | Throughput (Mbps) |
|---|---|
| 5 | 22.81 |
| 10 | 21.61 |
| 15 | 17.87 |
| 20 | 14.70 |
| 25 | 12.49 |
| 30 | 9.56 |
| 35 | 5.63 |
| 40 | 0 |

### Plot

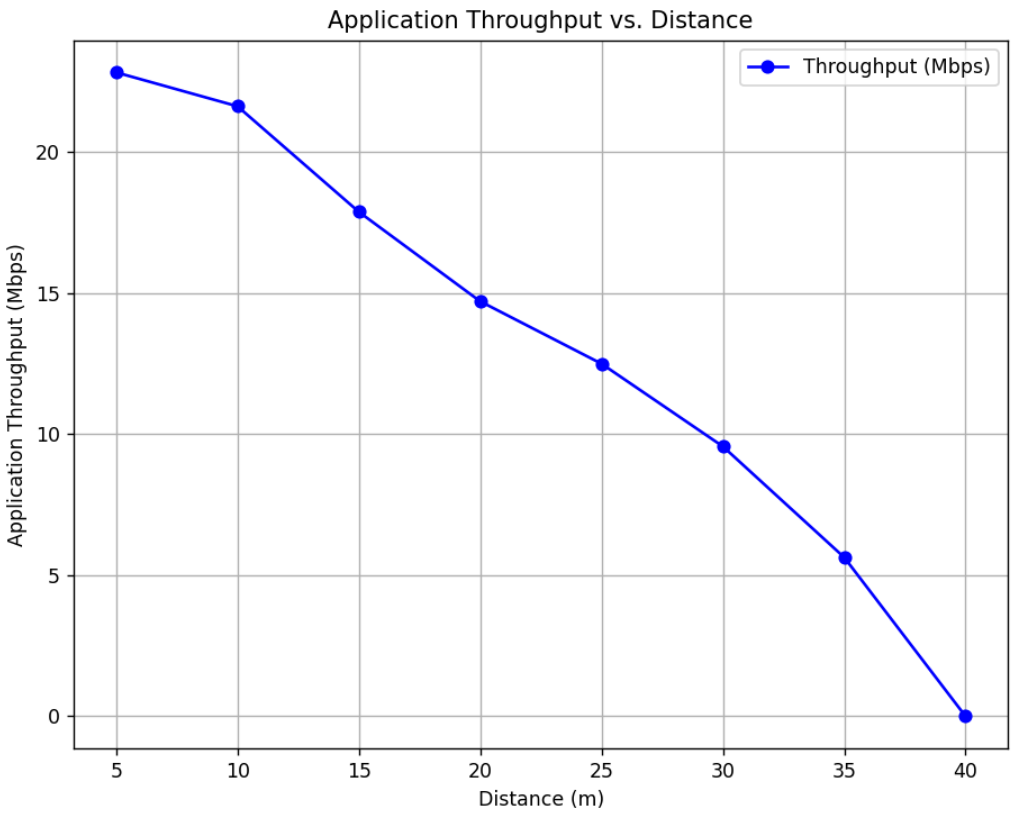



**Figure 4-10:** *Plot of Distance vs Throughput.*

### 4.3.2 Effect of Pathloss Exponent

Open NetSim and Select Examples ▷ Internetworks ▷ Wi-Fi ▷ Effect of Pathloss Exponent then click on the tile in the middle panel to load the example as shown in Figure 4-11.

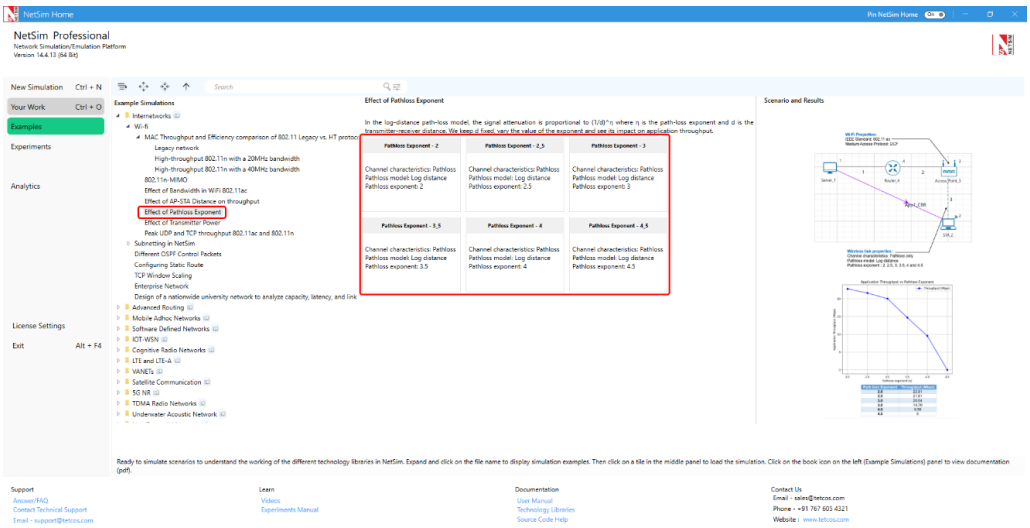


**Figure 4-11:** *List of scenarios for the example of effect of pathloss exponent.*

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file as shown Figure 4-12.

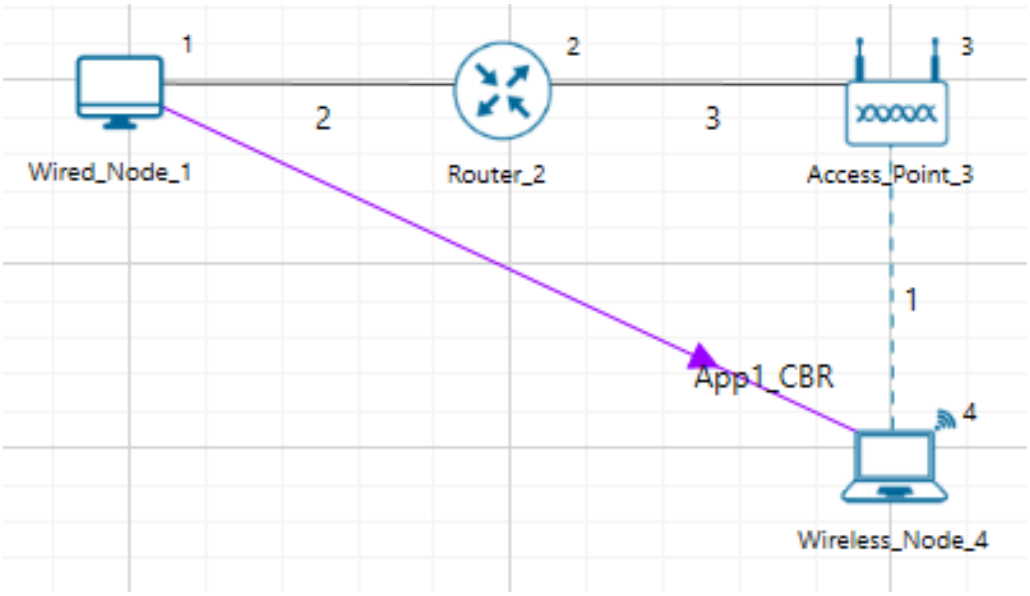



**Figure 4-12:** *Network setup for studying the effect of pathloss exponent.*

Path Loss or Attenuation of RF signals occurs naturally with distance. Losses can be increased by increasing the path loss exponent ($\eta$). This option is available in channel characteristics. Users can compare the results by changing the path loss exponent ($\eta$) value.

### Network Settings

1. Set grid length to 80m × 40m from the grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Distance between Access Point and the Wireless Node is set to 20m.
3. Set DCF as the medium access layer protocol under datalink layer properties of access point and wireless node. 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 below steps.
4. WLAN Standard is set to 802.11ac and No. of Tx and Rx Antenna is set to 1 in both access point and wireless node and Bandwidth is set to 20 MHz in both Access-point and wireless-node and Transmitter Power set to 100mW in both Access-point and wireless-node.
5. Click on the wireless link and open property panel on the right and set the Channel Characteristics: Path Loss Only, Path Loss Model: Log Distance, Path Loss Exponent: 2.
6. Configure an application between any two nodes by selecting a CBR application from the Set Traffic tab in the ribbon on the top. Click on created application and expand the application property panel on the right and set transport protocol to UDP, packet size to 1460 B and Inter arrival time to 116 $\mu$s.
7. Run simulation for 10s.
8. Go back to the scenario and increase the Path Loss Exponent from 2 to 2.5, 3, 3.5, 4, and 4.5 respectively and Run simulation for 10s.

### Results

**Table 4-4:** *Result comparison of different pathloss exponent value vs. throughput.*

| Path loss Exponent | Throughput (Mbps) |
|---|---|
| 2.0 | 22.81 |
| 2.5 | 21.61 |
| 3.0 | 20.04 |
| 3.5 | 14.70 |
| 4.0 | 9.56 |
| 4.5 | 0 |

### Plot

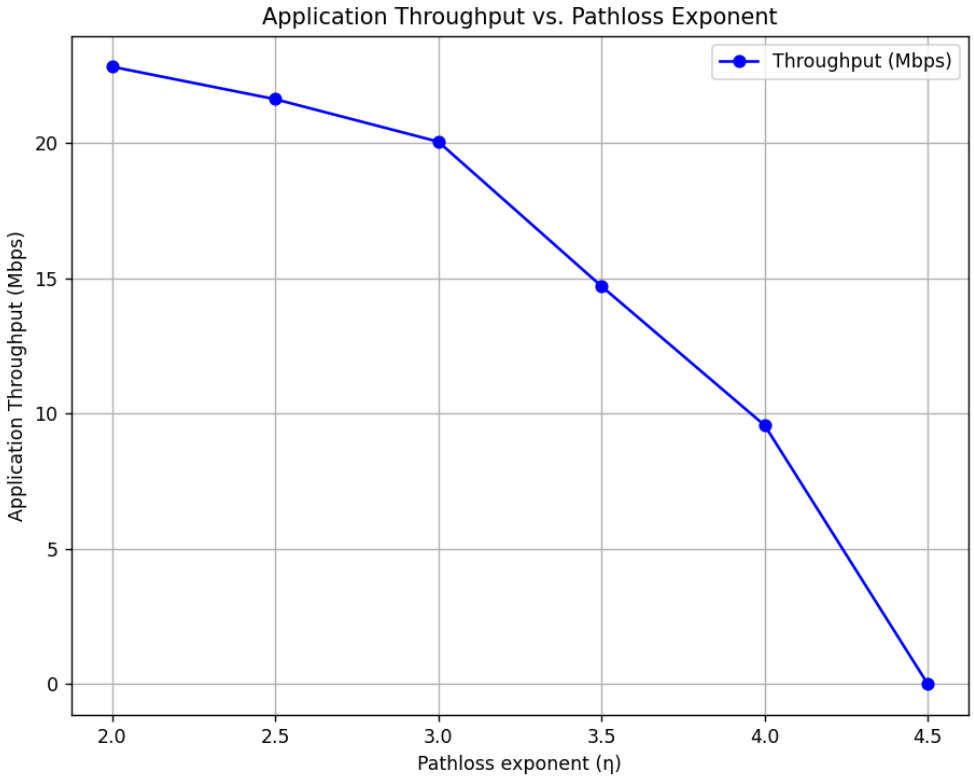



**Figure 4-13:** *Plot of Path loss Exponent vs Throughput.*

### 4.3.3 Effect of Transmitter power

Open NetSim and Select Examples ▷ Internetworks ▷ Wi-Fi ▷ Effect of Transmitter Power then click on the tile in the middle panel to load the example as shown in Figure 4-14.

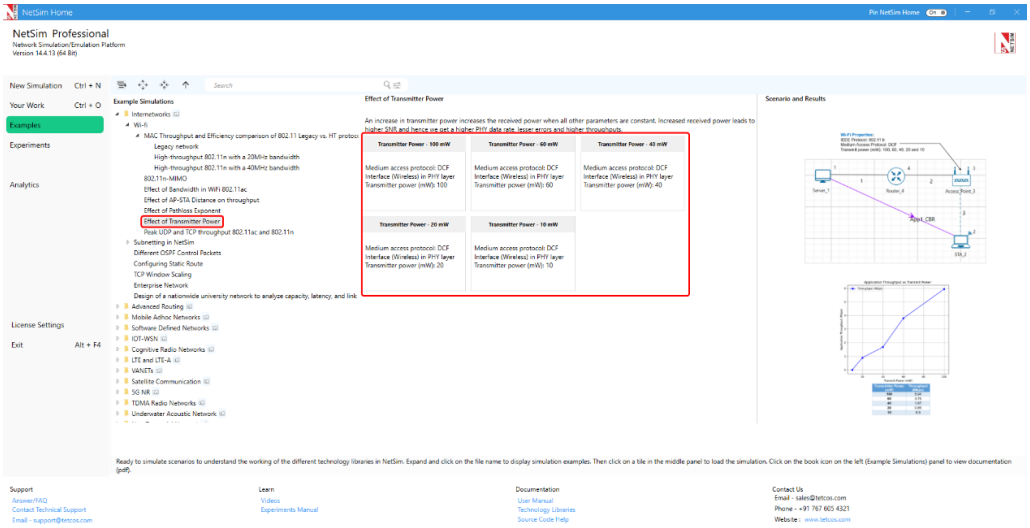



**Figure 4-14:** *List of scenarios for the example of effect of transmitter power.*

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file see Figure 4-15.

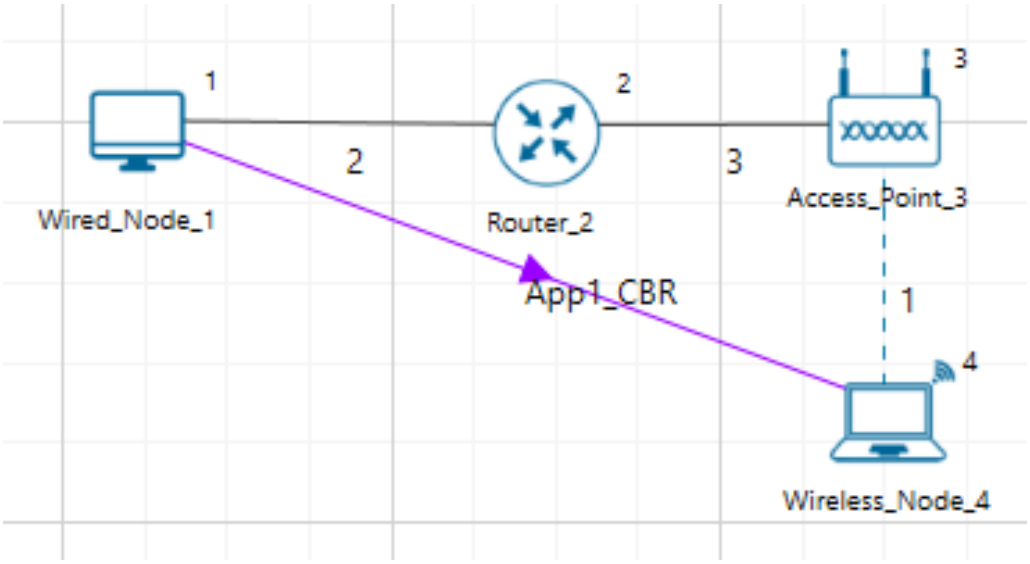



**Figure 4-15:** *Network setup for studying the effect of transmitter power.*

Increase in transmitter power increases the received power when all other parameters are constant. Increased received power leads to higher SNR and hence higher PHY Data rates, fewer errors and higher throughputs.

### Network Settings

1. Set grid length to 120m × 60m from the grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Distance between Access Point and the Wireless Node is set to 35m.
3. Set transmitter power to 100mW under Interface Wireless ▷ Physical layer properties of Access point. 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 below steps.

4. Set DCF as the medium access layer protocol under datalink layer properties of access point and wireless node.
5. Click on wireless link and open property panel on the right and set the Channel Characteristics: Path Loss Only, Path Loss Model: Log Distance, Path Loss Exponent: 3.5.
6. Configure CBR application between server and STA by clicking on the set traffic tab from the ribbon at the top. Click on created application and expand the application property panel on the right and set transport protocol to UDP, packet size to 1460 B and Inter arrival time to 1168 $\mu$s.
7. Run the simulation for 10s.
8. Go back to the scenario and decrease the Transmitter Power to 60, 40, 20 and 10 respectively and run simulation for 10s. See that, there is a decrease in the Throughput gradually.

### Results

**Table 4-5:** *Result comparison of different transmitter power vs. throughput.*

| Transmitter Power (mW) | Throughput (Mbps) | Phy Rate (Mbps) |
|---|---|---|
| 100 | 5.94 | 11 |
| 60 | 3.79 | 5 |
| 40 | 1.67 | 2 |
| 20 | 0.89 | 1 |
| 10 | 0.0 | 0 |

### Plot

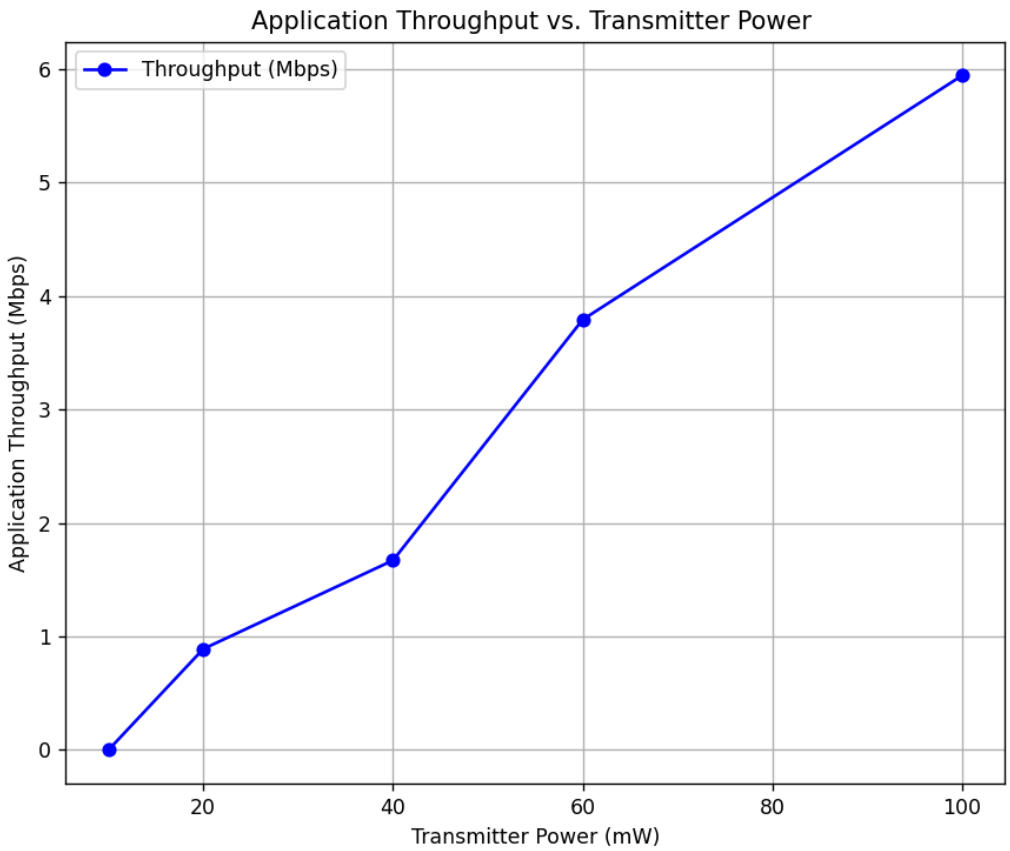



**Figure 4-16:** *Transmitter power vs Throughput.*

## 4.4 Peak UDP and TCP throughput 802.11ac and 802.11n

Open NetSim, Select Examples ▷ Internetworks ▷ Wi-Fi ▷ Peak UDP and TCP throughput 802.11ac and 802.11n then click on the tile in the middle panel to load the example as shown Figure 4-17.

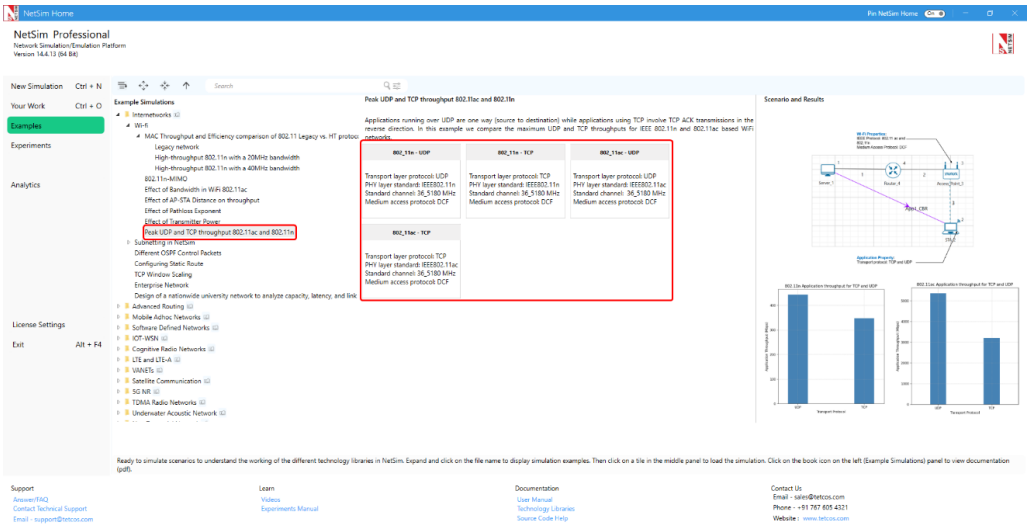


**Figure 4-17:** *List of scenarios for the example of Peak UDP and TCP throughput 802.11ac and 802.11n.*

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file as shown Figure 4-18.

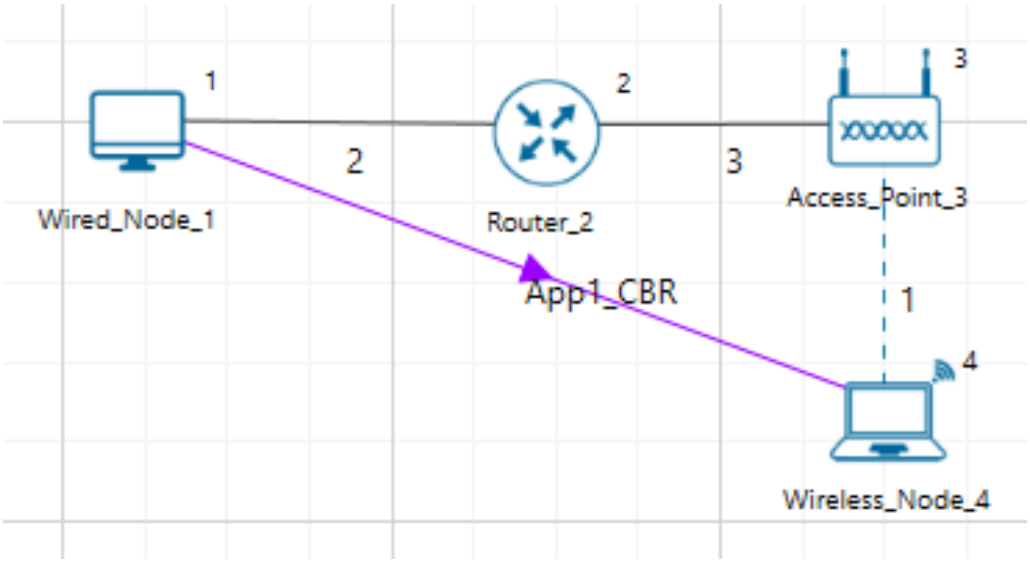



**Figure 4-18:** *Network setup for studying the Peak UDP and TCP throughput 802.11ac and 802.11n.*

### 4.4.1 IEEE802.11n

**Network Settings**

1. Set the following property in access point and wireless node as shown in below given Table 4-6. To configure any properties in the nodes, click on the node, expand the property panel on the right side, and change the properties as follows.

**Table 4-6:** *Detailed Network Parameters for IEEE802.11n.*

| Interface Parameters | Value |
|---|---|
| **Physical Layer** | |
| Standard | IEEE802.11n |
| No. of Frames to aggregate | 64 |
| Transmitter Power | 100mW |
| Frequency Band | 5 GHz |
| Bandwidth | 40 MHz |
| Standard Channel | 36 (5180MHz) |
| Guard Interval | 400ns |
| **Antenna** | |
| Antenna height | 1m |
| Antenna Gain | 0 |
| Transmitting Antennas | 4 |
| Receiving Antennas | 4 |
| **Datalink Layer** | |
| Rate Adaptation | False |
| Short Retry Limit | 7 |
| Buffer Size | 100 MB |
| Long Retry Limit | 4 |
| Dot11_RTSThreshold | 3000 bytes |
| Medium Access Protocol | DCF |

2. To configure wired and wireless link properties, click on link, expand the property panel on the right and set wired link properties as shown below.
3. Uplink speed and Downlink speed (Mbps) - 1000 Mbps.
4. Uplink BER and Downlink BER - 0.
5. Uplink and Downlink Propagation Delay($\mu$s) - 10.
6. The Channel Characteristics were set as No pathloss in wireless link properties.
7. Configure CBR application from source node (Wired Node) to destination node (Wireless Node) by clicking on the set traffic tab from the ribbon at the top. Click on created application, expand the application property panel on the right and set the following properties.

**Table 4-7:** *Application Parameters.*

| Application Properties | Value |
|---|---|
| App1 CBR | |
| Packet Size (Byte) | 1460 |
| Inter Arrival Time ($\mu$s) | 11.6 |
| Transport Protocol | UDP |

8. Run simulation for 5 sec. After simulation completes go to metrics window and note down throughput value from application metrics.

9. Go Back to 802.11n UDP scenario and change transport protocol to TCP, window scaling is set to true and scale shift count set to 5 in the transport layer of wired node and wireless node for the other sample (i.e 802.11n TCP), run the simulation for 5 sec and note down throughput value from application metrics.

**Results**

**Table 4-8:** *Results comparison of TCP and UDP throughputs for IEEE802.11n.*

| Transport Protocol | Throughput (Mbps) |
|---|---|
| UDP | 443.16 |
| TCP | 347.51 |

**Plot**

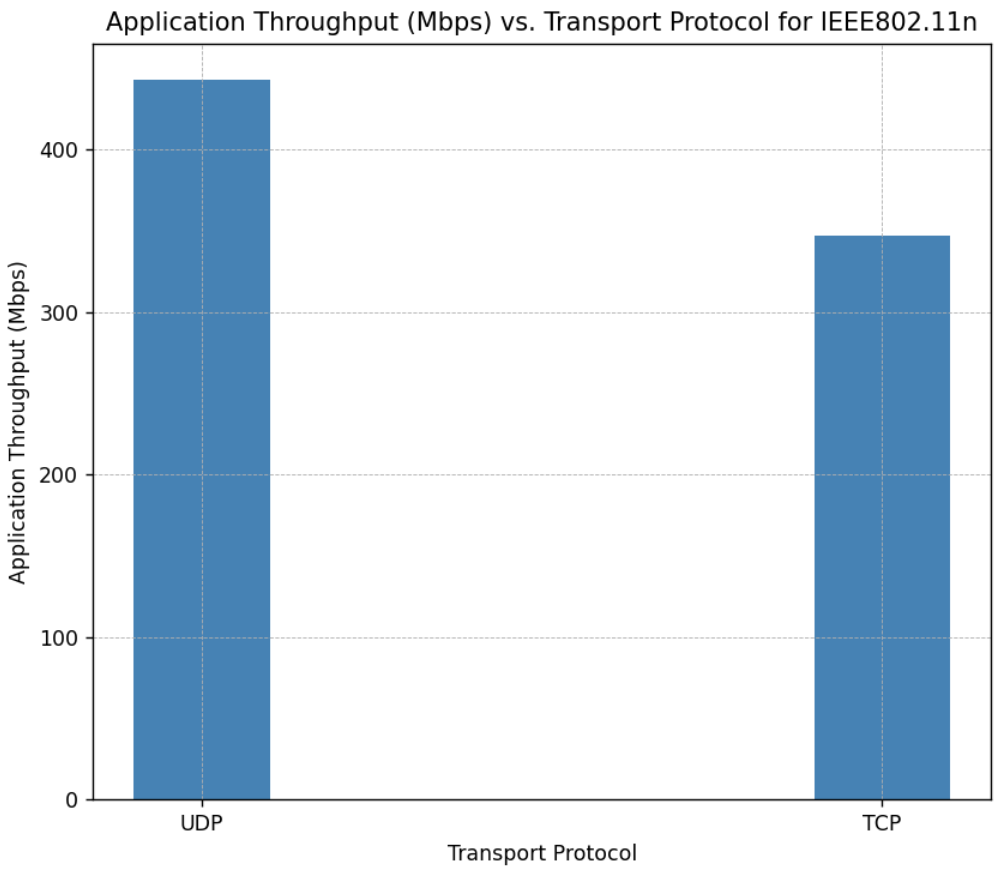



**Figure 4-19:** *Plot of Throughput (Mbps) Vs. Transport Protocol for IEEE802.11n.*

### 4.4.2 IEEE802.11ac

**Network Settings**

1. Set the following property in access point and wireless node as shown in below given table:

**Table 4-9:** *Detailed Network Parameters for IEEE802.11ac.*

| Interface Parameters | Value |
|---|---|
| **Datalink Layer** | |
| Rate Adaptation | False |
| Short Retry Limit | 7 |
| Long Retry Limit | 4 |
| Dot11_RTSThreshold | 3000 bytes |
| Medium Access Protocol | DCF |
| **Physical Layer** | |
| Standard | IEEE802.11ac |
| No. of Frames to aggregate | 1024 |
| Transmitter Power | 100mW |
| Frequency Band | 5 GHz |
| Bandwidth | 160 MHz |
| Standard Channel | 36 (5180MHz) |
| Guard Interval | 400ns |
| **Antenna** | |
| Antenna height | 1m |
| Antenna Gain | 0 |
| Transmitting Antennas | 8 |
| Receiving Antennas | 8 |
| **Access Point** | |
| Buffer Size | 100MB |

2. To configure wired and wireless link properties, click on link, expand the property panel on the right and set wired link properties as shown below.
3. Uplink speed and Downlink speed (Mbps) - 10000 Mbps.
4. Uplink BER and Downlink BER - 0.
5. Uplink and Downlink Propagation Delay($\mu$s) - 10.
6. The Channel Characteristics were set as No pathloss in wireless link properties.
7. Configure CBR application from source node (Wired Node) to destination node (Wireless Node) by clicking on the set traffic tab from the ribbon at the top. Click on created application, expand the application property panel on the right and set the following properties.

**Table 4-10:** *Application Parameters.*

| Application Properties | Value |
|---|---|
| App1_CBR | |
| Packet Size (Byte) | 1450 |
| Inter Arrival Time ($\mu$s) | 1.93 |
| Transport Protocol | UDP |

8. Run simulation for 10 sec. After simulation completes go to metrics window and note down throughput value from application metrics.
9. Go Back to the 802.11ac UDP scenario and change transport protocol to TCP, window scaling is set to true and scale shift count set to 5 in the transport layer of wired node and wireless node for the other sample (i.e 802.11ac TCP), run the simulation for 10 sec and note down throughput value from application metrics.

**Results**

**Table 4-11:** *Results comparison of TCP and UDP throughputs for IEEE802.11ac.*

| Transport Protocol | Throughput (Mbps) |
|---|---|
| UDP | 5589.08 |
| TCP | 3397.16 |

**Plot**

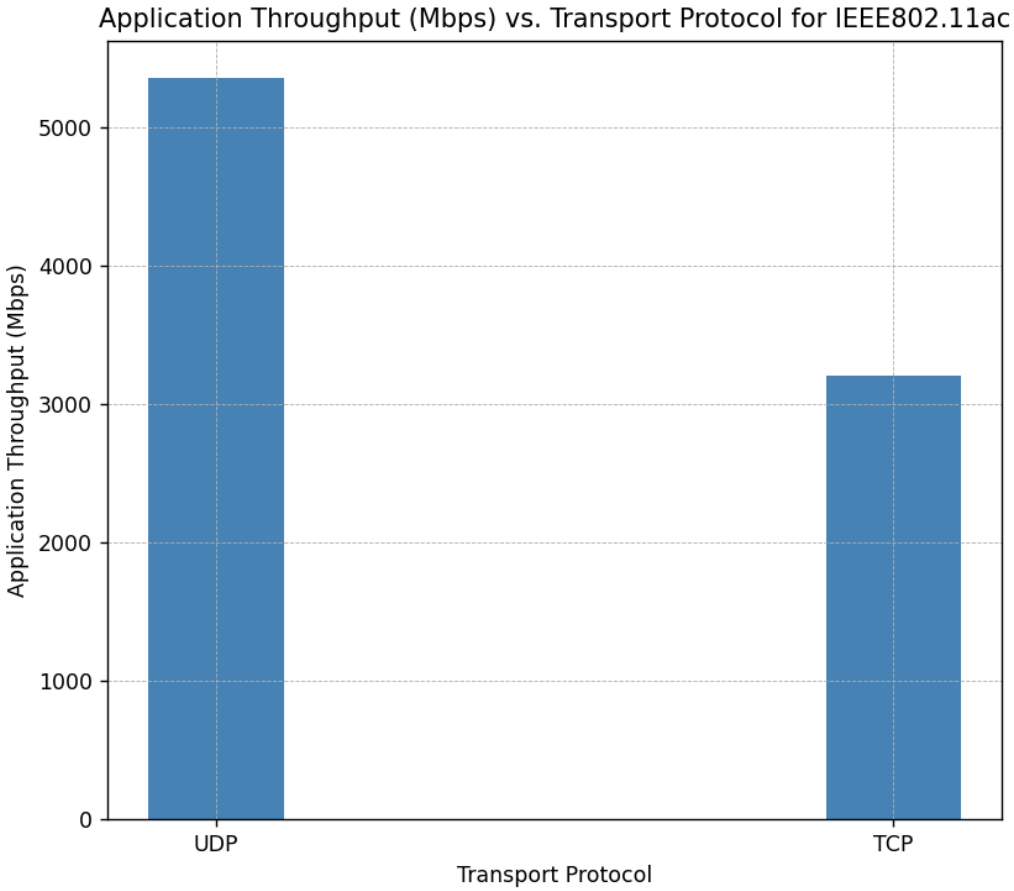



**Figure 4-20:** *Plot of Throughput (Mbps) Vs. Transport Protocol for IEEE802.11ac.*

## 4.5 MAC Throughput and Efficiency comparison of 802.11 Legacy vs. HT protocols

This example is based on Part II, Section 9 by E. Perahia and R. Stacey, from 'Next Generation Wireless LANs,' published by Cambridge University Press.

### 4.5.1 Network Scenario

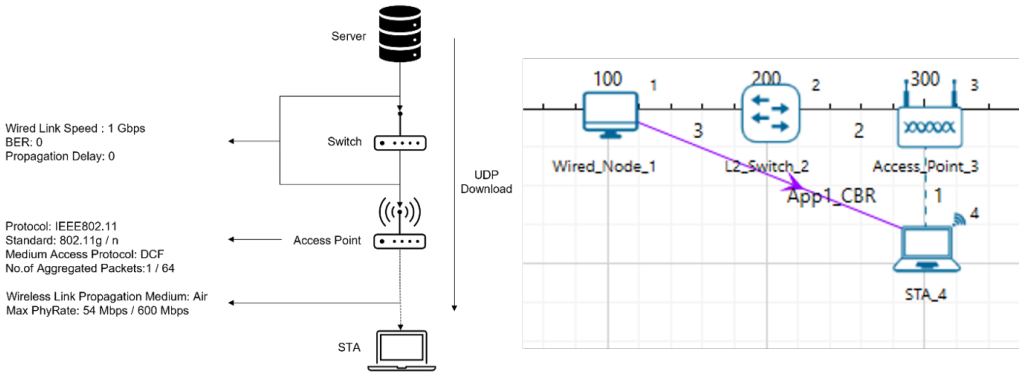



**Figure 4-21:** *Network Scenario consists of 1 Wired Node, 1 Switch, 1 AP and 1 STA.*

### 4.5.2 Part I: Legacy

Steps:

1. Create a scenario with 1 Wired Node,1 L2 Switch, 1 Access Point and 1 STA as shown in Figure 4-21.
2. Click on Wired link and expand property panel on the right and set the Wired Link Properties, Uplink and Downlink Speed as 1000 Mbps, Bit Error Rate to 0, and Propagation Delay to 0 $\mu$s.
3. Click on Wireless link and expand property panel and configure the Wireless Link Properties as shown below.

**Table 4-12:** *Wireless Link Properties.*

| Wireless Link Parameters | Value |
|---|---|
| Channel Characteristics | Pathloss |
| Pathloss Model | Log Distance |
| Pathloss Exponent | 2 |

4. Configure the CBR application between two nodes by clicking on the set traffic tab from the ribbon at the top, selecting Source ID as 1 and Destination ID as 4. Click on the created application and set the transport protocol to UDP with a Packet Size as 1460 Bytes and Inter Arrival Time as 194.66 $\mu$s.
5. Configure the Access Point and Wireless Node properties as shown below.

**Table 4-13:** *Network Configuration for IEEE802.11g.*

| Parameter | Value |
|---|---|
| **Interface (Wireless) Datalink Layer** | |
| Rate Adaptation | False |
| RTS Threshold | 4692480 Bytes |
| Long Retry Limit | 4 |
| Medium Access Protocol | DCF |
| Buffer Size (Access Point) | 100 MB |
| **Interface (Wireless) Physical Layer** | |
| Standard | IEEE802.11g |
| Transmitter Power | 100mW |
| Frequency Band | 2.4 GHz |
| Bandwidth | 20MHz |
| Standard Channel | 1 (2412MHz) |
| **Antenna** | |
| Antenna height | 0 m |
| Antenna Gain | 0 |

6. Go to Configure Reports Tab ▷ Click on Plots ▷ Expand Network Logs ▷ Enable IEEE 802.11 Radio Measurements Log.

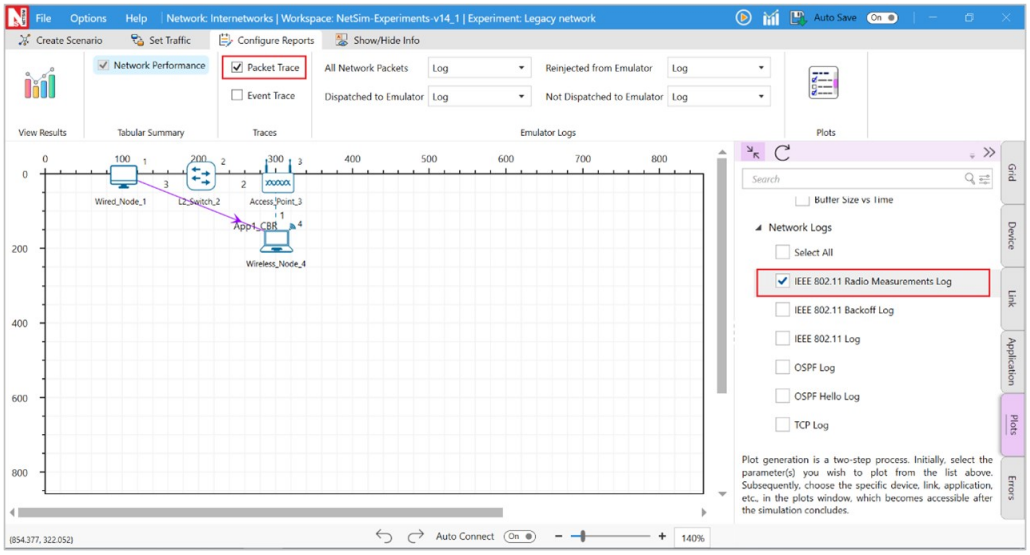



**Figure 4-22:** *Enabling IEEE 802.11 Radio Measurements logs.*

7. Run the Simulation for 10 seconds.

8. Similarly vary the distance between Access Point and Wireless Nodes, Note down the results such as MCS and Throughput (Mbps) by filtering the control packet type to data packets in IEEE 802.11 Radio Measurements Log.

### 4.5.3 Part II: High Throughput (HT)

#### Case 1: High-throughput 802.11n with a 20MHz bandwidth

1. Consider the same above Legacy network case.
2. Configure the CBR application between two nodes by clicking on the set traffic tab from the ribbon at the top, selecting Source ID as 1 and Destination ID as 4. Click on the created application and set the transport protocol to UDP with a Packet Size as 1460 Bytes and Inter Arrival Time as 57.14 $\mu$s.
3. Configure the Access Point and Wireless Node properties as shown below.

**Table 4-14:** *Network Configuration for IEEE802.11n.*

| Parameter | Value |
|---|---|
| **Interface (Wireless) Datalink Layer** | |
| Rate Adaptation | False |
| RTS Threshold | 4692480 Bytes |
| Long Retry Limit | 4 |
| Medium Access Protocol | DCF |
| Buffer Size (Access Point) | 100 MB |
| **Interface (Wireless) Physical Layer** | |
| Standard | IEEE802.11n |
| No. of Frames aggregated | 1 |
| Transmitter Power | 100mW |
| Frequency Band | 2.4 GHz |
| Bandwidth | 20MHz |
| Standard Channel | 1 (2412MHz) |
| Guard Interval | 400 ns |
| **Antenna** | |
| Antenna height | 0 m |
| Antenna Gain | 0 |
| TX Antenna Count | 1 |
| RX Antenna Count | 1 |

4. Go to Configure Reports tab and enable packet trace and click on Plots > on the right-hand side Tab > expand Network Logs > enable IEEE 802.11 Radio Measurements (see Figure 4-22).
5. Run the simulation for 10 seconds.
6. Similarly vary Antenna Count and the distance between Access Point and Wireless Node as shown in Table 4-16.
7. Run the simulation for 10 seconds. Record the results such as MCS from IEEE 802.11 Radio Measurements Log and Throughput (Mbps) from the Application Metrics.

8. Change the No. of Frames Aggregated to 64, repeat the Step 6 and 7 again. The similar settings are done in 802.11n-20MHz-Frames-Aggregation:64 sample

**NOTE:**

- In NetSim, we have provided one sample for each case. For additional samples, you need to run simulations with varying values for antenna count and distance as mentioned in Table 4-15, Table 4-16 and Table 4-17.
- In the IEEE 802.11 Radio Measurements Log, filter the Control Packet Type to App1_CBR to note down the MCS value for all cases.

### Case 2: High-throughput 802.11n with a 40MHz bandwidth

1. For the same case 1, change the bandwidth to 40MHz, No. of Frames Aggregated to 1 and rerun the sample by varying Antenna Count and the distance between Access Point and Wireless Node as shown in Table 4-17.
2. Run the Simulation for 10 seconds. Record the results such as MCS from IEEE 802.11 Radio Measurements Log and Throughput (Mbps) from the Application Metrics.
3. Again, change the No. of Frames Aggregated to 64. Vary Antenna Count and the distance between Access Point and Wireless Node as shown in Table 4-17. The similar settings are done in 802.11n 40MHz Frames Aggregation 64 sample.
4. Run the Simulation for 10 seconds. Record the results such as MCS from IEEE 802.11 Radio Measurements Log and Throughput (Mbps) from the Application Metrics.

**MAC Efficiency:**

MAC efficiency is defined as

$$\text{Efficiency (\%)} = \frac{\text{Throughput (Mbps)}}{\text{PHY Rate (Mbps)}} \times 100 \tag{19}$$

### Results

**802.11g (Legacy):**

**Table 4-15:** *Results of 802.11g (Legacy).*

| Standard | Distance (m) | MCS | Rx Sens. (dBm) | PHY Rate | Throughput | Efficiency (%) |
|---|---|---|---|---|---|---|
| 802.11g | 166 | 7 | −65 | 54 | 29.21 | 54 |
| 802.11g | 196 | 6 | −66 | 48 | 27.31 | 57 |
| 802.11g | 301 | 5 | −70 | 36 | 22.78 | 63 |
| 802.11g | 496 | 4 | −74 | 24 | 17.11 | 71 |
| 802.11g | 691 | 3 | −77 | 18 | 13.70 | 76 |
| 802.11g | 871 | 2 | −79 | 12 | 9.80 | 82 |
| 802.11g | 1096 | 1 | −81 | 9 | 7.63 | 85 |
| 802.11g | 1111 | 0 | −82 | 6 | 5.29 | 88 |

**NOTE:**

The PHY Rate and Receiver Sensitivity values are taken from the IEEE 802.11 (Wi-Fi) Standard as follows:

- IEEE 802.11g: PHY Rate and Receiver Sensitivity are specified in Table 17-4 and Table 17-18 of the IEEE 802.11-2020 standard.
- IEEE 802.11n: PHY Rate is specified in Tables 19-27 to 19-34, and Receiver Sensitivity is specified in Table 19-23 of the IEEE 802.11-2020 standard.

### Case 1: 802.11n 20 MHz:

**Table 4-16:** *Results for 802.11n with 20 MHz Bandwidth varying Distance, MCS and Phy Rate.*

| | | | | | Aggregation = 1 | | Aggregation = 64 | |
|---|---|---|---|---|---|---|---|---|
| Dist. (m) | Ant. | MCS | Rx Sens. | PHY Rate | Tput | Eff. (%) | Tput | Eff. (%) |
| 151 | 1x1 | 7 | −64 | 72.2 | 23.99 | 33 | 66.88 | 93 |
| 166 | 1x1 | 6 | −65 | 65 | 23.11 | 36 | 60.38 | 93 |
| 196 | 1x1 | 5 | −66 | 57.8 | 22.08 | 38 | 53.84 | 93 |
| 301 | 1x1 | 4 | −70 | 43.3 | 19.47 | 45 | 40.56 | 94 |
| 496 | 1x1 | 3 | −74 | 28.9 | 15.77 | 55 | 27.23 | 94 |
| 691 | 1x1 | 2 | −77 | 21.7 | 13.25 | 61 | 20.50 | 94 |
| 871 | 1x1 | 1 | −79 | 14.4 | 10.00 | 69 | 13.64 | 95 |
| 886 | 1x1 | 0 | −82 | 7.2 | 5.79 | 80 | 6.84 | 95 |
| 151 | 2x2 | 7 | −64 | 144.4 | 31.05 | 0 | 130.67 | 90 |
| 166 | 2x2 | 6 | −65 | 130 | 30.29 | 22 | 118.25 | 91 |
| 196 | 2x2 | 5 | −66 | 115.6 | 29.39 | 23 | 105.69 | 91 |
| 301 | 2x2 | 4 | −70 | 86.7 | 26.99 | 25 | 80.08 | 92 |
| 496 | 2x2 | 3 | −74 | 57.8 | 23.21 | 31 | 53.94 | 93 |
| 691 | 2x2 | 2 | −77 | 43.3 | 20.34 | 40 | 40.62 | 94 |
| 871 | 2x2 | 1 | −79 | 28.9 | 16.34 | 47 | 27.26 | 94 |
| 886 | 2x2 | 0 | −82 | 14.4 | 10.23 | 57 | 13.65 | 95 |
| 151 | 3x3 | 7 | −64 | 216.7 | 33.37 | 71 | 191.18 | 88 |
| 166 | 3x3 | 6 | −65 | 195 | 32.78 | 0 | 173.32 | 89 |
| 196 | 3x3 | 5 | −66 | 173.3 | 32.08 | 15 | 155.18 | 90 |
| 301 | 3x3 | 4 | −70 | 130 | 30.13 | 17 | 118.21 | 91 |
| 496 | 3x3 | 3 | −74 | 86.7 | 26.87 | 19 | 80.06 | 92 |
| 691 | 3x3 | 2 | −77 | 65 | 24.24 | 23 | 60.49 | 93 |
| 871 | 3x3 | 1 | −79 | 43.3 | 20.27 | 31 | 40.61 | 94 |
| 886 | 3x3 | 0 | −82 | 21.7 | 13.62 | 37 | 20.52 | 95 |
| 151 | 4x4 | 7 | −64 | 288.9 | 35.96 | 47 | 249.53 | 86 |
| 166 | 4x4 | 6 | −65 | 260 | 35.44 | 63 | 226.67 | 87 |
| 196 | 4x4 | 5 | −66 | 231.1 | 34.81 | 12 | 203.44 | 88 |
| 301 | 4x4 | 4 | −70 | 173.3 | 33.07 | 14 | 155.54 | 90 |
| 496 | 4x4 | 3 | −74 | 115.6 | 30.07 | 15 | 105.83 | 92 |
| 691 | 4x4 | 2 | −77 | 86.7 | 27.56 | 19 | 80.15 | 92 |
| 871 | 4x4 | 1 | −79 | 57.8 | 23.62 | 26 | 53.97 | 93 |
| 886 | 4x4 | 0 | −82 | 28.9 | 16.54 | 32 | 27.27 | 94 |

### Case 2: 802.11n 40 MHz:

**Table 4-17:** *Results for 802.11n with 40 MHz Bandwidth varying Distance, MCS and Phy Rate.*

| | | | | | Aggregation = 1 | | Aggregation = 64 | |
|---|---|---|---|---|---|---|---|---|
| Dist. (m) | Ant. | MCS | Rx Sens. | PHY Rate | Tput | Eff. (%) | Tput | Eff. (%) |
| 106 | 1x1 | 7 | −61 | 150 | 30.71 | 20 | 135.29 | 90 |
| 121 | 1x1 | 6 | −62 | 135 | 29.99 | 22 | 122.43 | 91 |
| 136 | 1x1 | 5 | −63 | 120 | 29.14 | 24 | 109.44 | 91 |
| 211 | 1x1 | 4 | −67 | 90 | 26.86 | 30 | 82.96 | 92 |
| 346 | 1x1 | 3 | −71 | 60 | 23.23 | 39 | 55.92 | 93 |
| 496 | 1x1 | 2 | −74 | 45 | 20.46 | 45 | 42.17 | 94 |

*Continued on next page*

| Dist. | Ant. | MCS | Rx Sens. | PHY Rate | Agg. = 1 Tput | Agg. = 1 Eff. | Agg. = 64 Tput | Agg. = 64 Eff. |
|---|---|---|---|---|---|---|---|---|
| 616 | 1x1 | 1 | −76 | 30 | 16.53 | 55 | 28.28 | 94 |
| 631 | 1x1 | 0 | −79 | 15 | 10.47 | 70 | 14.21 | 95 |
| 106 | 2x2 | 7 | −61 | 300 | 36.17 | 12 | 258.20 | 86 |
| 121 | 2x2 | 6 | −62 | 270 | 35.66 | 13 | 234.62 | 87 |
| 136 | 2x2 | 5 | −63 | 240 | 35.06 | 15 | 210.66 | 88 |
| 211 | 2x2 | 4 | −67 | 180 | 33.35 | 19 | 161.20 | 90 |
| 346 | 2x2 | 3 | −71 | 120 | 30.40 | 25 | 109.70 | 91 |
| 496 | 2x2 | 2 | −74 | 90 | 27.93 | 31 | 83.12 | 92 |
| 616 | 2x2 | 1 | −76 | 60 | 24.02 | 40 | 55.99 | 93 |
| 631 | 2x2 | 0 | −79 | 30 | 16.92 | 56 | 28.29 | 94 |
| 106 | 3x3 | 7 | −61 | 450 | 37.16 | 8 | 368.53 | 82 |
| 121 | 3x3 | 6 | −62 | 405.5 | 36.81 | 9 | 336.73 | 83 |
| 136 | 3x3 | 5 | −63 | 360 | 36.36 | 10 | 303.27 | 84 |
| 211 | 3x3 | 4 | −67 | 270 | 35.11 | 13 | 234.25 | 87 |
| 346 | 3x3 | 3 | −71 | 180 | 32.86 | 18 | 161.03 | 89 |
| 496 | 3x3 | 2 | −74 | 135 | 30.90 | 23 | 122.66 | 91 |
| 616 | 3x3 | 1 | −76 | 90 | 27.59 | 31 | 83.07 | 92 |
| 631 | 3x3 | 0 | −79 | 45 | 20.88 | 46 | 42.20 | 94 |
| 106 | 4x4 | 7 | −61 | 600 | 39.19 | 7 | 471.84 | 79 |
| 121 | 4x4 | 6 | −62 | 540 | 38.90 | 7 | 432.30 | 80 |
| 136 | 4x4 | 5 | −63 | 480 | 38.53 | 8 | 391.31 | 82 |
| 211 | 4x4 | 4 | −67 | 360 | 37.46 | 10 | 304.43 | 85 |
| 346 | 4x4 | 3 | −71 | 240 | 35.51 | 15 | 210.91 | 88 |
| 496 | 4x4 | 2 | −74 | 180 | 33.76 | 19 | 161.35 | 90 |
| 616 | 4x4 | 1 | −76 | 120 | 30.75 | 26 | 109.77 | 91 |
| 631 | 4x4 | 0 | −79 | 60 | 24.23 | 40 | 56.01 | 93 |

## Comparison Plots and Discussion

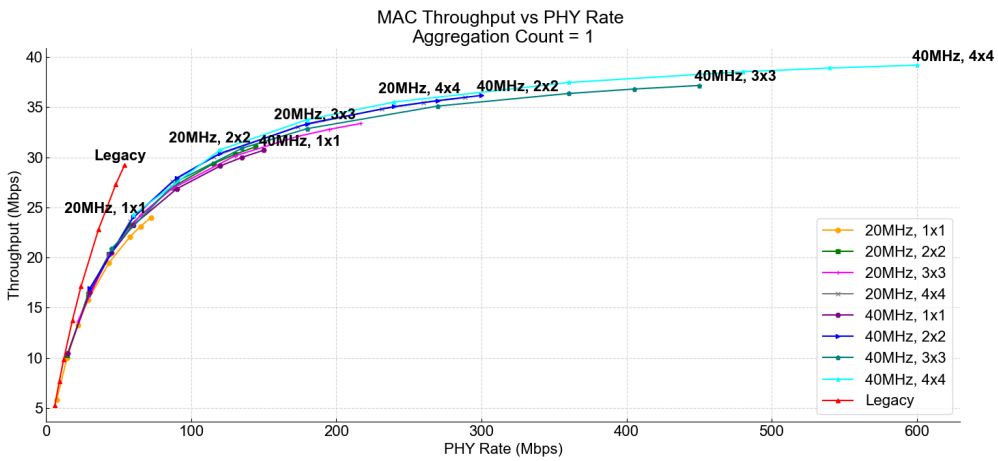



**Figure 4-23:** *Plot for MAC Throughput vs PHY Rate for Legacy and HT with 20MHz and 40 MHz with Aggregation count set to 1.*

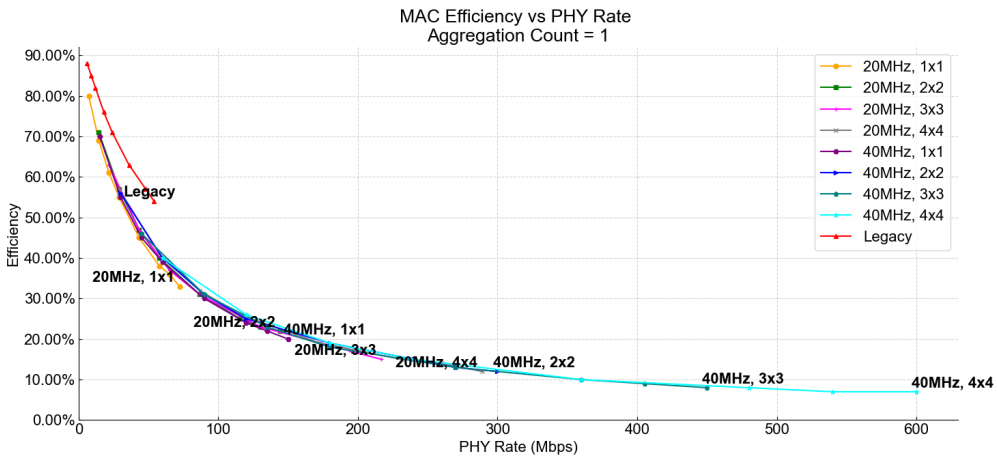



**Figure 4-24:** *Plot for MAC Efficiency vs PHY Rate, Legacy and HT with 20MHz and 40 MHz with Aggregation count set to 1.*

Let us observe Figure 4-23 and Figure 4-24.

Without packet aggregation (aggregation count = 1), throughput increases with PHY rate initially because higher PHY rates can transmit more data per unit of time. However, the throughput flattens at higher PHY rates due to inefficiencies such as protocol overhead, inter-frame spacing, and acknowledgments, which do not scale with PHY rate.

Efficiency, defined as the ratio of throughput to PHY rate, naturally decreases at higher PHY rates without aggregation because the non-scalable overhead consumes a larger proportion of the available bandwidth.

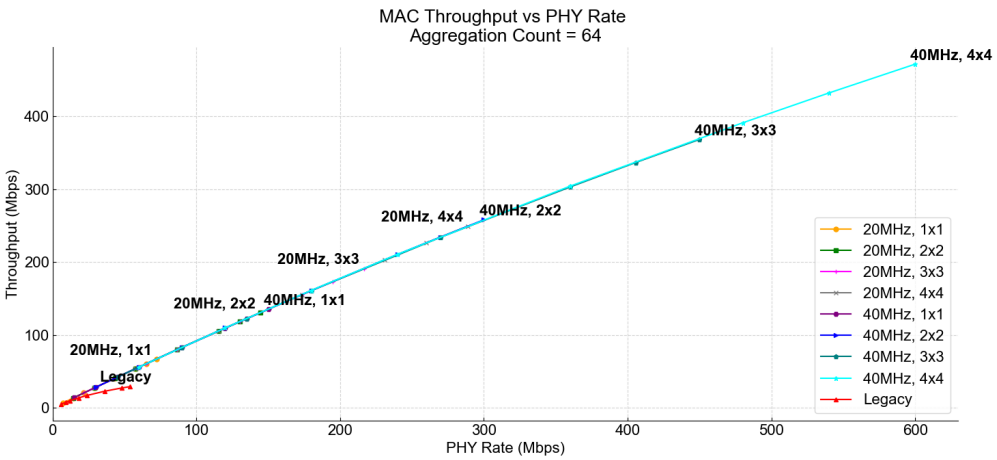



**Figure 4-25:** *Plot for MAC Throughput vs PHY Rate, Legacy and HT with 20MHz and 40 MHz with Aggregation count set to 64.*

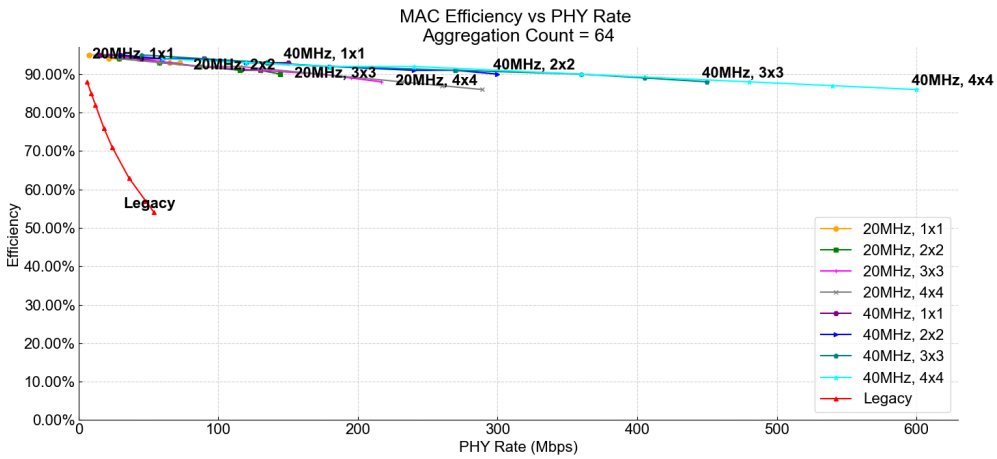



**Figure 4-26:** *Plot for MAC Efficiency vs PHY Rate, Legacy and HT with 20MHz and 40 MHz with Aggregation = 64.*

We now turn to Figure 4-25 and Figure 4-26.

In Wi-Fi 802.11n, packet aggregation involves combining multiple smaller packets into a larger one before transmission. This technique reduces overhead by spreading the fixed costs of transmitting a packet, such as headers and acknowledgment frames, over a larger payload.

With packet aggregation (aggregation count = 64), the overhead is amortized over a larger amount of data, resulting in higher throughput. This increase in throughput with packet aggregation is less impacted by the overhead, allowing the network to maintain higher efficiency even as the PHY rate increases.

The improvement with aggregation is more pronounced at higher PHY rates, as the relative overhead reduction becomes more significant. Hence, packet aggregation is a key technique to improve both throughput and efficiency, especially at higher data rates.

## 4.6 Configuring IP addresses, subnets and applying firewall rules based on subnets using Class B IP addresses

### 4.6.1 IP Addressing

A unique number ID is assigned to one of the hosts or interfaces in a network. An IP address is an address used to uniquely identify a device on an IP network. An IPv4 address is made up of 32 binary bits, which can be divided into a network portion and a host portion with the help of a subnet mask. These 32 binary bits are further broken into four octets (1 octet = 8 bits). Each octet is converted to decimal and separated by a period (dot). For this reason, an IP address is said to be expressed in dotted decimal format (for example, 172.16.0.0). The value in each octet ranges from 0 to 255 in decimal, or 00000000 – 11111111 in binary.

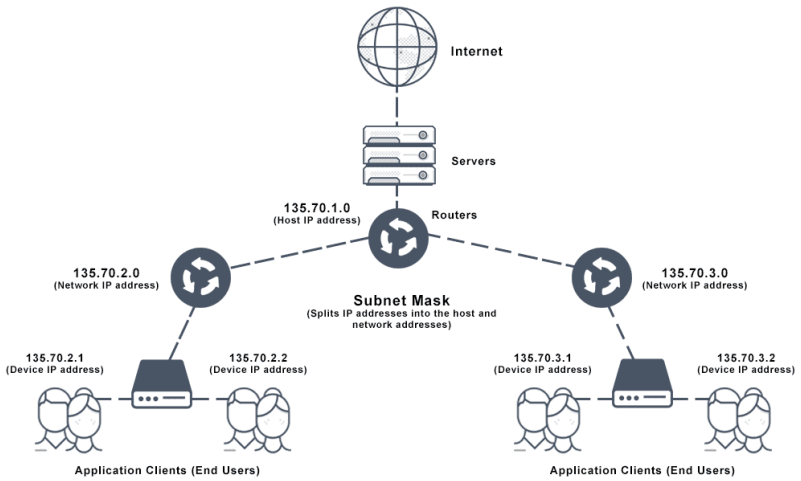



**Figure 4-27:** *Subnet mask diagram.*

### 4.6.2 IP address classes

**Table 4-18:** *IP address class and its application.*

| Class | Address range | Subnet masking | Leading Bits | Max Networks | Max Hosts | Application |
|---|---|---|---|---|---|---|
| IP CLASS A | 1 to 126 | 255.0.0.0 | 8 | 128 | 16,777,214 | Used for a large number of hosts. |
| IP CLASS B | 128 to 191 | 255.255.0.0 | 16 | 16384 | 65,534 | Used for medium-size networks. |
| IP CLASS C | 192 to 223 | 255.255.255.0 | 24 | 2097157 | 254 | Used for local area network. |
| IP CLASS D | 224 to 239 | N/A | N/A | N/A | N/A | Reserved for multicast. |
| IP CLASS E | 240 to 254 | N/A | N/A | N/A | N/A | Reserved for R&D. |

### 4.6.3 Configuring Class-B address in NetSim

The default IP addressing in NetSim is class A addressing. However, users can reconfigure (static) IP addresses with different classes. These settings are available in the network layer of end nodes, routers, and L3 switches.

**Example 1:** In this example, we have created a simple LAN network and modified the IP address of the routers and the users in the LAN network. Refer Figure 4-28. In this, we have used the IP address range 172.16.0.1–172.16.255.254 with a mask 255.255.0.0. When it is put differently, the IP is 172.16.0.0/16.

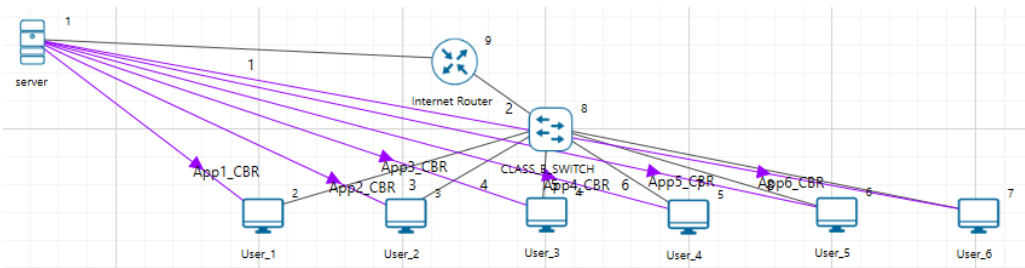



**Figure 4-28:** *IPv4 Class B IP addressing.*

### 4.6.4 Subnetting

Subnetting is the practice of dividing a network into two or more smaller networks. It increases routing efficiency, enhances security, and reduces the size of the broadcast domain.

A subnet mask is used to divide an IP address into two parts. One part identifies the host (computer), and the other part identifies the network to which it belongs. For better understanding of how an IP address and subnet masks work, look at an IP address and see how it is organized.

**Table 4-19:** *Class-B Subnetting using the slash method.*

| Subnet | Host | Slash Method |
|---|---|---|
| 1 | 65536 | /16 |
| 2 | 32768 | /17 |
| 4 | 16384 | /18 |
| 8 | 8192 | /19 |
| 16 | 4096 | /20 |
| 32 | 2048 | /21 |
| 64 | 1024 | /22 |
| 128 | 512 | /23 |
| 256 | 256 | /24 |
| 512 | 128 | /25 |
| 1024 | 64 | /26 |
| 2048 | 32 | /27 |
| 4096 | 16 | /28 |
| 8192 | 8 | /29 |
| 16384 | 4 | /30 |
| 32768 | 2 | /31 |
| 65536 | 1 | /32 |

**Table 4-20:** *Using subnet mask 255.255.192.0 i.e., /18 creating 4 different subnets with 16382 usable hosts.*

| Network ID | Subnet Mask | Host ID Range | Usable Host | Broadcast ID |
|---|---|---|---|---|
| 172.16.0.0 | /18 | 172.16.0.1–172.16.63.254 | 16382 | 172.16.63.255 |
| 172.16.64.0 | /18 | 172.16.64.1–172.16.127.254 | 16382 | 172.16.127.255 |
| 172.16.128.0 | /18 | 172.16.128.1–172.16.192.254 | 16382 | 172.16.192.255 |
| 172.16.192.0 | /18 | 172.16.192.1–172.16.255.254 | 16382 | 172.16.255.255 |

### 4.6.5 Configuring Class-B subnetting

We provide two examples to explain Class B subnets. The examples show how to create 4 subnets with 16382 hosts using: (i) a single switch and (ii) multiple switches.

(a) **Subnets using single switch:** Note that IP address and subnet masks are configured. The application (traffic) flow is configured for intra-subnet communications.

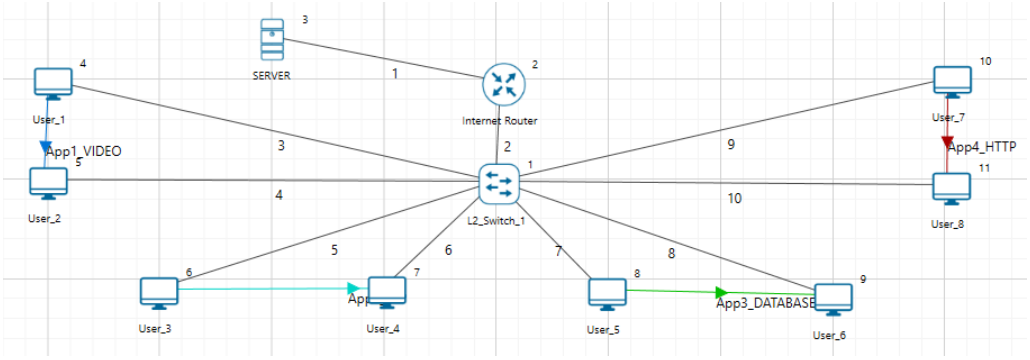



**Figure 4-29:** *Pair of users communicating with each other belong to a separate subnet per Table 4-19.*

Configuring IP addresses and subnets in NetSim is as simple as configuring it in MS Operating System. The devices in NetSim are configurable via GUI. To set the IP and subnet, users need to modify the Network Layer as shown below.

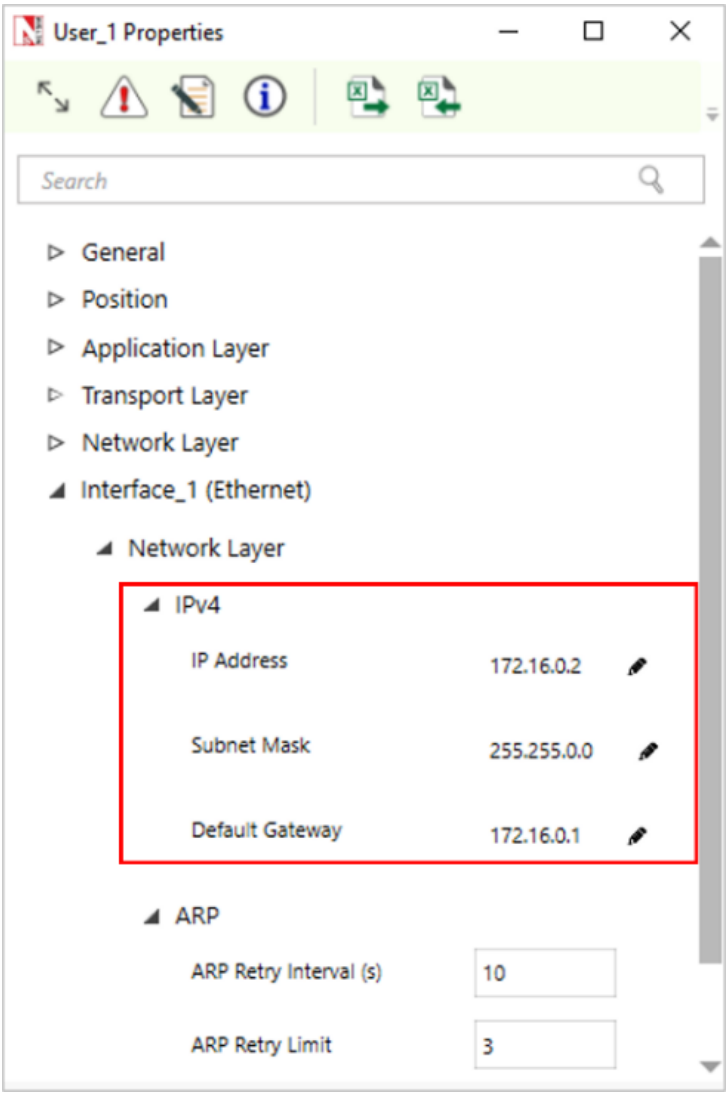



**Figure 4-30:** *Configuring IP address and subnet mask in network layer of device in NetSim.*

(b) **Subnets using multiple switches:** Here subnets have been configured using multiple switches. In the given example as shown in Figure 4-31, we have considered 4 departments

in a university campus by configuring subnets for CS, EC, MECH, and EE. Application (traffic flow) is set for both intra and inter-subnet communication.

**Table 4-21:** *Subnets in a university campus based on Table 4-19.*

| Department Name | IP Address Range |
|---|---|
| Computer Science (CS) | 172.16.0.1–172.16.63.254 |
| Electronics and Communication (EC) | 172.16.64.1–172.16.127.254 |
| Mechanical Engineering (ME) | 172.16.128.1–172.16.192.254 |
| Electrical Engineering (EE) | 172.16.192.1–172.16.255.254 |

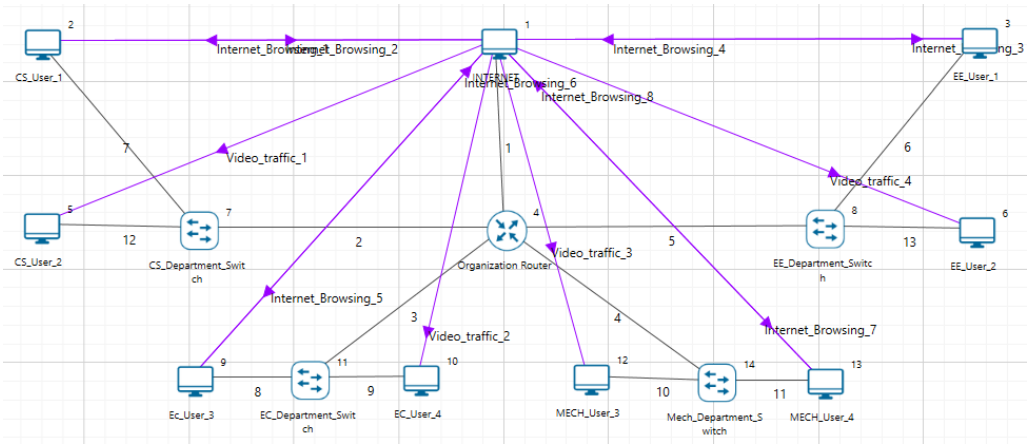



**Figure 4-31:** *All the users are communicating from different subnets using a router and switch by configuring Class-B subnetting.*

### 4.6.6 Firewall rules based on subnets

An important benefit of subnetting is security. Firewall/ACL rules can be configured at a subnet level. NetSim provides options for users to configure ACL/firewall rules i.e., to PERMIT or DENY traffic at a router based on (i) IP address/Network address (ii) Protocol (iii) Inbound / Outbound traffic.

**Example 4:** In this example, we explain how users can set firewall rules to DENY traffic at a subnet level.

The topology considered here is the university network as shown in Figure 4-31.

The firewall/ACL rules are set in the Organization Router.

**ACL Rules:**

1. For the CS-department Video traffic is denied
2. For the EC-department TCP traffic is denied
3. For the Mechanical department, Video Traffic is denied.
4. For the EE department TCP traffic is denied

These rules can be set in the NetSim UI just by filling an ACL application of the router as shown below in Figure 4-32.

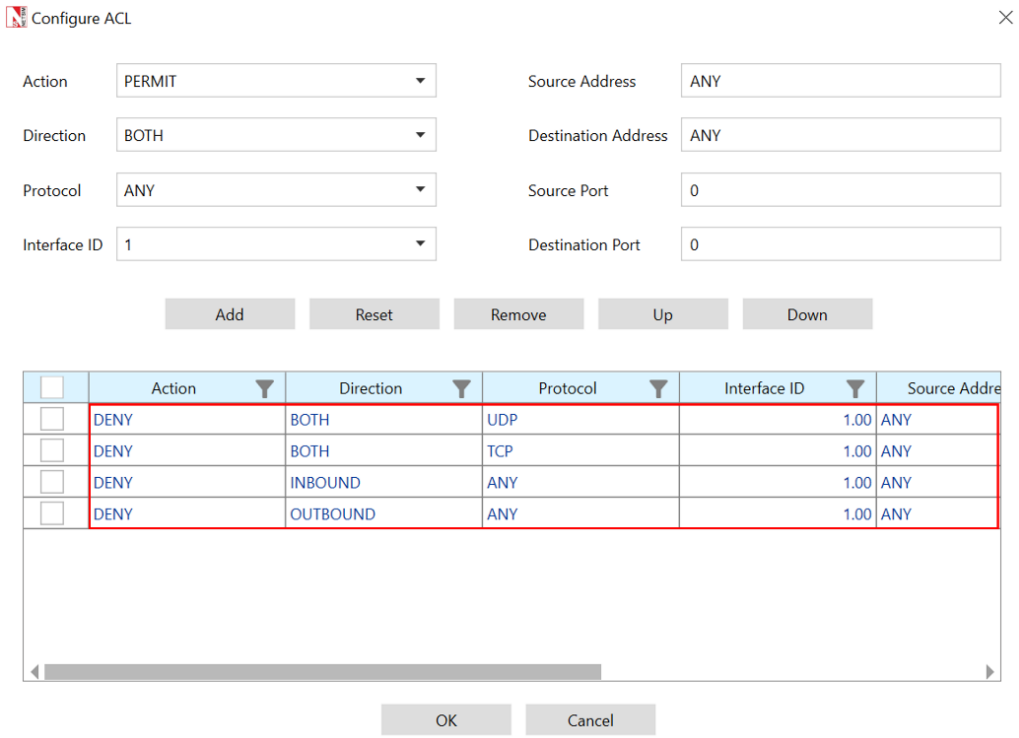
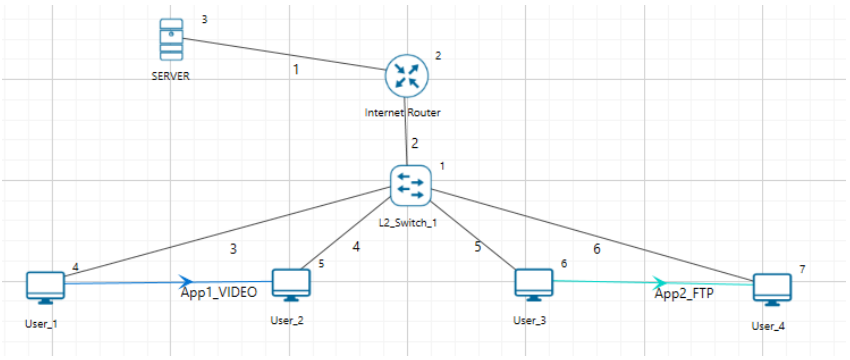



**Figure 4-32:** *Setting firewall rules in the organization router.*

## Exercises

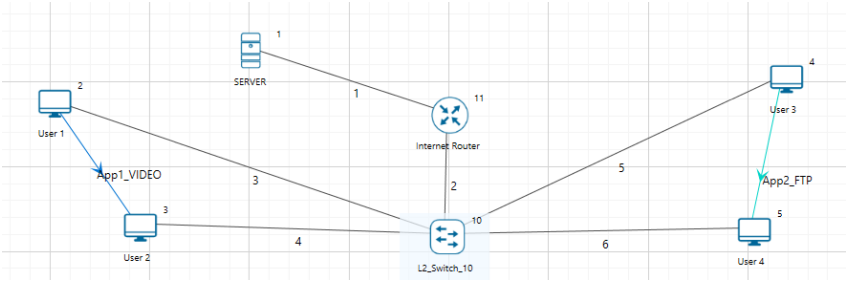
1. **Subnetting a Class B Network:** Dividing 172.16.0.0/17 into Two Subnets

   Construct the scenario as shown below and create Video and FTP application between the nodes. Using the Class B IP address range of 172.16.0.0/17, divide the network into two subnets.



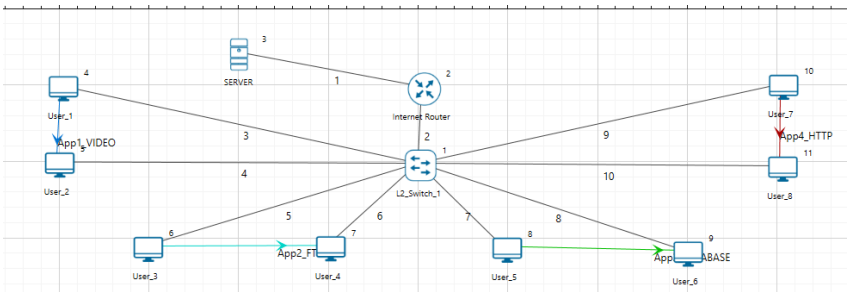

2. **Subnetting a Class C Network:** Dividing 192.168.1.0 into Two Subnets

   Construct the scenario as shown below and create Video and FTP application between the nodes. Using the Class C IP address range of 192.168.1.0/25, divide the network into two subnets.




3. **Configuring the Class-A subnetting for NetSim example**

   Construct the following network scenario to configure Class A subnetting using the 10.0.0.0/8 IP address range. Divide the network into four subnets to optimize communication between devices, including servers and users.




## 4.7 Different OSPF Control Packets

There are five distinct OSPF packet types.

**Table 4-22:** *Different OSPF Control Packets.*

| Type | Description |
|---|---|
| 1 | Hello |
| 2 | Database Description |
| 3 | Link State Request |
| 4 | Link State Update |
| 5 | Link State Acknowledgement |

1. **The Hello packets.** Hello packets are OSPF packet type 1. These packets are sent periodically on all interfaces in order to establish and maintain neighbor relationships. In addition, Hello Packets are multicast on those physical networks having a multicast or broadcast capability, enabling dynamic discovery of neighboring routers. All routers connected to a common network must agree on certain parameters (Network mask, Hello Interval and Router Dead Interval). These parameters are included in Hello packets, so that differences can inhibit the forming of neighbor relationships.

2. **The Database Description packet.** Database Description packets are OSPF packet type 2. These packets are exchanged when an adjacency is being initialized. They describe the contents of the link-state database. Multiple packets may be used to describe the database. For this purpose, a poll-response procedure is used. One of the routers is designated to be the master, the other the slave. The master sends Database Description

packets (polls) which are acknowledged by Database Description packets sent by the slave (responses). The responses are linked to the polls via the packet DD sequence numbers.

3. **The Link State Request packet.** Link State Request packets are OSPF packet type 3. After exchanging Database Description packets with a neighboring router, a router may find that parts of its link-state database are out-of-date. The Link State Request packet is used to request the pieces of the neighbor's database that are more up to date. Multiple Link State Request packets may need to be used. A router that sends a Link State Request packet has in mind the precise instance of the database pieces it is requesting. Each instance is defined by its LS sequence number, LS checksum, and LS age, although these fields are not specified in the Link State Request Packet itself. The router may receive even more recent instances in response.

4. **The Link State Update packet.** Link State Update packets are OSPF packet type 4. These packets implement the flooding of LSAs. Each Link State Update packet carries a collection of LSAs one hop further from their origin. Several LSAs may be included in a single packet. Link State Update packets are multicast on those physical networks that support multicast/broadcast. In order to make the flooding procedure reliable, flooded LSAs are acknowledged in Link State Acknowledgment packets. If retransmission of certain LSAs is necessary, the retransmitted LSAs are always sent directly to the neighbor.

5. **The Link State Acknowledgment packet.** Link State Acknowledgment Packets are OSPF packet type 5. To make the flooding of LSAs reliable, flooded LSAs are explicitly acknowledged. This acknowledgment is accomplished through the sending and receiving of Link State Acknowledgment packets. Multiple LSAs can be acknowledged in a single Link State Acknowledgment packet.

Open NetSim, Select Examples ▷ Internetworks ▷ Different OSPF Control Packets then click on the tile in the middle panel to load the example as shown in Figure 4-33.

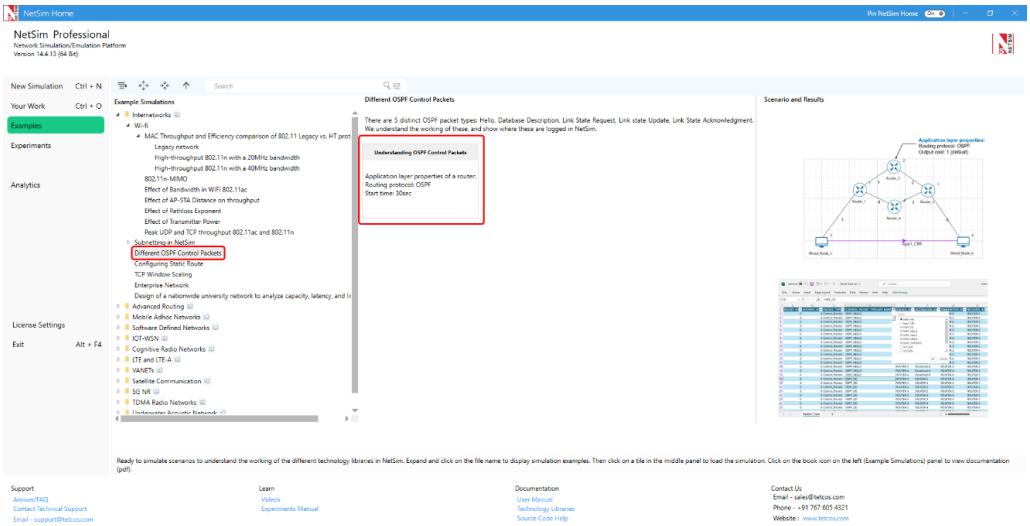



**Figure 4-33:** *List of scenarios for the example of different OSPF control packets.*

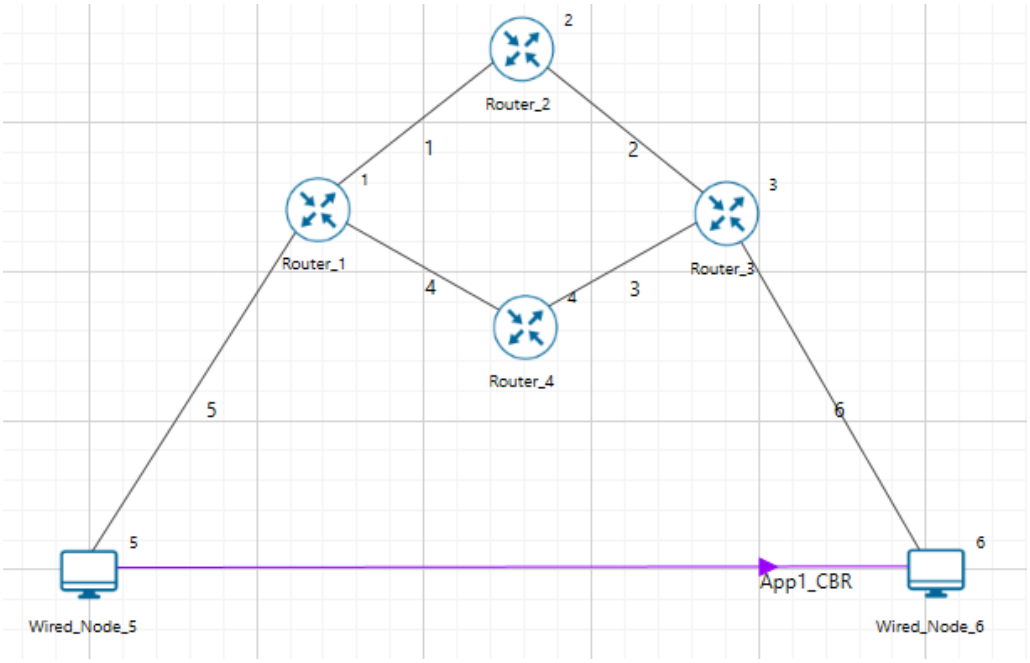



**Figure 4-34:** *Network setup for studying the different OSPF control packets.*

### Network Settings

1. Set OSPF Routing protocol under Application Layer properties of a router by clicking on the router and expanding the property panel in right.
2. Configure CBR application from set traffic tab from ribbon on the top. Click on created application and expand the application property panel on the right, set application Start Time(s) to 30 Sec by keeping other properties as default.
3. Enable Packet Trace from the configure reports tab in ribbon on the top.
4. Simulate for 100 sec.

### Results and Discussion

The OSPF neighbors are dynamically discovered by sending Hello packets out each OSPF-enabled interface on a router. Then Database description packets are exchanged when an adjacency is being initialized. They describe the contents of the topological database. After exchanging Database Description packets with a neighboring router, a router may find that parts of its topological database are out of date. The Link State Request packet is used to request the pieces of the neighbor's database that are more up to date. The sending of Link State Request packets is the last step in bringing up an adjacency. A packet that contains fully detailed LSAs, typically sent in response to an LSR message. LSAck is sent to confirm receipt of an LSU message.

The same can be observed in Packet trace. Open packet trace from simulation results window and set CONTROL PACKET TYPE/ APP NAME to OSPF HELLO, OSPF DD, OSPF LSACK, OSPF LSUPDATE and OSPF LSREQ packets as shown below in Figure 4-35.

| PACKET_ID | SEGMENT_ID | PACKET_TYPE | CONTROL_PACKET_TYPE/APP_NAME | SOURCE_ID | DESTINATION_ID | TRANSMITTER_ID | RECEIVER_ID | AP |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-1 | Broadcast-0 | ROUTER-1 | ROUTER-2 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-1 | Broadcast-0 | ROUTER-1 | ROUTER-4 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-2 | Broadcast-0 | ROUTER-2 | ROUTER-1 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-2 | Broadcast-0 | ROUTER-2 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-3 | Broadcast-0 | ROUTER-3 | ROUTER-2 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-3 | Broadcast-0 | ROUTER-3 | ROUTER-4 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-4 | Broadcast-0 | ROUTER-4 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-4 | Broadcast-0 | ROUTER-4 | ROUTER-1 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-1 | Broadcast-0 | ROUTER-1 | ROUTER-2 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-1 | Broadcast-0 | ROUTER-1 | ROUTER-4 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-2 | Broadcast-0 | ROUTER-2 | ROUTER-1 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-2 | Broadcast-0 | ROUTER-2 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-3 | Broadcast-0 | ROUTER-3 | ROUTER-2 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-3 | Broadcast-0 | ROUTER-3 | ROUTER-4 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-4 | Broadcast-0 | ROUTER-4 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-4 | Broadcast-0 | ROUTER-4 | ROUTER-1 | |
| 0 | 0 | Control_Packet | OSPF_DD | ROUTER-2 | ROUTER-1 | ROUTER-2 | ROUTER-1 | |
| 0 | 0 | Control_Packet | OSPF_DD | ROUTER-4 | ROUTER-1 | ROUTER-4 | ROUTER-1 | |
| 0 | 0 | Control_Packet | OSPF_DD | ROUTER-1 | ROUTER-2 | ROUTER-1 | ROUTER-2 | |
| 0 | 0 | Control_Packet | OSPF_DD | ROUTER-3 | ROUTER-2 | ROUTER-3 | ROUTER-2 | |
| 0 | 0 | Control_Packet | OSPF_DD | ROUTER-2 | ROUTER-3 | ROUTER-2 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_DD | ROUTER-4 | ROUTER-3 | ROUTER-4 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_DD | ROUTER-1 | ROUTER-2 | ROUTER-1 | ROUTER-2 | |
| 0 | 0 | Control_Packet | OSPF_DD | ROUTER-1 | ROUTER-4 | ROUTER-1 | ROUTER-4 | |
| 0 | 0 | Control_Packet | OSPF_DD | ROUTER-2 | ROUTER-3 | ROUTER-2 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_DD | ROUTER-3 | ROUTER-4 | ROUTER-3 | ROUTER-4 | |
| 0 | 0 | Control_Packet | OSPF_LSREQ | ROUTER-1 | ROUTER-2 | ROUTER-1 | ROUTER-2 | |
| 0 | 0 | Control_Packet | OSPF_LSREQ | ROUTER-1 | ROUTER-4 | ROUTER-1 | ROUTER-4 | |
| 0 | 0 | Control_Packet | OSPF_LSREQ | ROUTER-2 | ROUTER-3 | ROUTER-2 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_LSREQ | ROUTER-3 | ROUTER-4 | ROUTER-3 | ROUTER-4 | |
| 0 | 0 | Control_Packet | OSPF_LSREQ | ROUTER-2 | ROUTER-1 | ROUTER-2 | ROUTER-1 | |
| 0 | 0 | Control_Packet | OSPF_LSREQ | ROUTER-4 | ROUTER-1 | ROUTER-4 | ROUTER-1 | |
| 0 | 0 | Control_Packet | OSPF_LSREQ | ROUTER-3 | ROUTER-2 | ROUTER-3 | ROUTER-2 | |
| 0 | 0 | Control_Packet | OSPF_LSREQ | ROUTER-4 | ROUTER-3 | ROUTER-4 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-2 | Broadcast-0 | ROUTER-2 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-3 | Broadcast-0 | ROUTER-3 | ROUTER-4 | |
| 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-4 | Broadcast-0 | ROUTER-4 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-1 | Broadcast-0 | ROUTER-1 | ROUTER-4 | |
| 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-3 | Broadcast-0 | ROUTER-3 | ROUTER-2 | |
| 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-1 | Broadcast-0 | ROUTER-1 | ROUTER-2 | |
| 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-2 | Broadcast-0 | ROUTER-2 | ROUTER-1 | |
| 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-4 | Broadcast-0 | ROUTER-4 | ROUTER-1 | |
| 0 | 0 | Control_Packet | OSPF_LSACK | ROUTER-1 | ROUTER-4 | ROUTER-1 | ROUTER-4 | |
| 0 | 0 | Control_Packet | OSPF_LSACK | ROUTER-4 | Broadcast-0 | ROUTER-4 | ROUTER-3 | |
| 0 | 0 | Control_Packet | OSPF_LSACK | ROUTER-4 | Broadcast-0 | ROUTER-4 | ROUTER-1 | |

**Figure 4-35:** *Different OSPF control packets in the Packet Trace.*

## 4.8 Configuring Static Routing in NetSim

### Static Routing

Routers forward packets using either route information from route table entries that are configured manually or the route information that is calculated using dynamic routing algorithms. Static routes, which define explicit paths between two routers, cannot be automatically updated; you must manually reconfigure static routes when network changes occur. Static routes use less bandwidth than dynamic routes. No CPU cycles are used to calculate and analyze routing updates.

Static routes are used in environments where network traffic is predictable and where the network design is simple. You should not use static routes in large, constantly changing networks because static routes cannot react to network changes. Most networks use dynamic routes to communicate between routers but might have one or two static routes configured for special cases. Static routes are also useful for specifying a gateway of last resort (a default router to which all unrouteable packets are sent).

Note that the static route configuration running with TCP protocol requires reverse route configuration.

### How to Setup Static Routes

In NetSim, static routes can be configured either prior to the simulation or during the simulation.

Static route configuration prior to simulation:

1. Via static route GUI configuration
2. Via file input (Interactive-Simulation/SDN)

Static route configuration during the simulation:

3. Via device NetSim Console (Interactive-Simulation/ SDN)

### Static route configuration via GUI

Open NetSim, Select Examples ▷ Internetworks ▷ Configuring Static Route then click on the tile in the middle panel to load the example as shown in Figure 4-36.

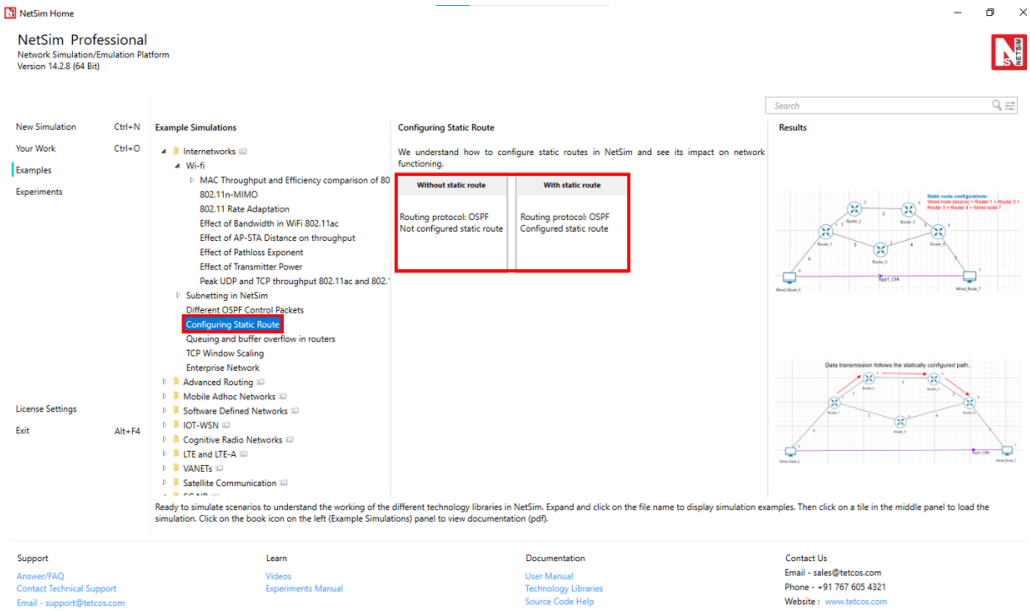



**Figure 4-36:** *List of scenarios for the example of Configuring Static Route.*

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file for Configuring Static Routing in NetSim as shown in Figure 4-37.

### 4.8.1 Without Static Route

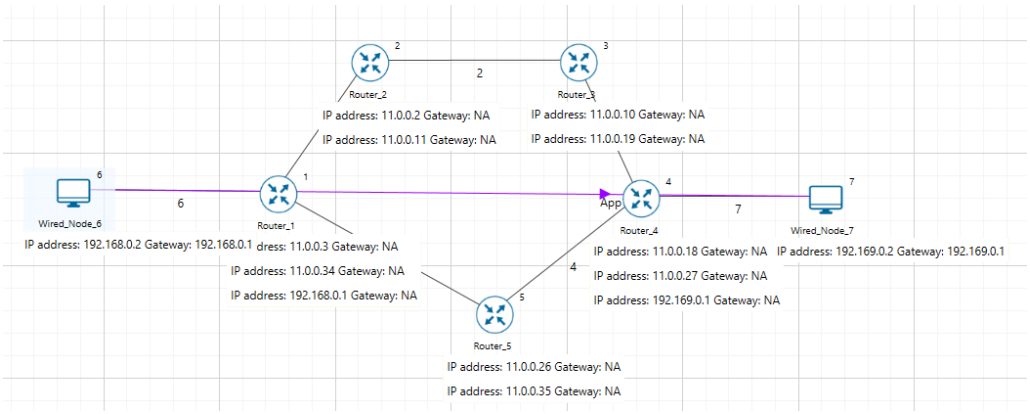



**Figure 4-37:** *Network setup for studying static route configuration.*

#### Network Settings

1. Set grid length as 500m × 250m from grid setting property panel on the right. This needs to be done before any device is placed on the grid.
2. Create a scenario as shown in the above screenshot.
3. The default routing protocol is OSPF in application layer of routers by clicking on the router and expanding the property panel on the right.
4. Wired link properties are default.
5. Configure CBR application between wired node 6 and 7 by clicking on the set traffic tab from the ribbon at the top. Click on created application and expand the application property panel on the right and set transport protocol to UDP.
6. Enable packet trace from configure reports tab in ribbon on the top.
7. Run a simulation for 10 seconds.
8. In packet trace, filter the CONTROL PACKET TYPE to APP1 CBR and observe the packet flow from Wired Node 6 ▹ Router 1 ▹ Router 5 ▹ Router 4 ▹ Wired Node 7 as shown in Figure 4-38.

| 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-6 | NODE-7 | NODE-6 | ROUTER-1 |
| 2 | 0 | CBR | App1_CBR | NODE-6 | NODE-7 | NODE-6 | ROUTER-1 |
| 3 | 0 | CBR | App1_CBR | NODE-6 | NODE-7 | NODE-6 | ROUTER-1 |
| 3 | 0 | CBR | App1_CBR | NODE-6 | NODE-7 | ROUTER-1 | ROUTER-5 |
| 3 | 0 | CBR | App1_CBR | NODE-6 | NODE-7 | ROUTER-5 | ROUTER-4 |
| 3 | 0 | CBR | App1_CBR | NODE-6 | NODE-7 | ROUTER-4 | NODE-7 |

**Figure 4-38:** *Packet flows from Wired Node 6 ▹ Router 1 ▹ Router 5 ▹ Router 4 ▹ Wired Node 7.*

### 4.8.2 With Static Route

#### Static routing configuration

1. Open Router 1 property ▷ Network Layer, Enable Static IP Route ▷ Click on via GUI and set the properties as per the screenshot below and click on Add and then click on OK.

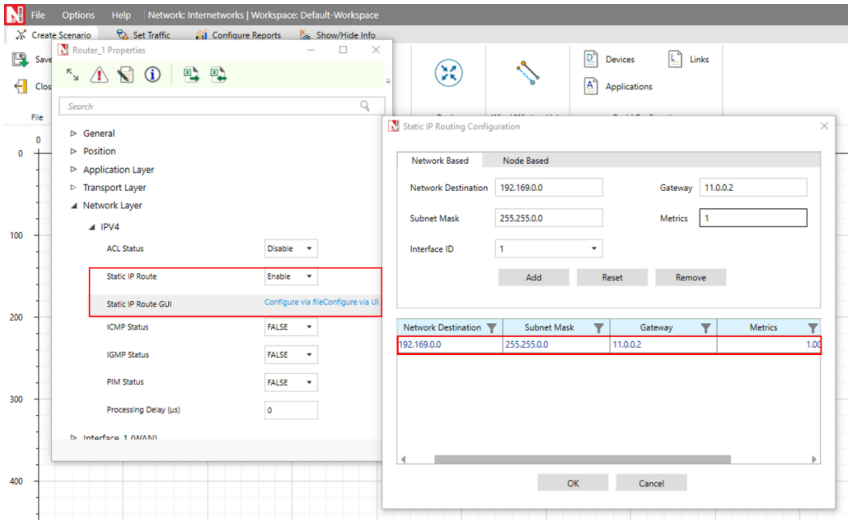



**Figure 4-39:** *Static IP Routing Dialogue window.*

This creates a text file for every router in the temp path of NetSim which is in the format below:

```
Router 1:
route ADD 192.169.0.0 MASK 255.255.0.0 11.0.0.2 METRIC 1 IF 1
route ADD destinationip MASK subnet_mask gateway_ip METRIC metric_value IF Interface_Id
```

where

route ADD: command to add the static route.

destination_ip: is the Network address for the destination network.

MASK: is the Subnet mask for the destination network.

gateway_ip: is the IP address of the next-hop router/node.

METRIC: is the value used to choose between two routes.

IF: is the Interface to which the gateway_ip is connected. The default value is 1.

1. Similarly Configure Static Route for all the routers as given in Table 4-23.

**Table 4-23:** *Static Route configuration for routers.*

| Devices | Network Destination | Gateway | Subnet Mask | Metrics | Interface ID |
|---|---|---|---|---|---|
| Router 1 | 192.169.0.0 | 11.0.0.2 | 255.255.0.0 | 1 | 1 |
| Router 2 | 192.169.0.0 | 11.0.0.10 | 255.255.0.0 | 1 | 2 |
| Router 3 | 192.169.0.0 | 11.0.0.18 | 255.255.0.0 | 1 | 2 |
| Router 4 | 192.169.0.0 | 192.169.0.2 | 255.255.0.0 | 1 | 3 |

2. After configuring the above router properties run the simulation for 10 seconds.

3. In packet trace, filter the CONTROL PACKET TYPE to APP1 CBR and observe the change in the packet flow from Wired Node 6 ▹ Router 1 ▹ Router 2 ▹ Router 3 ▹ Router 4 ▹ Wired Node 7 due to static route configurations as shown in Figure 4-40.

| 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-6 | NODE-7 | NODE-6 | ROUTER-1 |
| 1 | 0 | CBR | App1_CBR | NODE-6 | NODE-7 | ROUTER-1 | ROUTER-2 |
| 1 | 0 | CBR | App1_CBR | NODE-6 | NODE-7 | ROUTER-2 | ROUTER-3 |
| 1 | 0 | CBR | App1_CBR | NODE-6 | NODE-7 | ROUTER-3 | ROUTER-4 |
| 1 | 0 | CBR | App1_CBR | NODE-6 | NODE-7 | ROUTER-4 | NODE-7 |

**Figure 4-40:** *Packet trace shows the data flow after configuring static routes.*

### Disabling Static Routing

4. If static routes were configured via GUI, it can be manually removed prior to the simulation from the Static IP Routing Dialogue or from the file input.

5. If static routes were configured during the run time, the entries can be deleted using the route delete command during runtime.

## 4.9 TCP Window Scaling

Open NetSim, Select Examples ▹ Internetworks ▹ TCP Window Scaling then click on the tile in the middle panel to load the example as shown in Figure 4-41.

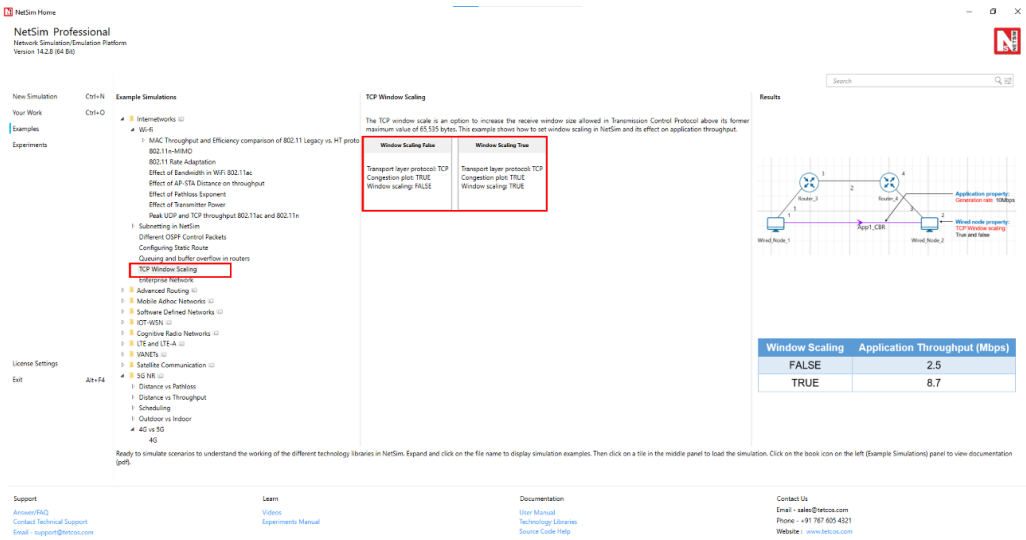



**Figure 4-41:** *List of scenarios for the example of TCP Window Scaling.*

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file for TCP Window scaling as shown in Figure 4-42.

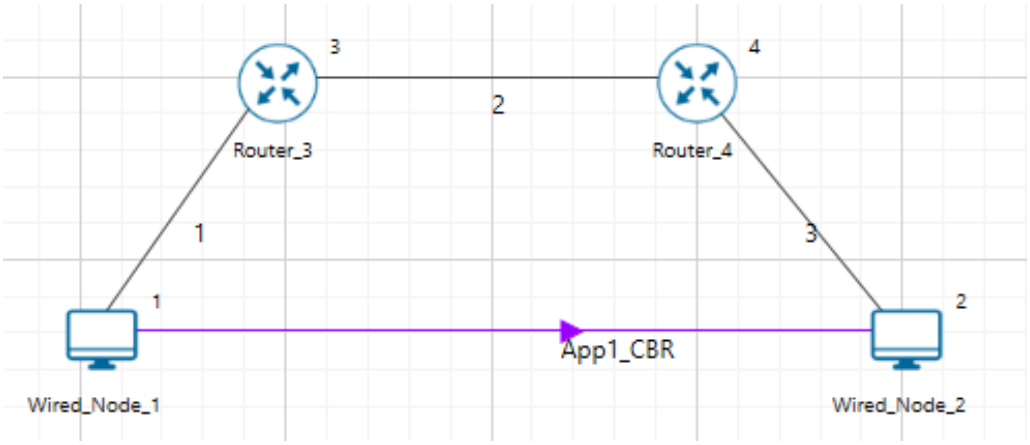



**Figure 4-42:** *Network setup for studying the TCP Window Scaling.*

The TCP throughput of a link is limited by two windows: the congestion window and the receive window. The congestion window tries not to exceed the capacity of the network; the receive window tries not to exceed the capacity of the receiver to process data.

The TCP window scale option is an option to increase the receive window size allowed in Transmission Control Protocol above its former maximum value of 65,535 bytes.

The TCP window scale option is needed for efficient transfer of data when the bandwidth-delay product is greater than 64K. For instance, if a transmission line of 1.5 Mbit/second was used over a satellite link with a 513 milliseconds round trip time (RTT), the bandwidth-delay product is $1{,}500{,}000 \times 0.513 = 769{,}500$ bits or about 96,187 bytes.

Using a maximum window size of 64 KB only allows the buffer to be filled to $\frac{65535}{96187} = 68\%$ of the theoretical maximum speed of 1.5 Mbps, or 1.02 Mbps.

By using the window scale option, the receive window size may be increased up to a maximum value of 1,073,725,440 bytes. This is done by specifying a one-byte shift count in the header options field. The true receive window size is left shifted by the value in shift count. A maximum value of 14 may be used for the shift count value. This would allow a single TCP connection to transfer data over the example satellite link at 1.5 Mbps utilizing all of the available bandwidth.

### Network Settings

1. Create the scenario as shown in the above screenshot.
2. To configure the properties below in the nodes, click on the node, expand the property panel on the right side, and change the properties as mentioned in the below steps.
3. Set TCP Window Scaling to FALSE (by default).
4. Enable Wireshark Capture as offline under General Properties of Wired Node 1.
5. Configure CBR application between two nodes by clicking on the set traffic tab from the ribbon on the top. Click on the created application and expand the application property panel on the right, packet size to 1460 and Inter arrival time to 1168 $\mu$s (Generation rate = 10 Mbps).
6. To configure wired and wireless link properties, click on link, expand the property panel on the right and set wired link properties as shown below.

7. Set Bit error rate (Uplink and Downlink) to 0 in all wired links.
8. Link1 & Link3 Propagation delay (uplink and downlink) to 5 (Microsec) (by default).
9. Change the Link2 speed to 10 Mbps, Propagation delay (uplink and downlink) to 100000 (Microsec).
10. Simulate for 100s and note down the throughput.
11. Now change the Window Scaling to TRUE (for all wired nodes).
12. Enable Window Size vs Time plot under TCP Congestion window from the Plots tab located in the right panel.

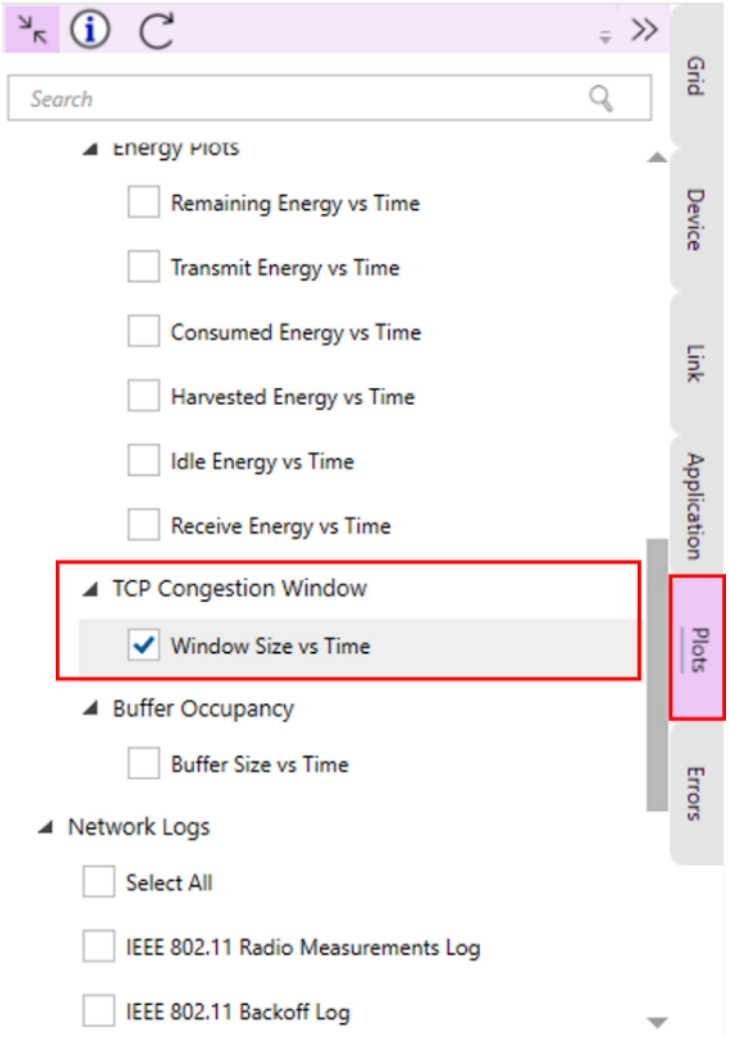



**Figure 4-43:** *Enabling TCP Congestion plot.*

Simulate for 100s and note down the throughput.

**Results and Discussion**

**Table 4-24:** *Results comparison for with/without Window Scaling.*

| Window Scaling | Application Throughput (Mbps) |
|---|---|
| FALSE | 2.5 |
| TRUE | 8.7 |

**Throughput calculation (Without Window Scaling):**

Theoretical Throughput = Window size / Round trip time = $\frac{65525 \times 8}{200\text{ms}}$ = 2.62 Mbps.

Go to the simulation result window ▷ plots ▷ TCP Congestion Window Plot (Figure 4-44 and Figure 4-45).

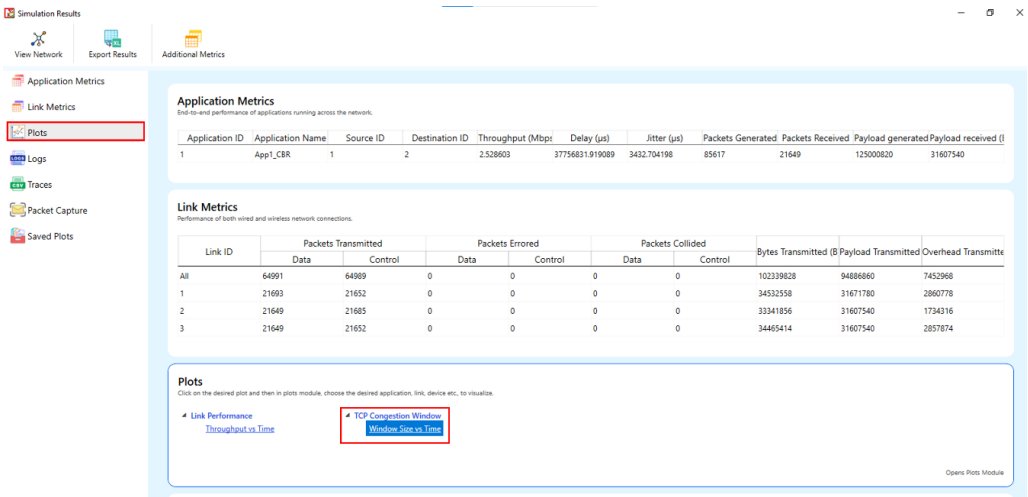



**Figure 4-44:** *Open TCP Congestion Window plot from the result dashboard.*

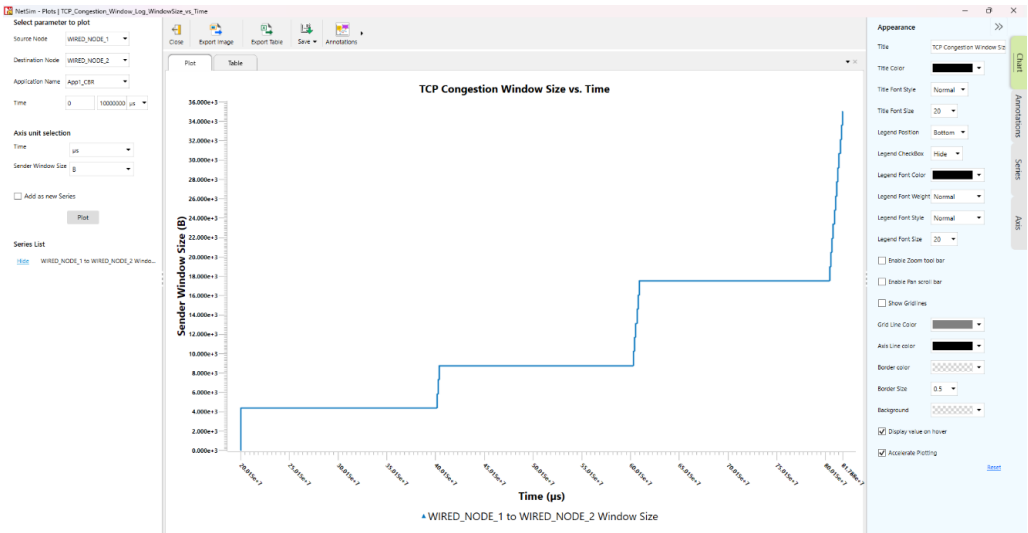



**Figure 4-45:** *TCP Congestion Window Plot for wired node 2 (Window Scaling FALSE).*

In Window Scaling False, the Application Throughput is 2.5 Mbps less than the theoretical throughput since it initially takes some time for the window to reach 65535 B.

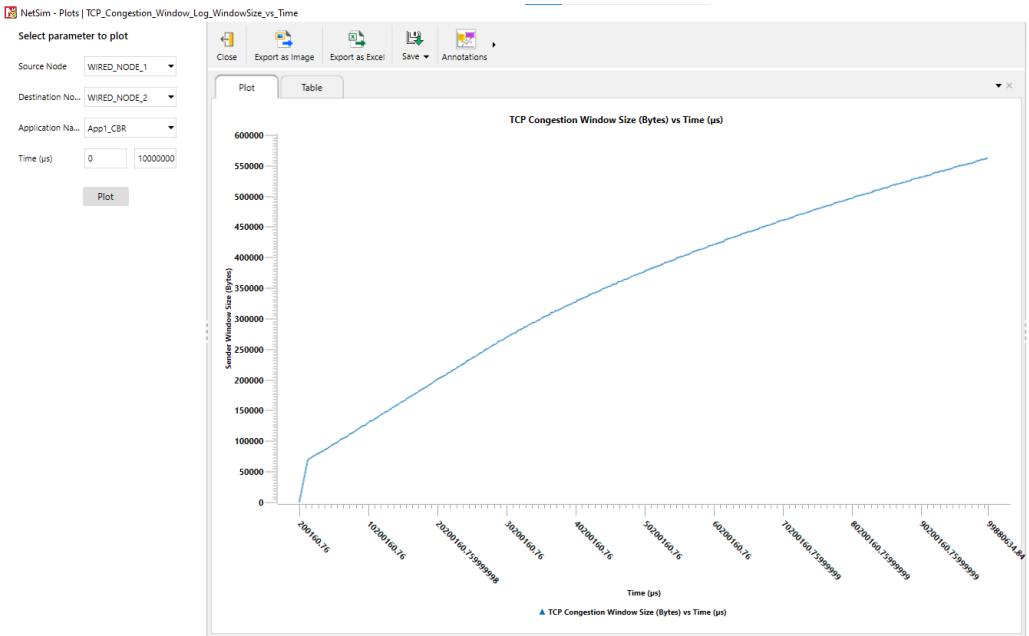



**Figure 4-46:** *TCP Congestion Window Plot for wired node 2 (Window Scaling TRUE).*

In Window Scaling TRUE, from the above screenshot, users can notice that the window size grows up to 560192 Bytes because of Window Scaling. This leads to a higher Application throughput compared to the case without window scaling.

We have enabled Wireshark Capture in the Wired Node 1. The PCAP file is generated silently at the end of the simulation. Double-click on the WIRED NODE1_1.pcap file available in the result window under packet captures. In Wireshark, the window scaling graph can be obtained as follows. Select any data packet with a left click, then, go to Statistics ▷ TCP Stream Graphs ▷ Window Scaling ▷ Select Switch Direction.

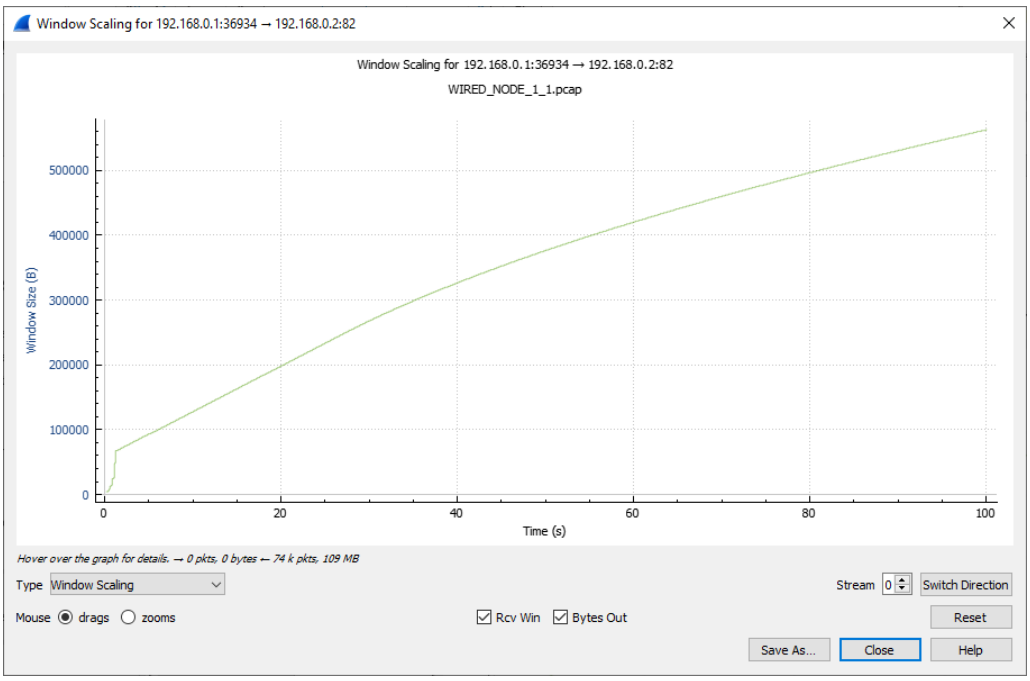



**Figure 4-47:** *Wireshark Window when Window Scaling is TRUE.*

## 4.10 An enterprise network comprising different subnets and running various applications

We consider a simple enterprise network, comprising of two branches, headquarters and a data center. Branches and headquarters are connected to the data center over the public cloud. Branch 1 has 10 systems, branch 2 has 10 systems, HQ has 5 systems, and they connect to a data center which houses a DB server, an email server, and an FTP server.

Open NetSim, Select Examples ▷ Internetworks ▷ Enterprise Network then click on the tile in the middle panel to load the example as shown in Figure 4-48.

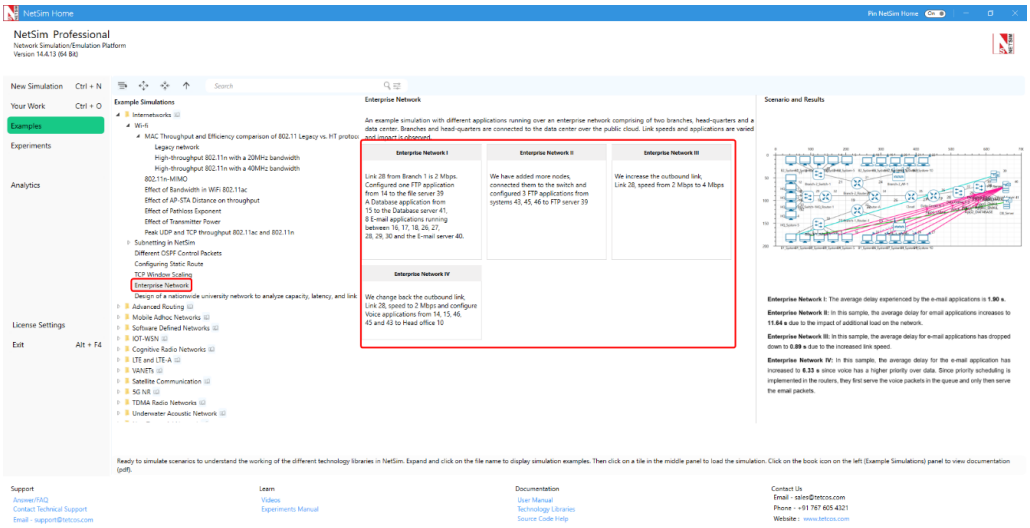



**Figure 4-48:** *List of scenarios for the example of enterprise networks.*

The following network diagram illustrates what the NetSim UI displays when you open the example configuration file for Enterprise Network in NetSim as shown in Figure 4-49.

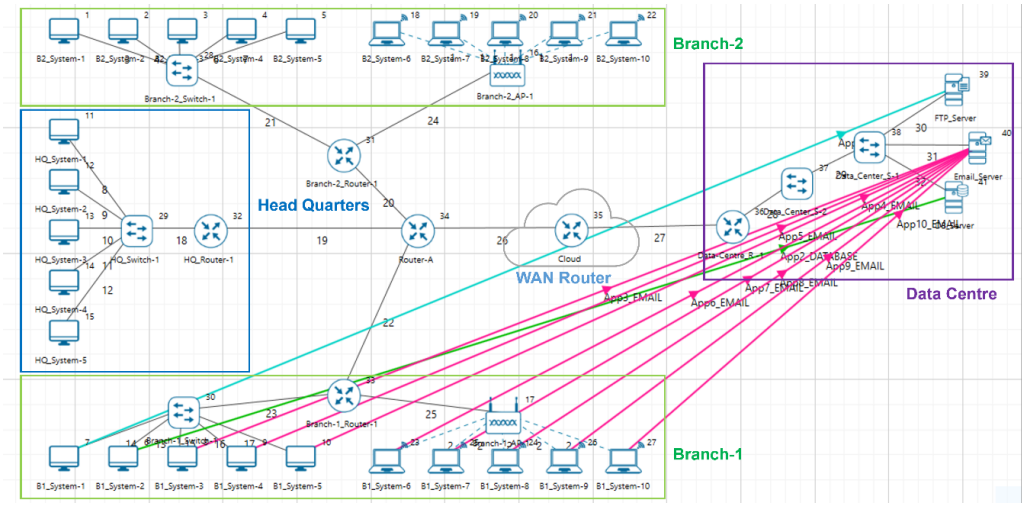



**Figure 4-49:** *Network setup for studying the enterprise network. Labelling (Branch-1, Branch-2, HQ, Data center) has been added to the screen shot. Links 29, 30 and the WAN Router can be thought of as the internet cloud over which traffic flows to reach the data center.*

### 4.10.1 Network Settings

#### Enterprise Network I

1. Link rate for the outbound link i.e., link 28 from Branch 1 is set to 2 Mbps. Link properties can be set by clicking on the click, expanding the link properties panel on the right.

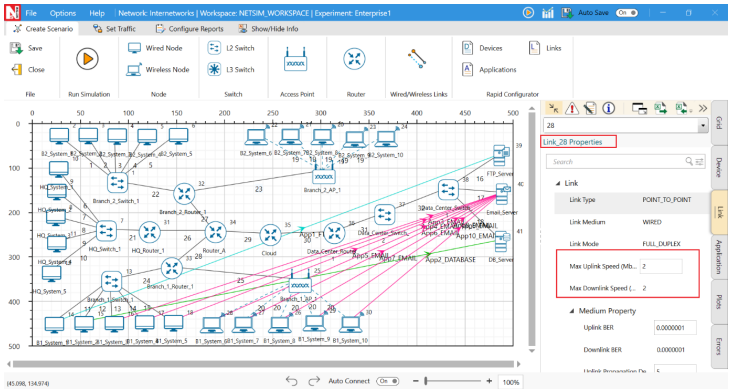


**Figure 4-50:** *Setting link speed for outbound link (link 28).*

2. Configure one FTP application from 7 to the file server 39, a DB application from 6 to the Database server 41, and eight email applications running between 8, 9, 10, 23, 25, 24, 26, 27 and the Email server 40. Refer to 3.9.3.2 section in user manual to configure multiple applications using rapid configurator.

3. Run the simulation for 100 s.

#### Enterprise Network II

In this sample, we add more nodes via the switch and configured 3 FTP applications from systems 11, 12, 13 i.e. Wired node 43, 44 and 45 to FTP server 39 as shown in Figure 4-51 by keeping the Link rate for the outbound link (link 28) as 2 Mbps.

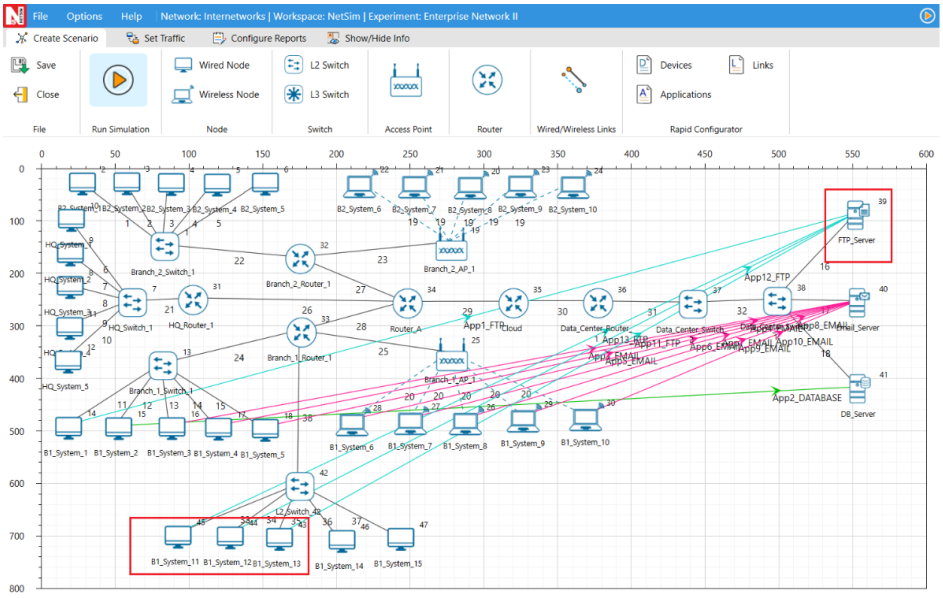



**Figure 4-51:** *Configuring FTP applications from systems 43, 44, 45 to FTP server 39.*

Simulate for 100 seconds.

### Enterprise Network III

In this sample, we change the outbound link speed i.e., Link 28 to 4 Mbps and simulate for 100 seconds.

### Enterprise Network IV

In this sample, we change the outbound link speed i.e., Link 28 to 2 Mbps and configure Voice applications from 14, 15, 43, 44 and 45 to Head office 10 as shown in Figure 4-52.

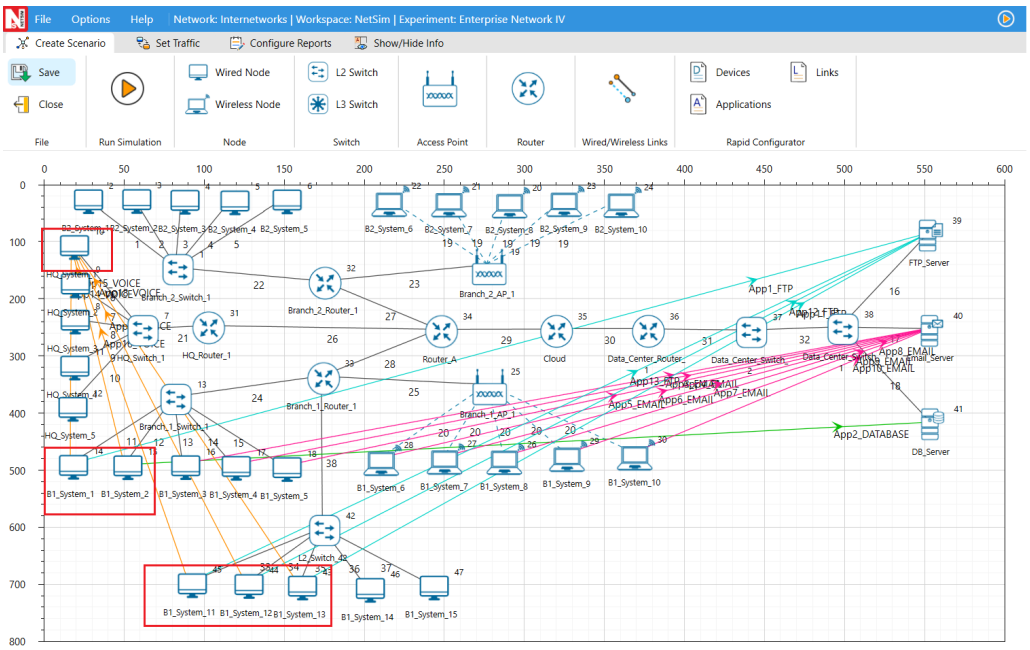



**Figure 4-52:** *Configuring voice applications from 14, 15, 43, 44 and 45 to Head office 10.*

1. Also, Scheduling type is set to Priority under Network Layer Properties of Router 38 Interface WAN properties as shown in Figure 4-53.

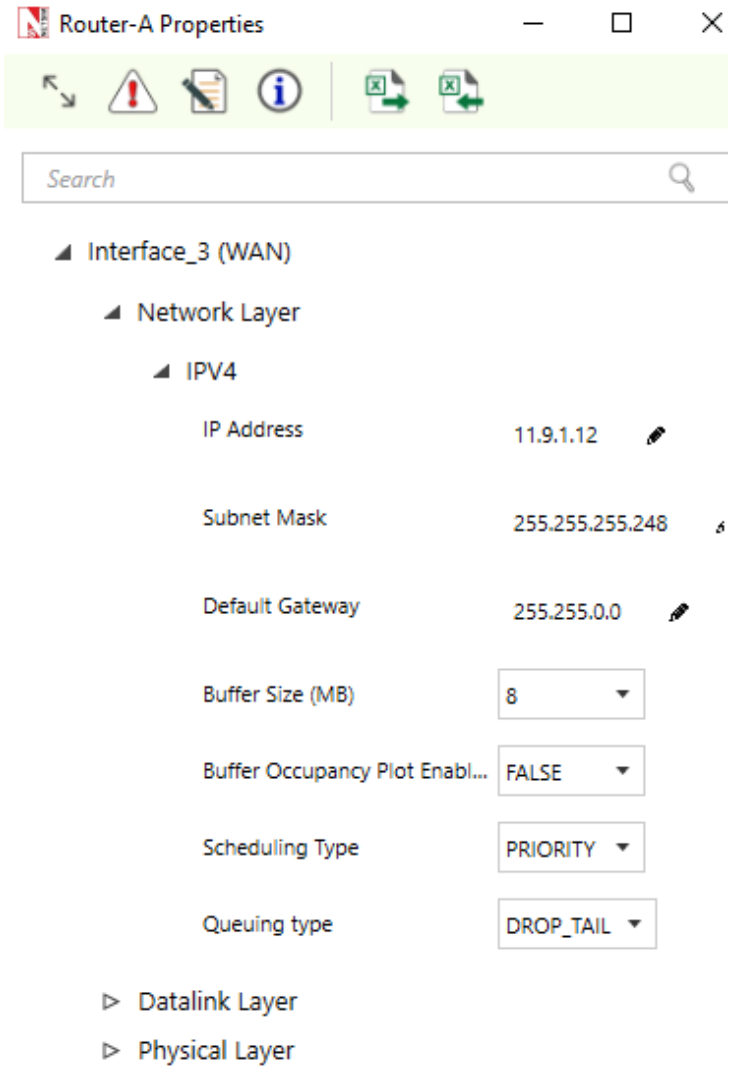
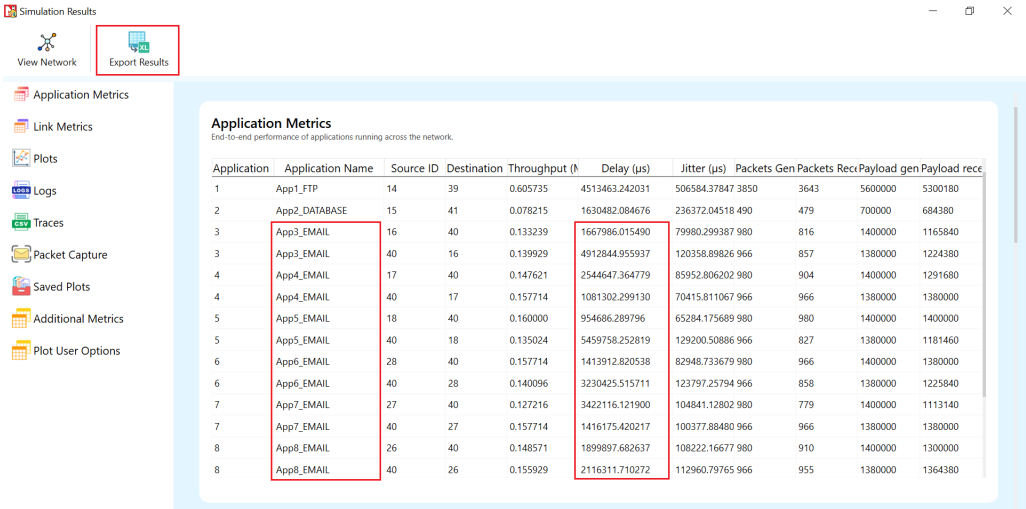



**Figure 4-53:** *WAN Interface – Network layer properties window.*

Simulate for 100 seconds.

### 4.10.2 Results and Observations

**Results and Discussion**




| Application | Application Name | Source ID | Destination | Throughput (M | Delay (µs) | Jitter (µs) | Packets Gen | Packets Rec | Payload gen | Payload rece |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | App1_FTP | 14 | 39 | 0.605735 | 4513463.242031 | 506584.37847 | 3850 | 3643 | 5600000 | 5300180 |
| 2 | App2_DATABASE | 15 | 41 | 0.078215 | 1630482.084676 | 236372.04518 | 490 | 479 | 700000 | 684380 |
| 3 | App3_EMAIL | 16 | 40 | 0.133239 | 1667986.015490 | 79980.299387 | 980 | 816 | 1400000 | 1165840 |
| 3 | App3_EMAIL | 40 | 16 | 0.139929 | 4912844.955937 | 120358.89826 | 966 | 857 | 1380000 | 1224380 |
| 4 | App4_EMAIL | 17 | 40 | 0.147621 | 2544647.364779 | 85952.806202 | 980 | 904 | 1400000 | 1291680 |
| 4 | App4_EMAIL | 40 | 17 | 0.157714 | 1081302.299130 | 70415.811067 | 966 | 966 | 1380000 | 1380000 |
| 5 | App5_EMAIL | 18 | 40 | 0.160000 | 954686.289796 | 65284.175689 | 980 | 980 | 1400000 | 1400000 |
| 5 | App5_EMAIL | 40 | 18 | 0.135024 | 5459758.252819 | 129200.50886 | 966 | 827 | 1380000 | 1181460 |
| 6 | App6_EMAIL | 28 | 40 | 0.157714 | 1413912.820538 | 82948.733679 | 980 | 966 | 1400000 | 1380000 |
| 6 | App6_EMAIL | 40 | 28 | 0.140096 | 3230425.515711 | 123797.25794 | 966 | 858 | 1380000 | 1225840 |
| 7 | App7_EMAIL | 27 | 40 | 0.127216 | 3422116.121900 | 104841.12802 | 980 | 779 | 1400000 | 1113140 |
| 7 | App7_EMAIL | 40 | 27 | 0.157714 | 1416175.420217 | 100377.88480 | 966 | 966 | 1380000 | 1380000 |
| 8 | App8_EMAIL | 26 | 40 | 0.148571 | 1899897.682637 | 108222.16677 | 980 | 910 | 1400000 | 1300000 |
| 8 | App8_EMAIL | 40 | 26 | 0.155929 | 2116311.710272 | 112960.79765 | 966 | 955 | 1380000 | 1364380 |

**Figure 4-54:** *Application metrics table for Enterprise Network I.*

**Enterprise Network I:** In the simulation results window, observe the application metrics and calculate the average delay for email application as shown below. For earlier delay calculations, the user can also export the results by clicking on the export results option present in the simulation results window.

The average delay experienced by the e-mail applications is 2.69 s.

**Enterprise Network II:** In this sample, the average delay for email applications increases to 15.79 s due to the impact of additional load on the network.

**Enterprise Network III:** In this sample, the average delay for e-mail applications has dropped down to 0.68 s due to the increased link speed.

**Enterprise Network IV:** In this sample, the average delay for the e-mail application has increased to 5.73 s since voice has a higher priority over data. Since priority scheduling is implemented in the routers, they first serve the voice packets in the queue and only then serve the email packets.

## 4.11 Design of a nationwide university network to analyze capacity, latency, and link failures

### 4.11.1 Introduction

SWITCHlan is the national research and education network in Switzerland. It is operated by SWITCH, the Swiss National Research and Education Network organization. This network connects Swiss universities, research institutions, and other educational organizations, providing them with high-speed and reliable network connectivity. It allows Swiss institutions to benefit from global connectivity and participate in international research collaborations.

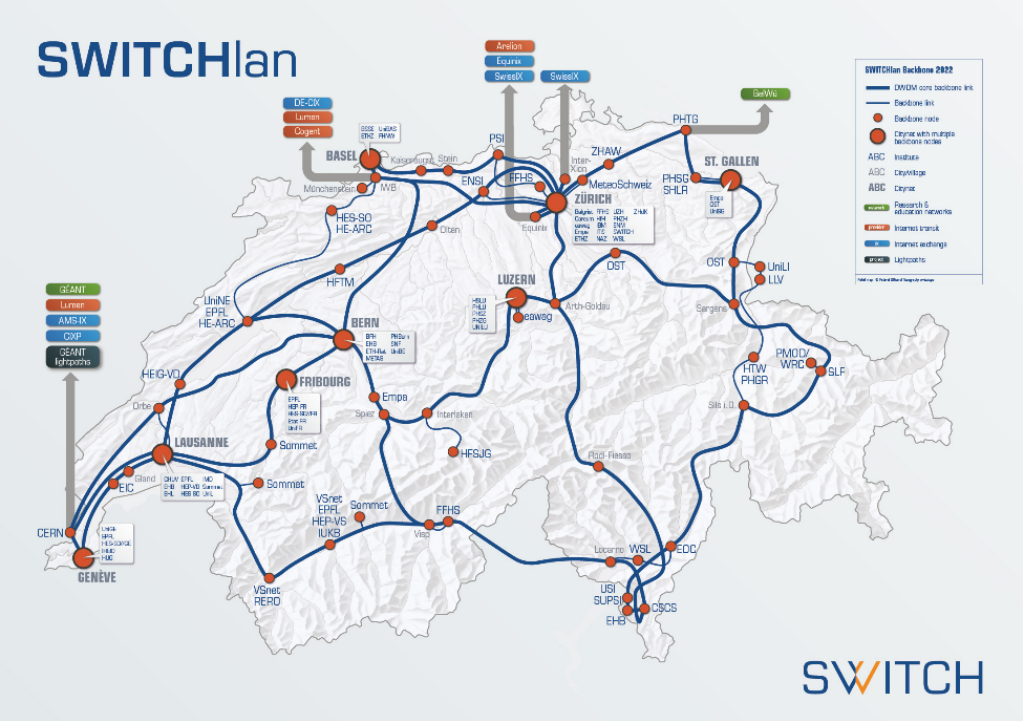



**Figure 4-55:** *The network connection between universities of different cities in Switzerland.*

NetSim allows you to model and simulate complex networks, including LANs and WANs, using different types of network devices. By abstracting SWITCHlan with routers and WAN links in NetSim, you can simulate various traffic loads, test different configurations, and evaluate the performance of the network under different conditions.

In this study, we simulate SWITCHlan by modeling six universities located in Lausanne, Zurich, Basel, Bern, Luzern, and Fribourg. Each university is represented by a router and a server. WAN links connect these routers, simulating the network topology of SWITCHlan.

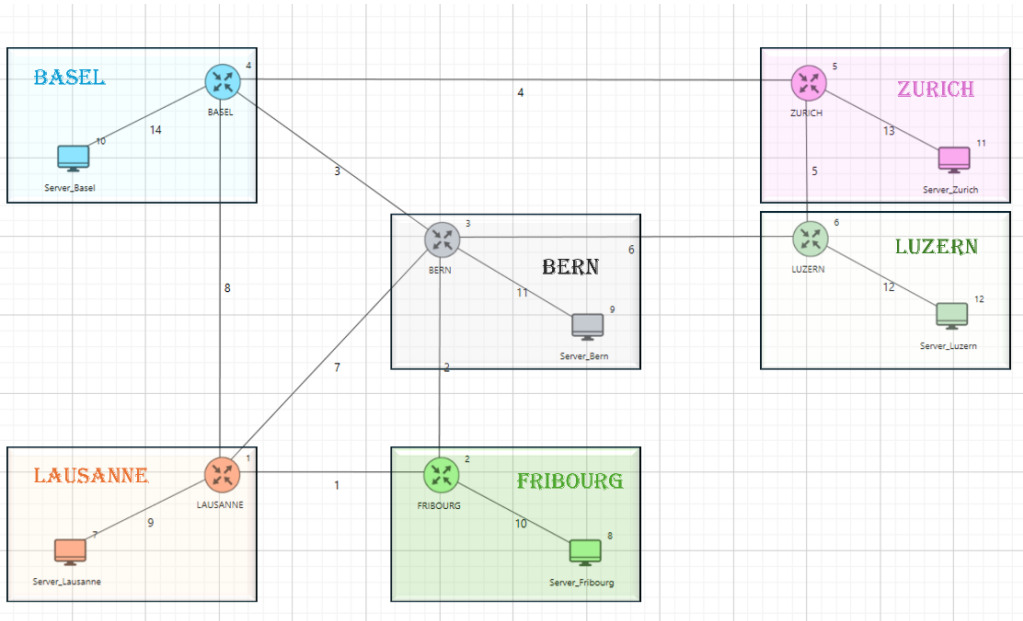



**Figure 4-56:** *Network Scenario in NetSim representing the different cities.*

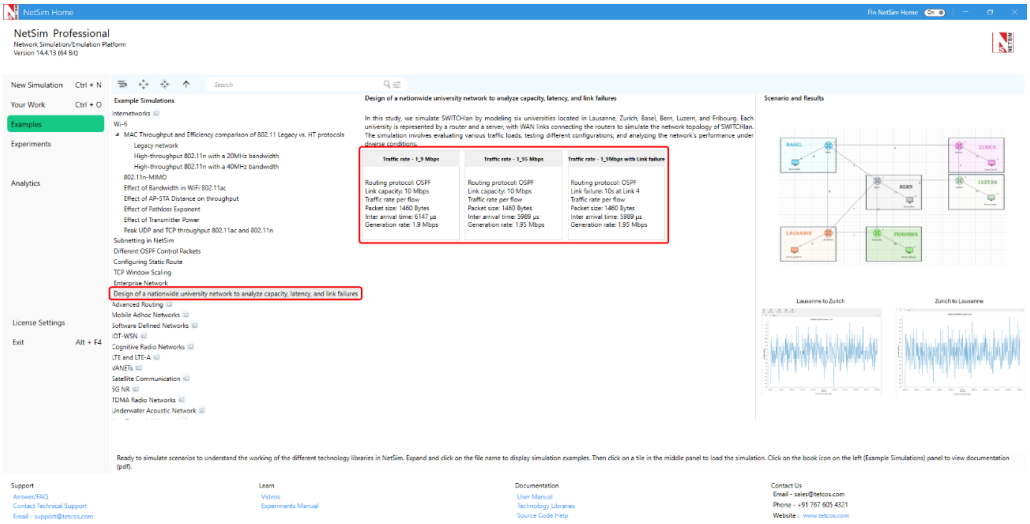



**Figure 4-57:** *List of scenarios for the example of design of nationwide university network to analyze capacity, latency, and link failures.*

Open NetSim, Select Examples ▷ Internetworks ▷ Design of a nationwide university network to analyze capacity, latency, and link failures then click on the tile in the middle panel to load the example shown in below.

### 4.11.2 Network Settings

1. Drop 6 routers and wired node (server) in NetSim design window as shown in Figure 4-58. This represents six different countries.
2. Connect the devices as shown in scenario and set the link capacity for all wired links to 10 Mbps.
3. Application layer routing protocol is set to OSPF in router.
4. Configure the custom application from each city to every other city (30 flows in total) to enable data transfer. Refer to Table 4-25 for application properties.
5. Enable Packet Trace from the configure reports tab in ribbon on the top.
6. Enable latency and throughput plots by clicking on the Plots/Logs tab in the right panel.
7. Simulate the scenario for 50 seconds.

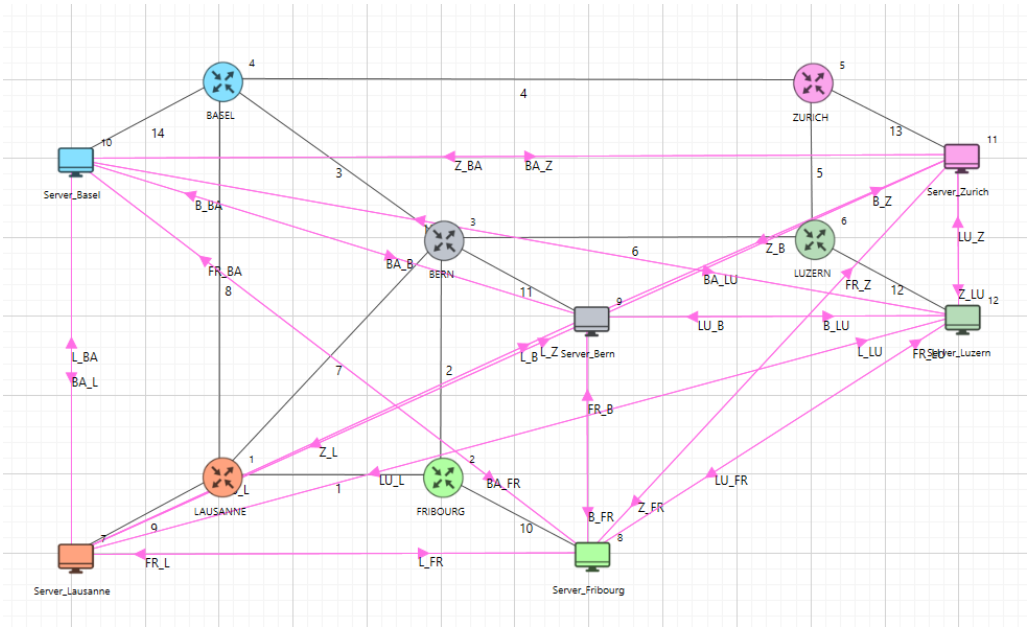



**Figure 4-58:** *This figure shows the traffic configuration between all cites. Since there are 6 cities, there is a total of* $6 * 5 = 30$ *traffic flows.*

### 4.11.3 Simulation cases and Application settings

#### Case 1: Regular Operation (Traffic Rate 1.9 Mbps)

A traffic rate of 1.9 Mbps is chosen because there are five flows passing through each bottleneck link (connecting a router to a server) in one direction. These five flows represent the traffic coming from and going to the other five cities. Consequently, the bottleneck flow rate is approximately 2 Mbps (10 Mbps/5). To avoid saturation, the traffic rate is set slightly below this capacity at 1.9 Mbps per flow. The parameter configuration is as follows:

**Table 4-25:** *Application settings for Case 1.*

| Parameter | Value |
|---|---|
| Application type | Custom |
| Transport Protocol | UDP |
| Traffic Configuration | UL & DL between all cities |
| Packet Size – Value | 1460 bytess |
| Packet Size – Distribution | Constant |
| Inter Arrival Time – Mean | 6147 |
| Inter Arrival Time – Distribution | Exponential |

#### Case 2: Increased Traffic Load (Traffic Rate 1.95 Mbps)

**Table 4-26:** *Application settings for Case 2.*

| Parameter | Value |
|---|---|
| Application type | Custom |
| Transport Protocol | UDP |
| Traffic Configuration | UL & DL between all cities |
| Packet Size – Value | 1460 bytes |
| Packet Size – Distribution | Constant |
| Inter Arrival Time – Mean | 5989 |
| Inter Arrival Time – Distribution | Exponential |

### Case 3: Link Failure (Basel–Zurich Link)

In this case, the traffic configuration is similar to Case 1 (1.9 Mbps). However, here we study the impact of a link failure by failing the link between Basel and Zurich at 10 seconds in the Link 4 properties, as shown below.

**Table 4-27:** *Application settings for Case 3.*

| Parameter | Value |
|---|---|
| Application type | Custom |
| Transport Protocol | UDP |
| Traffic Configuration | UL & DL between all cities |
| Packet Size – Value | 1460 bytes |
| Packet Size – Distribution | Constant |
| Inter Arrival Time – Mean | 6147 |
| Inter Arrival Time – Distribution | Exponential |

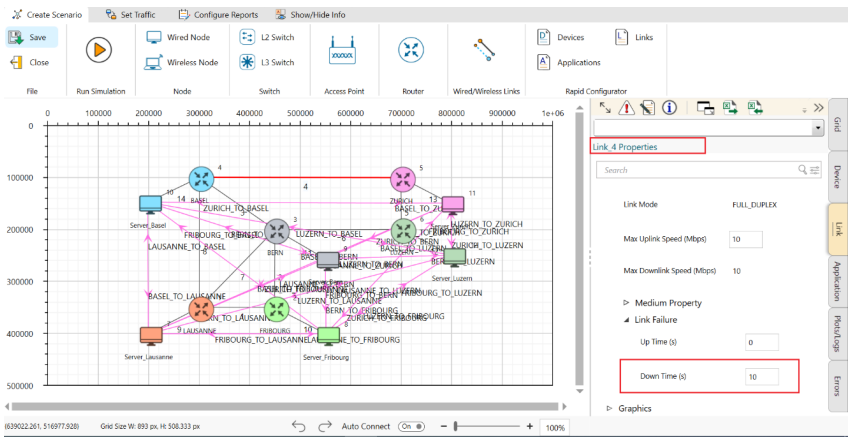



**Figure 4-59:** *Configuring the link failure for link 4 (between Basel and Zurich).*

Traffic is rerouted via alternate paths using the OSPF protocol.

## 4.11.4 Results and Observations

### Case 1: Regular Operation

In the application metrics, we observe that the total packets generated are 7,978, packets received are 7,922, and errored packets are 56.

**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. Recd. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | LAUSANNE_TO_FRIBOUI | 7 | 8 | 1.900000 | 1.900336 | 57913.862099 | 2575.951734 | 8177 | 8135 |
| 2 | FRIBOURG_TO_LAUSANI | 8 | 7 | 1.900000 | 1.872070 | 47403.867702 | 2522.746863 | 8048 | 8014 |
| 3 | LAUSANNE_TO_BERN | 7 | 9 | 1.900000 | 1.921360 | 73431.118981 | 2523.905009 | 8281 | 8225 |
| 4 | BERN_TO_LAUSANNE | 9 | 7 | 1.900000 | 1.828387 | 47017.674686 | 2563.579293 | 7856 | 7827 |
| 5 | LAUSANNE_TO_BASEL | 7 | 10 | 1.900000 | 1.886554 | 64766.053382 | 2604.948328 | 8143 | 8076 |
| 6 | BASEL_TO_LAUSANNE | 10 | 7 | 1.900000 | 1.929069 | 60884.760979 | 2660.082100 | 8295 | 8258 |
| 7 | LAUSANNE_TO_ZURICH | 7 | 11 | 1.900000 | 1.850579 | 84564.142414 | 2887.235511 | 7978 | 7922 |
| 8 | ZURICH_TO_LAUSANNE | 11 | 7 | 1.900000 | 1.846842 | 91179.197543 | 2851.962758 | 7955 | 7906 |
| 9 | LAUSANNE_TO_LUZERN | 7 | 12 | 1.900000 | 1.889824 | 69037.895396 | 2639.726582 | 8143 | 8090 |
| 10 | LUZERN_TO_LAUSANNE | 12 | 7 | 1.900000 | 1.899635 | 53394.137147 | 2608.543649 | 8173 | 8132 |
| 11 | FRIBOURG_TO_BERN | 8 | 9 | 1.900000 | 1.881648 | 62341.002288 | 2571.254927 | 8088 | 8055 |
| 12 | BERN_TO_FRIBOURG | 9 | 8 | 1.900000 | 1.880013 | 46217.421115 | 2500.707566 | 8080 | 8048 |
| 13 | FRIBOURG_TO_BASEL | 8 | 10 | 1.900000 | 1.870202 | 55345.704541 | 2650.936804 | 8068 | 8006 |
| 14 | BASEL_TO_FRIBOURG | 10 | 8 | 1.900000 | 1.849645 | 62562.698062 | 2723.282181 | 7978 | 7918 |

**Figure 4-60:** *Results obtained for Case 1 in Application Metrics.*

The mean generation rate is 1.90 Mbps, and the obtained throughput is 1.85 Mbps. The small difference is due to the errored packets.

Now click on plots from Left hand side of result dashboard to access throughput and latency plots.

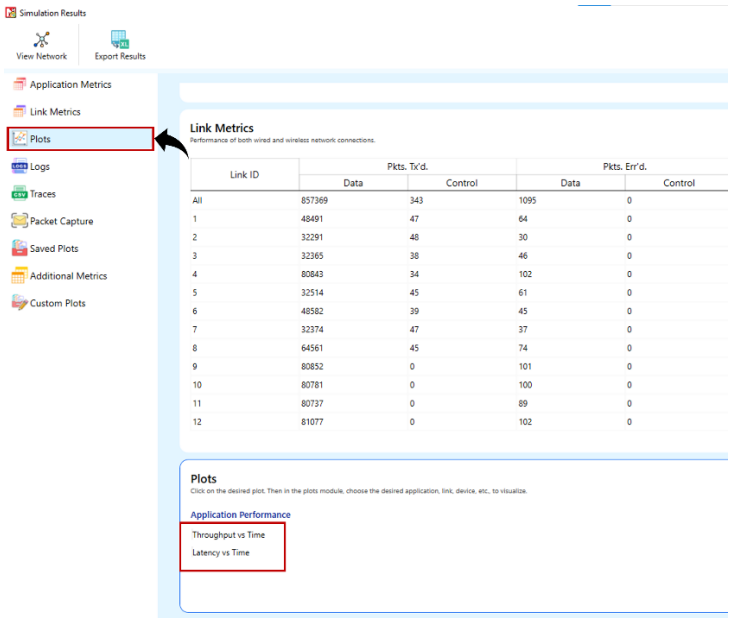



**Figure 4-61:** *Accessing Throughput vs Time and Latency vs Time plots.*

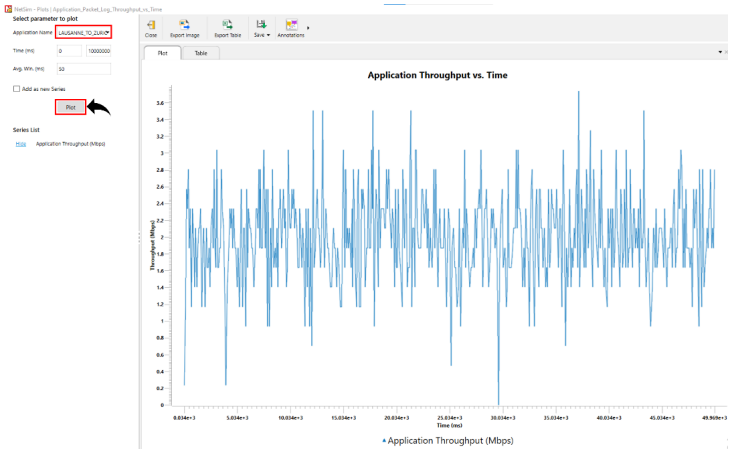



**Figure 4-62:** *Plotting Throughput vs Time.*

### Throughput plots (Lausanne to Zurich & Zurich to Lausanne):

The following plots shows the throughput vs time for:

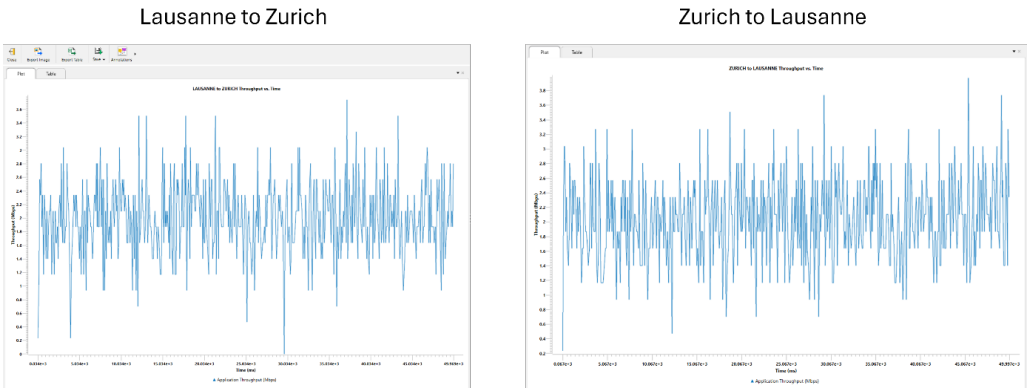



**Figure 4-63:** *NetSim plot of throughput vs. time for traffic flowing from Lausanne to Zurich and Zurich to Lausanne.*

The spread occurs because we are generating a variable bit rate, not a constant one, with a mean of 1.9 Mbps. This is achieved using an exponential random variable for inter-arrival time to simulate real-world traffic flow.

### Latency plots (Lausanne to Zurich & Zurich to Lausanne):

The following plots show the Latency vs. Time for traffic between Lausanne and Zurich Servers.

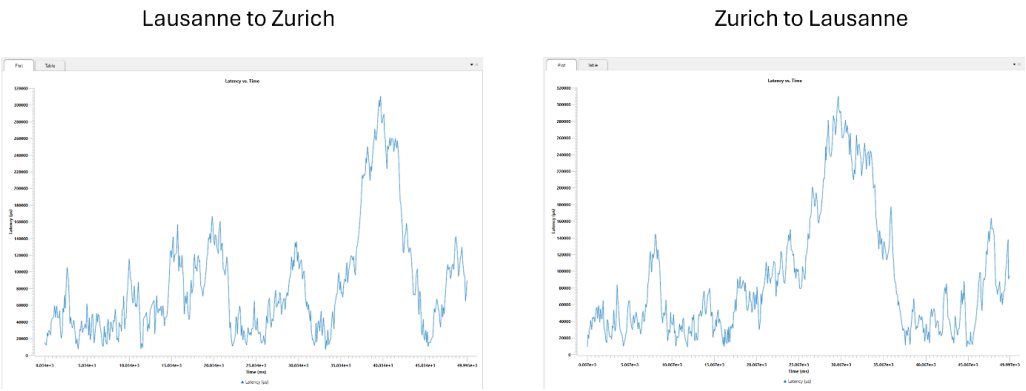



**Figure 4-64:** *NetSim plot of Latency vs. time for traffic flowing from Lausanne to Zurich and Zurich to Lausanne.*

We see that the average latency is not continuously increasing, but only varies around a mean. From this, we can infer that the network is able to fully handle the traffic flows between these two end points.

**OSPF convergence time:** We define the time at which the first data packet is forwarded by the Router as the convergence time. From the packet trace, we see that in this example, OSPF tables have converged at time ∼1217 $\mu$s or 1.2 ms (highlighted in orange). The 3 entries seen in the packet trace at 1217 $\mu$s are transmissions starting at 3 routers after table convergence.

| | A | B | C | D | E | F | G | H | I | J | K |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | PACKET_ID | SEGMENT_ID | PACKET_TYPE | CONTROL_PACKET_TYPE/APP_NAME | SOURCE_ID | DESTINATION_ID | TRANSMITTER_ID | RECEIVER_ID | APP_LAYER_ARRIVAL_TIME(US) | TRX_LAYER_ARRIVAL_TIME(US) | NW_LAYER_ARRIVAL_TIME(US) |
| 2 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-1 | Broadcast-0 | ROUTER-1 | ROUTER-2 | 0 | 0 | 0 |
| 3 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-1 | Broadcast-0 | ROUTER-1 | ROUTER-3 | 0 | 0 | 0 |
| 4 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-1 | Broadcast-0 | ROUTER-1 | ROUTER-4 | 0 | 0 | 0 |
| 5 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-2 | Broadcast-0 | ROUTER-2 | ROUTER-1 | 0 | 0 | 0 |
| 6 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-2 | Broadcast-0 | ROUTER-2 | ROUTER-3 | 0 | 0 | 0 |
| 7 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-3 | Broadcast-0 | ROUTER-3 | ROUTER-2 | 0 | 0 | 0 |
| 8 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-3 | Broadcast-0 | ROUTER-3 | ROUTER-4 | 0 | 0 | 0 |
| 9 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-3 | Broadcast-0 | ROUTER-3 | ROUTER-6 | 0 | 0 | 0 |
| 10 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-3 | Broadcast-0 | ROUTER-3 | ROUTER-1 | 0 | 0 | 0 |
| 11 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-4 | Broadcast-0 | ROUTER-4 | ROUTER-3 | 0 | 0 | 0 |
| 12 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-4 | Broadcast-0 | ROUTER-4 | ROUTER-5 | 0 | 0 | 0 |
| 13 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-4 | Broadcast-0 | ROUTER-4 | ROUTER-1 | 0 | 0 | 0 |
| 14 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-5 | Broadcast-0 | ROUTER-5 | ROUTER-4 | 0 | 0 | 0 |
| 15 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-5 | Broadcast-0 | ROUTER-5 | ROUTER-6 | 0 | 0 | 0 |
| 16 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-6 | Broadcast-0 | ROUTER-6 | ROUTER-5 | 0 | 0 | 0 |
| 17 | 0 | 0 | Control_Packet | OSPF_HELLO | ROUTER-6 | Broadcast-0 | ROUTER-6 | ROUTER-3 | 0 | 0 | 0 |
| 18 | 1 | 0 | Custom | LAUSANNE_TO_FRIBOURG | NODE-7 | NODE-8 | NODE-7 | ROUTER-1 | 0 | 0 | 0 |
| 19 | 1 | 0 | Custom | FRIBOURG_TO_LAUSANNE | NODE-8 | NODE-7 | NODE-8 | ROUTER-2 | 0 | 0 | 0 |
| 20 | 1 | 0 | Custom | BERN_TO_LAUSANNE | NODE-9 | NODE-7 | NODE-9 | ROUTER-3 | 0 | 0 | 0 |
| 21 | 1 | 0 | Custom | BASEL_TO_LAUSANNE | NODE-10 | NODE-7 | NODE-10 | ROUTER-4 | 0 | 0 | 0 |
| 22 | 1 | 0 | Custom | ZURICH_TO_LAUSANNE | NODE-11 | NODE-7 | NODE-11 | ROUTER-5 | 0 | 0 | 0 |
| 23 | 1 | 0 | Custom | LUZERN_TO_LAUSANNE | NODE-12 | NODE-7 | NODE-12 | ROUTER-6 | 0 | 0 | 0 |
| 24 | 1 | 0 | Custom | LAUSANNE_TO_FRIBOURG | NODE-7 | NODE-8 | ROUTER-1 | ROUTER-2 | 0 | 0 | 1217.16 |
| 25 | 1 | 0 | Custom | FRIBOURG_TO_LAUSANNE | NODE-8 | NODE-7 | ROUTER-2 | ROUTER-1 | 0 | 0 | 1217.16 |
| 26 | 1 | 0 | Custom | BASEL_TO_LAUSANNE | NODE-10 | NODE-7 | ROUTER-4 | ROUTER-1 | 0 | 0 | 1217.16 |
| 27 | 1 | 0 | Custom | LAUSANNE_TO_BERN | NODE-7 | NODE-9 | NODE-7 | ROUTER-1 | 0 | 0 | 0 |
| 28 | 1 | 0 | Custom | FRIBOURG_TO_BERN | NODE-8 | NODE-9 | NODE-8 | ROUTER-2 | 0 | 0 | 0 |
| 29 | 1 | 0 | Custom | BERN_TO_FRIBOURG | NODE-9 | NODE-8 | NODE-9 | ROUTER-3 | 0 | 0 | 0 |
| 30 | 1 | 0 | Custom | BASEL_TO_FRIBOURG | NODE-10 | NODE-8 | NODE-10 | ROUTER-4 | 0 | 0 | 0 |
| 31 | 1 | 0 | Custom | ZURICH_TO_FRIBOURG | NODE-11 | NODE-8 | NODE-11 | ROUTER-5 | 0 | 0 | 0 |
| 32 | 1 | 0 | Custom | LUZERN_TO_FRIBOURG | NODE-12 | NODE-8 | NODE-12 | ROUTER-6 | 0 | 0 | 0 |
| 33 | 1 | 0 | Custom | FRIBOURG_TO_BERN | NODE-8 | NODE-9 | ROUTER-2 | ROUTER-3 | 0 | 0 | 2429.32 |
| 34 | 1 | 0 | Custom | BASEL_TO_FRIBOURG | NODE-10 | NODE-8 | ROUTER-4 | ROUTER-1 | 0 | 0 | 2429.32 |

**1217 µs**

**Figure 4-65:** *OSPF Convergence time shown in packet trace.*

### Case 2: Increased Traffic Load

We observe application throughput is less than the generation rate of 1.95. However, we cannot immediately say if the network can or cannot handle this traffic load of 1.95 Mbps. To correctly estimate we need to observe how latency varies with time.

**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. Recd. |
|---|---|---|---|---|---|---|---|---|---|
| 1 | LAUSANNE_TO_FRIBOUI | 7 | 8 | 1.950000 | 1.928835 | 340706.403174 | 2593.636668 | 8427 | 8257 |
| 2 | FRIBOURG_TO_LAUSANI | 8 | 7 | 1.950000 | 1.899168 | 349160.920819 | 2558.792980 | 8256 | 8130 |
| 3 | LAUSANNE_TO_BERN | 7 | 9 | 1.950000 | 1.938413 | 364753.235476 | 2588.229458 | 8480 | 8298 |
| 4 | BERN_TO_LAUSANNE | 9 | 7 | 1.950000 | 1.867632 | 216374.491687 | 2654.261096 | 8088 | 7995 |
| 5 | LAUSANNE_TO_BASEL | 7 | 10 | 1.950000 | 1.919958 | 344770.642368 | 2644.009411 | 8378 | 8219 |
| 6 | BASEL_TO_LAUSANNE | 10 | 7 | 1.950000 | 1.955232 | 446133.864036 | 2670.465189 | 8498 | 8370 |
| 7 | LAUSANNE_TO_ZURICH | 7 | 11 | 1.950000 | 1.869968 | 374441.095785 | 2924.887060 | 8193 | 8005 |
| 8 | ZURICH_TO_LAUSANNE | 11 | 7 | 1.950000 | 1.875574 | 323476.864960 | 2865.428513 | 8152 | 8029 |
| 9 | LAUSANNE_TO_LUZERN | 7 | 12 | 1.950000 | 1.907811 | 362199.597683 | 2645.744695 | 8357 | 8167 |
| 10 | LUZERN_TO_LAUSANNE | 12 | 7 | 1.950000 | 1.928368 | 372448.931607 | 2607.698633 | 8388 | 8255 |
| 11 | FRIBOURG_TO_BERN | 8 | 9 | 1.950000 | 1.904541 | 369742.024433 | 2611.004980 | 8300 | 8153 |
| 12 | BERN_TO_FRIBOURG | 9 | 8 | 1.950000 | 1.918090 | 214890.810020 | 2572.981104 | 8304 | 8211 |
| 13 | FRIBOURG_TO_BASEL | 8 | 10 | 1.950000 | 1.900570 | 352132.898524 | 2656.929360 | 8253 | 8136 |
| 14 | BASEL_TO_FRIBOURG | 10 | 8 | 1.950000 | 1.874874 | 454635.930373 | 2780.243058 | 8182 | 8026 |

**Figure 4-66:** *Results obtained for Case 2 in Application Metrics.*

**Figure 4-67:** *NetSim plot of throughput vs. time for traffic flowing from Lausanne to Zurich and Zurich to Lausanne.*

**Figure 4-68:** *NetSim plot of Latency vs. time for traffic flowing from Lausanne to Zurich and Zurich to Lausanne.*

We see latency increasing with time. This is a sign of queuing and an indication that the network is unable to handle the traffic load of 1.95 Mbps. Saturation occurs somewhere between 1.90 to 1.95 Mbps.

### Case 3: Link Failure Analysis

Next, we study the impact of link failure. To do so, we fail the Basel–Zurich link at $t = 10$s by setting downtime to 10s in link 4 properties.

### Throughput plots: (Lausanne to Zurich & Zurich to Lausanne)

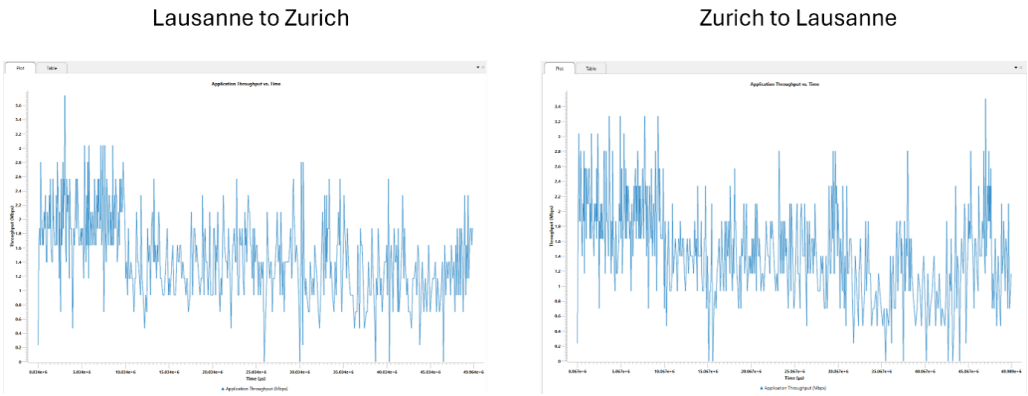



**Figure 4-69:** *NetSim plot of throughput vs. time for traffic flowing from Lausanne to Zurich and Zurich to Lausanne.*

### Latency plots: (Lausanne to Zurich & Zurich to Lausanne)

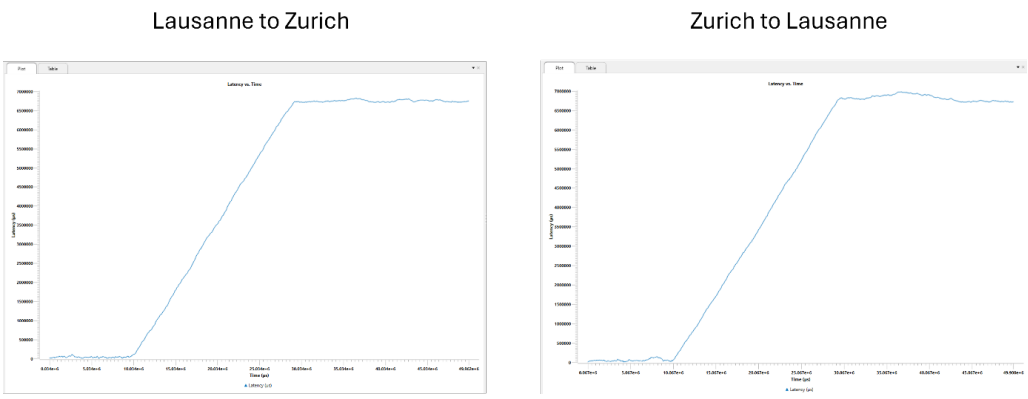



**Figure 4-70:** *NetSim plot of Latency vs. time for traffic flowing from Lausanne to Zurich and Zurich to Lausanne.*

Latency starts spiking up at the 10th second when the Basel–Zurich link fails.

- Pre-failure average delay: ∼51 ms
- Post-failure average delay: 5025 ms (5 seconds)

### Basel–Zurich Link:

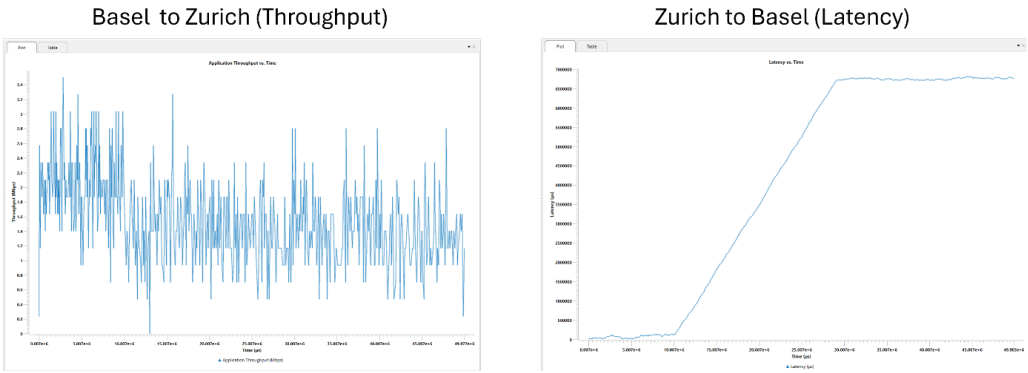



**Figure 4-71:** *NetSim plot of Throughput & Latency vs. time for Basel to Zurich.*

After the direct Basel-Zurich link fails:

- Pre-failure average delay: ∼61 ms
- Post-failure average delay: ∼2373 ms (2.4 seconds)

The new route taken by the packets, i.e., Basel ▷ Bern ▷ Luzern ▷ Zurich and can be observed from the packet trace.

**OSPF convergence time:**

- Initially, routes converge at time at ∼1217 $\mu$s or 1.22 ms.
- Before link failure, data flows using link 4 are Basel-Luzern, Basel-Zurich, Bern-Zurich, Fribourg-Zurich and Lausanne-Zurich.
- Transmissions in Link 4 get disrupted at 10 sec.
- It then takes approximately an average of 30 ms for OSPF to converge.
- During link failure, packet buffer at the router since packets can no longer be sent over the failed link.
- Once OSPF converges packets queued in the router buffer, along with newly arriving packets, start getting transmitted in the new route.
- For Bern-Zurich traffic, the new route data transmissions start at 10.030 s.

A171423 | 1626

| | PACKET_ID | SEGMENT_ID | PACKET_TYPE | CONTROL_PACKET_TYPE/APP_NAME | SOURCE_ID | DESTINATION_ID | TRANSMITTER_ID | RECEIVER_ID | PHY_LAYER_START_TIME(µS) |
|---|---|---|---|---|---|---|---|---|---|
| 170962 | 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-5 | Broadcast-0 | ROUTER-5 | ROUTER-6 | 10024722.67 |
| 170966 | 1622 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | NODE-9 | ROUTER-3 | 10025110.93 |
| 170970 | 1621 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-3 | ROUTER-6 | 10025390.32 |
| 170986 | 1622 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-3 | ROUTER-6 | 10026580.72 |
| 170987 | 1621 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-6 | ROUTER-5 | 10026585.72 |
| 170994 | 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-6 | Broadcast-0 | ROUTER-6 | ROUTER-5 | 10026806.52 |
| 171015 | 1622 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-6 | ROUTER-5 | 10027996.92 |
| 171024 | 1623 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | NODE-9 | ROUTER-3 | 10028747.41 |
| 171036 | 1621 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-5 | NODE-11 | 10029471.81 |
| 171043 | 1624 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | NODE-9 | ROUTER-3 | 10029959.57 |
| 171046 | 1623 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-3 | ROUTER-6 | 10030235.33 |
| 171055 | 1622 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-5 | NODE-11 | 10030683.97 |
| 171063 | 1624 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-3 | ROUTER-6 | 10031425.73 |
| 171064 | 1623 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-6 | ROUTER-5 | 10031430.73 |
| 171082 | 1624 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-6 | ROUTER-5 | 10032621.13 |
| 171093 | 1623 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-5 | NODE-11 | 10033108.29 |
| 171113 | 1624 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-5 | NODE-11 | 10034320.45 |
| 171200 | 1625 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | NODE-9 | ROUTER-3 | 10039656.85 |
| 171215 | 1625 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-3 | ROUTER-6 | 10040852.25 |
| 171235 | 1625 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-6 | ROUTER-5 | 10042047.65 |
| 171258 | 1625 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-5 | NODE-11 | 10043263.85 |
| 171284 | 1626 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | NODE-9 | ROUTER-3 | 10044505.49 |
| 171295 | 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-2 | Broadcast-0 | ROUTER-2 | ROUTER-1 | 10044896.57 |
| 171328 | 1638 | 0 | Custom | BASEL_TO_ZURICH | NODE-10 | NODE-11 | NODE-10 | ROUTER-4 | 10047197.59 |
| 171337 | 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-1 | Broadcast-0 | ROUTER-1 | ROUTER-3 | 10047572.6 |
| 171369 | 1638 | 0 | Custom | BASEL_TO_ZURICH | NODE-10 | NODE-11 | ROUTER-4 | ROUTER-3 | 10049409.31 |
| 171372 | 1639 | 0 | Custom | BASEL_TO_ZURICH | NODE-10 | NODE-11 | NODE-10 | ROUTER-4 | 10049621.91 |
| 171378 | 0 | 0 | Control_Packet | OSPF_LSUPDATE | ROUTER-2 | Broadcast-0 | ROUTER-2 | ROUTER-3 | 10049745.21 |
| 171383 | 1626 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-3 | ROUTER-6 | 10050375.45 |
| 171404 | 1626 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-6 | ROUTER-5 | 10051570.85 |
| 171423 | 1626 | 0 | Custom | BERN_TO_ZURICH | NODE-9 | NODE-11 | ROUTER-5 | NODE-11 | 10052787.05 |

Time to transmit data post convergence 30ms

**Figure 4-72:** *OSPF Convergence time after link failure is shown in packet trace.*

# 5 Internetworks Experiments in NetSim

Apart from examples, in-built experiments are also available in NetSim. Examples help the user understand the working of features in NetSim. Experiments are designed to help the user (usually students) learn networking concepts through simulation. The experiments contain objective, theory, set-up, results, and inference. The following experiments are available in the Experiments manual (pdf file):

1. Data traffic types and network performance measures
2. Throughput and Bottleneck Server Analysis
3. Delay and Little's Law
4. Understand working of ARP, and IP Forwarding within a LAN and across a router
5. Simulate and study the spanning tree protocol
6. Introduction to TCP connection management
7. Reliable data transfer with TCP
8. Mathematical Modelling of TCP Throughput Performance
9. Study how throughput and error of a Wireless LAN network changes as the distance between the Access Point and the wireless nodes is varied
10. Wi-Fi: UDP Download Throughput
11. How many downloads can a Wi-Fi access point simultaneously handle?
12. TCP Congestion Control Algorithms
13. Multi-AP Wi-Fi Networks: Channel Allocation
14. Study the working and routing table formation of Interior routing protocols, i.e. Routing Information Protocol (RIP) and Open Shortest Path First (OSPF)
15. M/D/1 and M/G/1 Queues
16. Wi-Fi Multimedia Extension (IEEE 802.11 EDCA)
17. Understand the working of OSPF
18. Understand the events involved in NetSim DES (Discrete Event Simulator) in simulating the flow of one packet from a Wired node to a Wireless node
19. Understand the working of TCP BIC Congestion control algorithm, simulate and plot the TCP congestion window
20. Simulating Link Failure

# 6 Reference Documents

1. IEEE 802.3 standard for Ethernet
2. IEEE 802.11 standards for Wireless LAN
3. RFCs 777, 760, 792 for Internet Control Message Protocol
4. IENs 108, 128 for Internet Control Message Protocol
5. RFC 2328 for Open Shortest Path First (OSPF)

# 7 Latest FAQs

Up-to-date FAQs on NetSim's Internetworks library are available at:

- https://tetcos.freshdesk.com/support/solutions/folders/14000108665
- https://tetcos.freshdesk.com/support/solutions/folders/14000113123
- https://tetcos.freshdesk.com/support/solutions/folders/14000119396