

# Internetworks

## Contents

|          |                                                           |          |
|----------|-----------------------------------------------------------|----------|
| <b>1</b> | <b>Introduction .....</b>                                 | <b>3</b> |
| <b>2</b> | <b>Simulation GUI .....</b>                               | <b>3</b> |
| 2.1      | Create Scenario .....                                     | 4        |
| 2.2      | Set Device, Link and Application Properties .....         | 4        |
| 2.2.1    | Device Properties .....                                   | 4        |
| 2.2.2    | Link Properties.....                                      | 6        |
| 2.2.3    | Application Properties.....                               | 6        |
| 2.3      | Enable Packet Trace, Event Trace & Plots (Optional) ..... | 7        |
| 2.4      | Run Simulation.....                                       | 7        |
| <b>3</b> | <b>Model Features.....</b>                                | <b>8</b> |
| 3.1      | WLAN 802.11a/b/g/n/ac .....                               | 8        |
| 3.1.1    | WLAN Channels in NetSim:.....                             | 8        |
| 3.1.2    | WLAN PHY Rates in NetSim .....                            | 11       |
| 3.1.3    | 802.11 n implementation.....                              | 11       |
| 3.1.4    | 802.11ac MAC and PHY Layer Implementation.....            | 14       |
| 3.1.5    | MAC Aggregation in NetSim .....                           | 15       |
| 3.1.6    | Radio Tx-Rx Execution for one Transmission .....          | 17       |
| 3.1.7    | Model Limitations.....                                    | 17       |
| 3.1.8    | IEEE802.11 Metrics .....                                  | 17       |
| 3.2      | Layer 2 (L2) Ethernet Switching .....                     | 19       |
| 3.2.1    | Spanning Tree Protocol .....                              | 19       |
| 3.2.2    | Switch Port States .....                                  | 19       |
| 3.2.3    | Model Limitations.....                                    | 20       |
| 3.2.4    | GUI Configuration of Switch.....                          | 20       |
| 3.3      | Open Shortest Path First (OSPF v2) Routing Protocol ..... | 22       |
| 3.3.1    | OSPF Overview .....                                       | 22       |
| 3.3.2    | OSPF Features.....                                        | 23       |
| 3.3.3    | Excluded Features.....                                    | 24       |
| 3.3.4    | GUI Configuration of OSPF .....                           | 25       |
| 3.4      | Transmission Control Protocol (TCP) .....                 | 27       |
| 3.4.1    | TCP overview .....                                        | 27       |

|          |                                                              |           |
|----------|--------------------------------------------------------------|-----------|
| 3.4.2    | TCP Features .....                                           | 28        |
| 3.4.3    | Congestion Control Algorithms in TCP.....                    | 28        |
| 3.4.4    | Limitations of TCP .....                                     | 28        |
| 3.4.5    | GUI configuration of TCP .....                               | 28        |
| 3.4.6    | TCP Performance Metrics.....                                 | 32        |
| 3.4.7    | TCP Reference Documents .....                                | 33        |
| 3.5      | User Datagram Protocol (UDP) .....                           | 33        |
| 3.5.1    | UDP Overview .....                                           | 33        |
| 3.5.2    | GUI Configuration of UDP.....                                | 34        |
| 3.5.3    | UDP Performance Metrics .....                                | 35        |
| 3.5.4    | UDP Reference Documents.....                                 | 35        |
| 3.6      | IP Protocol .....                                            | 35        |
| 3.6.1    | IP Performance Metrics .....                                 | 35        |
| 3.7      | Links .....                                                  | 36        |
| 3.7.1    | Modeling Error in Wired Links .....                          | 36        |
| <b>4</b> | <b>Featured Examples .....</b>                               | <b>37</b> |
| 4.1      | Factors affecting WLAN PHY Rate.....                         | 37        |
| 4.2      | Traffic Generation Rate and Service Rate .....               | 43        |
| 4.3      | Packet aggregation in 802.11n.....                           | 45        |
| 4.4      | Rate Adaptation in 802.11b .....                             | 47        |
| 4.5      | 802.11n MIMO .....                                           | 49        |
| 4.6      | Effect of Bandwidth and Guard Interval in WiFi 802.11ac..... | 51        |
| 4.7      | TCP Window Scaling.....                                      | 56        |
| 4.8      | TCP Bandwidth Delay Product .....                            | 59        |
| 4.9      | IP Addressing in NetSim .....                                | 63        |
| 4.10     | Configuring Static Routing in NetSim.....                    | 63        |
| 4.11     | Different OSPF Control Packets .....                         | 68        |
| 4.12     | Enterprise Network.....                                      | 72        |
| 4.13     | NetSim Interactive Simulation.....                           | 76        |
| <b>5</b> | <b>Reference Documents .....</b>                             | <b>87</b> |
| <b>6</b> | <b>Latest FAQs.....</b>                                      | <b>87</b> |

# 1 Introduction

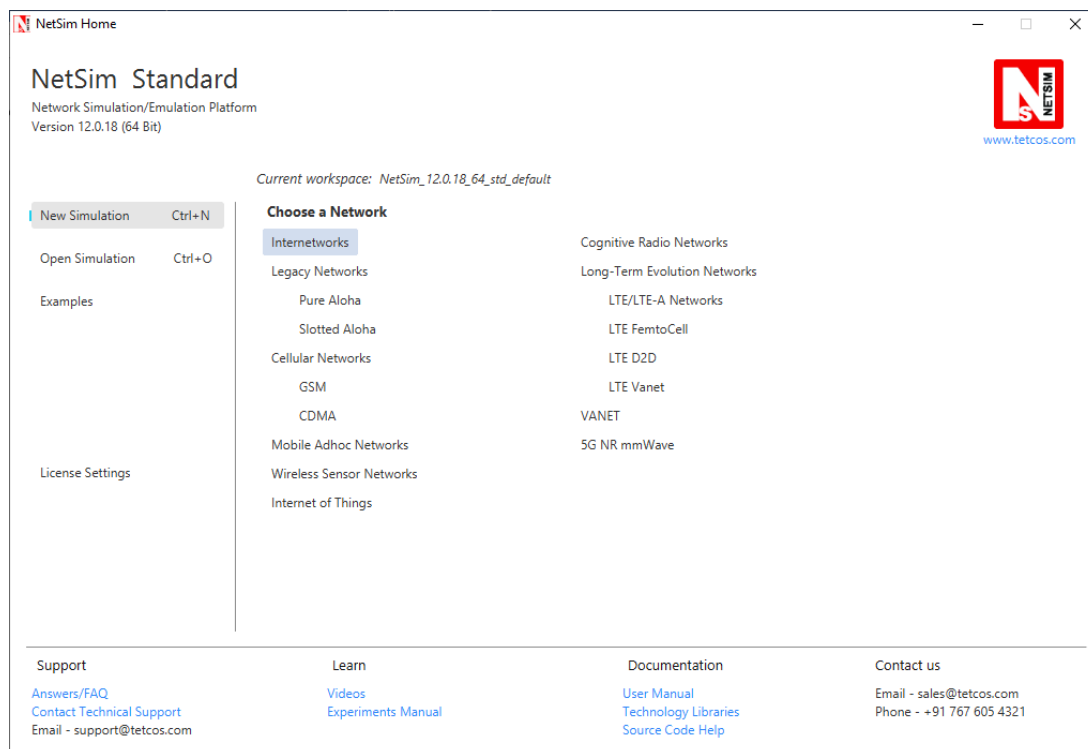
An internetwork is a collection of two or more computer networks (typically Local Area Networks or LANs) which are interconnected to form a bigger network. All networks in an Internetwork have a unique network address. Routers interconnect the networks.

Internetworks library in NetSim covers Ethernet, Address Resolution Protocol (ARP), Wireless LAN – 802.11 a / b / g / n / ac / p, 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) and Internet Group Management Protocol (IGMP).

You will be able to use the following devices when you simulate Internetworks: wireless node, wired node, switch, router, and access point (AP).

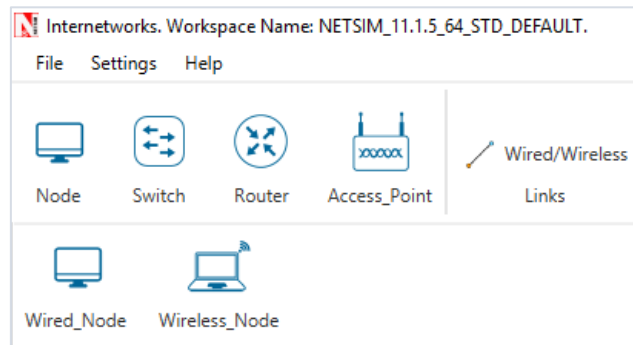
## 2 Simulation GUI

Open NetSim and click **New Simulation → Internetworks**



## 2.1 Create Scenario

Internetworks come with a palette of various devices like Switch, Router, Wired Node, Wireless Node, and AP. Select the desired device in the toolbar and click and drop on the grid.



To remove a device right click on it then click Remove.

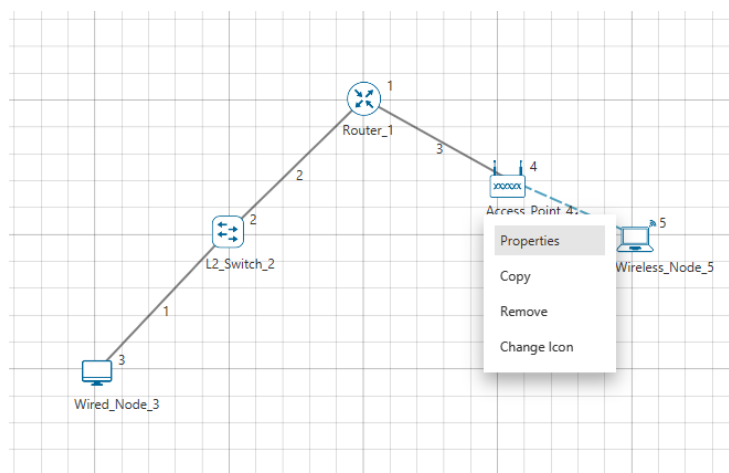
To connect two devices, select the Wired/Wireless link from the toolbar and connect the devices by clicking on device 1 followed by device 2.

A wireless link will be established when connecting Access point and Wireless Nodes.

## 2.2 Set Device, Link and Application Properties

### 2.2.1 Device Properties

To modify properties of a device, right click on it and select Properties.



Shown below are the Data Link layer and Physical Layer properties of an Access Point

Accesspoint

**Accesspoint**

GENERAL

INTERFACE\_1 (ETHERNET)

**INTERFACE\_2 (WIRELESS)**

DATA LINK\_LAYER

|                      |                |
|----------------------|----------------|
| <b>Protocol</b>      | IEEE802.11     |
| IEEE_802.11e         | Enable         |
| Rate_Adaptation      | FALSE          |
| MAC_Address          | AF1D00000402   |
| Bss Type             | INFRASTRUCTURE |
| Retry_Limit          | 4              |
| RTS_Threshold(bytes) | 3000           |
| Buffer Size(MB)      | 1              |

PHYSICAL\_LAYER

OK

Reset

Accesspoint

**Accesspoint**

GENERAL

INTERFACE\_1 (ETHERNET)

**INTERFACE\_2 (WIRELESS)**

DATA LINK\_LAYER

PHYSICAL\_LAYER

|                       |                  |
|-----------------------|------------------|
| <b>Protocol</b>       | IEEE802.11       |
| Connection_Medium     | WIRELESS         |
| Standard              | IEEE802.11b      |
| Trasmission_Type      | DSSS             |
| Trasmitter_Power(mW)  | 100              |
| Antenna Gain          | 0                |
| Antenna Height(m)     | 1                |
| SIFS( Microsec)       | 10               |
| Frequency_Band(GHz)   | 2.4              |
| Bandwidth(MHz)        | 20               |
| CCA Mode              | ENERGY_DETECTION |
| SlotTime( Microsec)   | 20               |
| Standard_Channel(MHz) | 1_2412           |
| CW_Min                | 31               |
| CW_Max                | 1023             |

OK

Reset

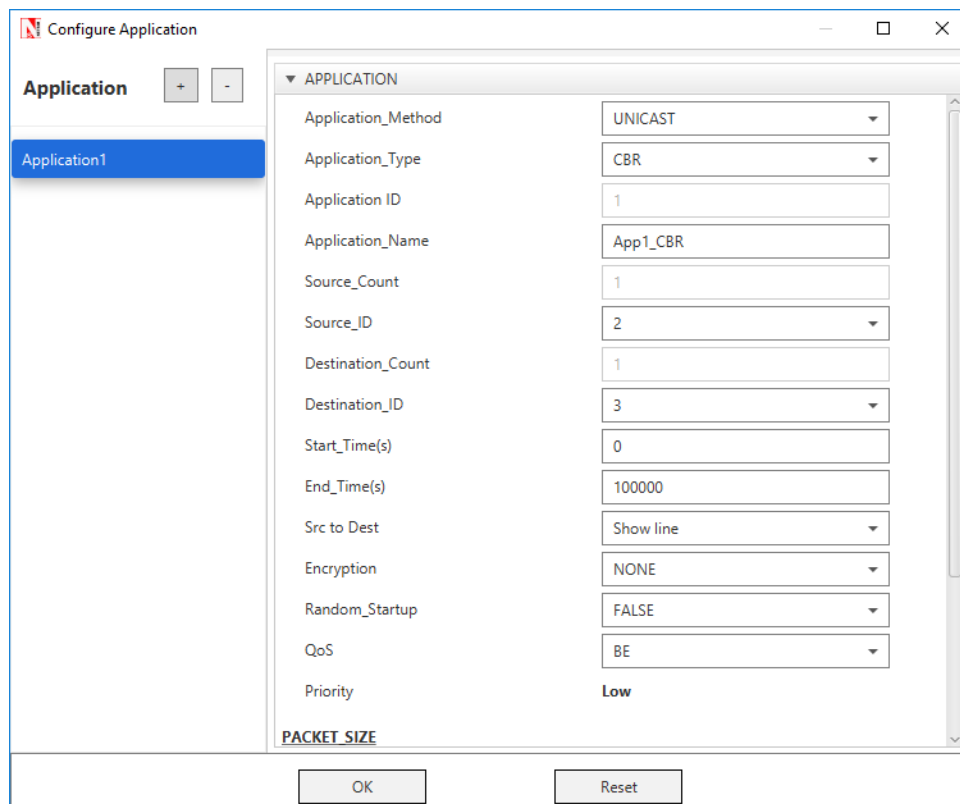
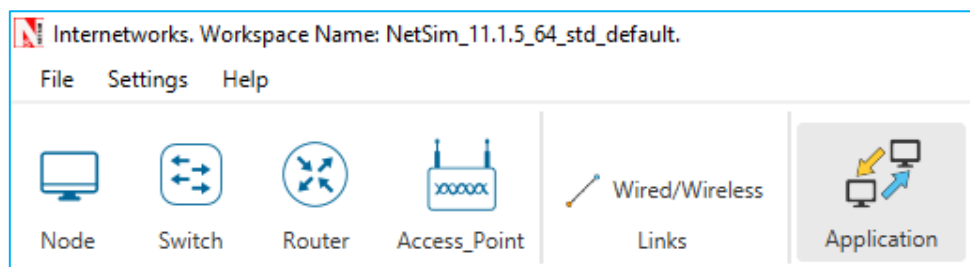
- **Global & Local Properties:** Certain properties are global in nature, i.e. changing properties in one node will automatically reflect in the others in that network. These include: Routing Protocol in Application Layer of router, user editable properties in Data Link Layer and Physical Layer etc.

## 2.2.2 Link Properties

Right click on the link and click on properties to set 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).

## 2.2.3 Application Properties

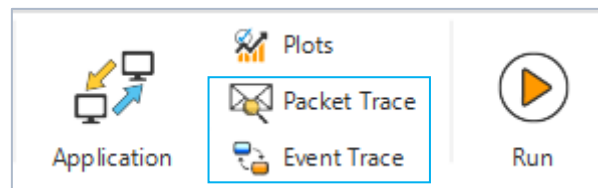
Click on the Application icon present on the ribbon and set properties.



Multiple applications can be generated by using add button in Application properties. Set the values and click OK. Detailed information on Application properties is available in section 5 of NetSim User Manual.

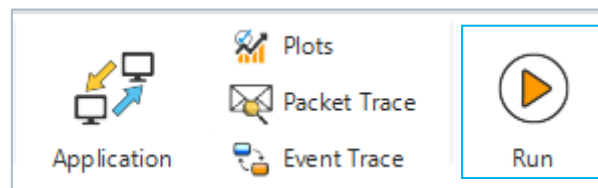
## 2.3 Enable Packet Trace, Event Trace & Plots (Optional)

Click Packet Trace / Event Trace icon in the tool bar and check Enable Packet Trace / Event Trace check box and click OK. To get detailed help, please refer sections 7.5 and 7.6 in User Manual. Select Plots icon for enabling Plots and click OK.



## 2.4 Run Simulation

Click on Run Simulation icon on the top toolbar.



Set the Simulation Time and click on **OK**.

## 3 Model Features

### 3.1 WLAN 802.11a/b/g/n/ac

#### 3.1.1 WLAN Channels in NetSim:

802.11a, 802.11b, 802.11g, 802.11n, 802.11ac, 802.11e and 802.11p are the WLAN standards available in NetSim.

| WLAN standard    | Frequency (GHz) | Bandwidth (MHz) |
|------------------|-----------------|-----------------|
| <b>802.11 a</b>  | 5               | 20              |
| <b>802.11 b</b>  | 2.4             | 20              |
| <b>802.11 g</b>  | 2.4             | 20              |
| <b>802.11 n</b>  | 2.4, 5          | 20, 40          |
| <b>802.11 ac</b> | 2.4, 5          | 20, 40, 80, 160 |

#### 3.1.1.1 RF spectrum implementations of 802.11b/g/n (2.4-GHz) and 802.11a/n/ac (5-GHz):

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

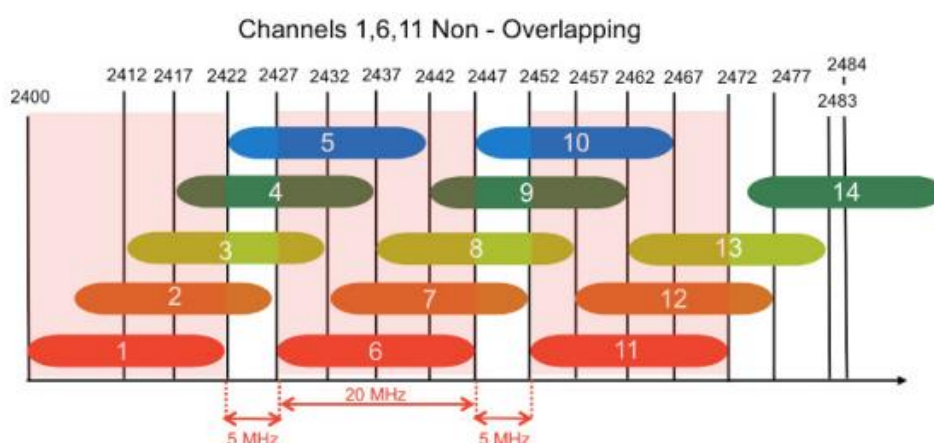
| Channel Number | Center Frequency (MHz) |
|----------------|------------------------|
| <b>1</b>       | 2412                   |
| <b>2</b>       | 2417                   |
| <b>3</b>       | 2422                   |
| <b>4</b>       | 2427                   |
| <b>5</b>       | 2432                   |
| <b>6</b>       | 2437                   |
| <b>7</b>       | 2442                   |
| <b>8</b>       | 2447                   |
| <b>9</b>       | 2452                   |
| <b>10</b>      | 2457                   |
| <b>11</b>      | 2462                   |

Channel 1, when IEEE 802.11b is configured, corresponds to a channel width of 22MHz and a center frequency of 2412MHz.

The 2.4-GHz band regulations of 802.11b/g/n have been relatively constant, given the length of time they have been in operation. The FCC (U.S) allows for 11 channels, ETSI (and most other parts of the world) allow for up to 13 channels, and Japan allows up to 14 channels but requires a special license and operating modes to operate in channel 14.

### 3.1.1.2 2.4 GHz Wi-Fi Channels

Channel plans for the 2.4 GHz band identify 14 Overlapping channels, but only 3 of these are Channels 1,6,11 are highlighted below, note that all other channels overlap or share boundaries



The following channel numbers are well-defined for **5GHz** standards:

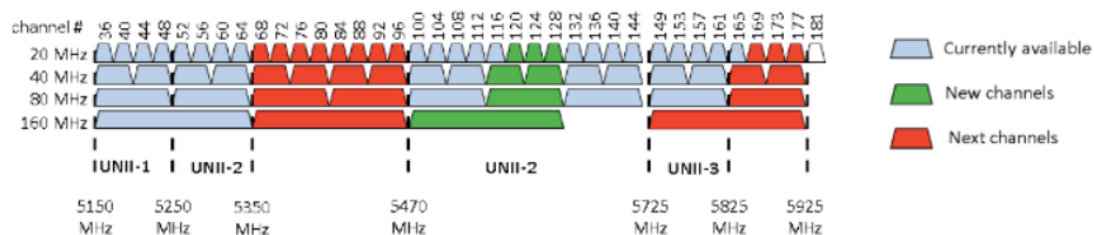
| Channel Number | Center Frequency (MHz) |
|----------------|------------------------|
| 36             | 5180                   |
| 38             | 5190                   |
| 40             | 5200                   |
| 42             | 5210                   |
| 44             | 5220                   |
| 46             | 5230                   |
| 48             | 5240                   |
| 52             | 5260                   |
| 56             | 5280                   |
| 60             | 5300                   |

|    |      |
|----|------|
| 64 | 5320 |
|----|------|

Channel 36, when IEEE 802.11n is configured at 5GHz, corresponds to a channel width of 20MHz and a center frequency of 5180MHz.

The 5.0-GHz band regulations of 802.11a/n/ac are more diverse in the channels they allow and their rules for operation. In general, with the advancement of 802.11ac most are now considering opening more spectrum for 5 GHz Wi-Fi – and all have more non overlapping channels in 5 GHz than is available anywhere in 2.4 GHz.

### 3.1.1.3 5GHz Channels



### 3.1.1.4 Channel Numbering

The standard method to denote 5 GHz channels has been to always use the 20 MHz center channel frequencies for both 20 MHz and 40 MHz wide channels.

| Channel Width | Channel Numbers                |
|---------------|--------------------------------|
| 20 MHz        | 36, 40, 44, 48, 52, 56, 60, 64 |
| 40 MHz        | 38, 46, 54, 62                 |
| 80 MHz        | 42, 58                         |
| 160 MHz       | 50                             |

The following are the channel numbers of the non-overlapping channels for 802.11ac in NetSim:

20MHz – 36, 40, 44, 48, 52, 56, 60, 64

40MHz – 36, 44, 52, 60

80MHz – 36, 52

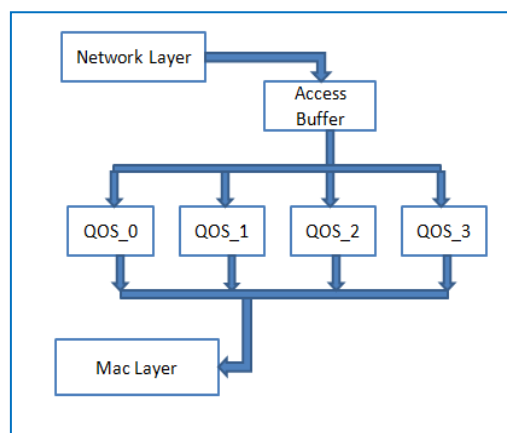
160MHz – 36

### 3.1.2 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             | 22              | 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 288.8                  |
|               |                 | 40              |              | Up to 600                    |
| ac            | 5               | 20              | 8            | Up to 346.8                  |
|               |                 | 40              |              | Up to 800                    |
|               |                 | 80              |              | Up to 1733.2                 |
|               |                 | 160             |              | Up to 3466.8                 |

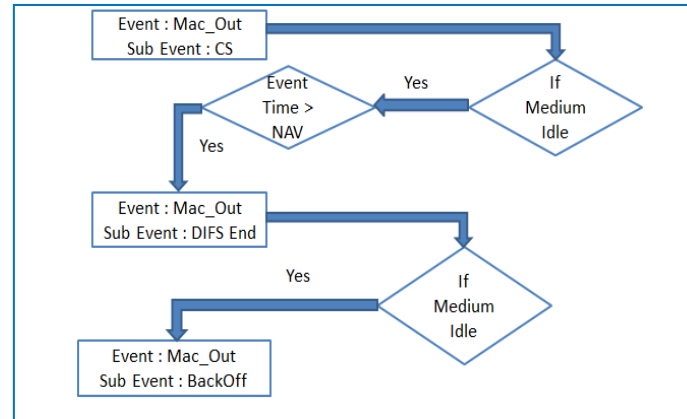
### 3.1.3 802.11 n implementation

Packets arriving from the NETWORK Layer gets queued up in an access buffer from which they are sorted according to their priority in the respective QOS buffer according to the IEEE 802.11e standard. An event MAC\_OUT with SubEvent CS (Carrier Sense – CSMA) is added to check if the medium is free



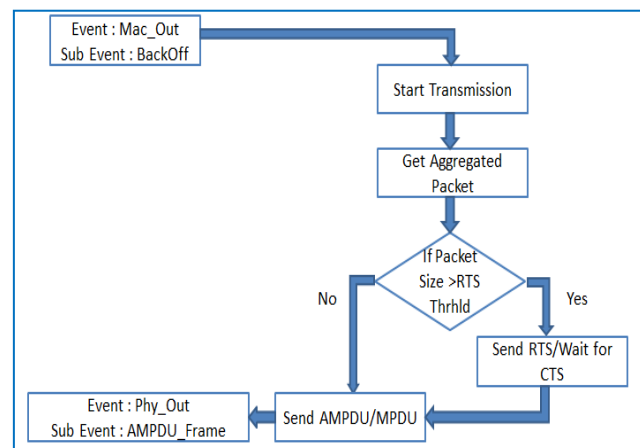
In CS, if the medium is free, then the NAV is checked. This is enabled if RTS/CTS mechanism is enabled which can be done so by adjusting the RTS Threshold. If the Present\_Time > NAV,

then an Event MAC\_OUT with SubEvent DIFS End is added at the time Present\_Time + DIFS time.



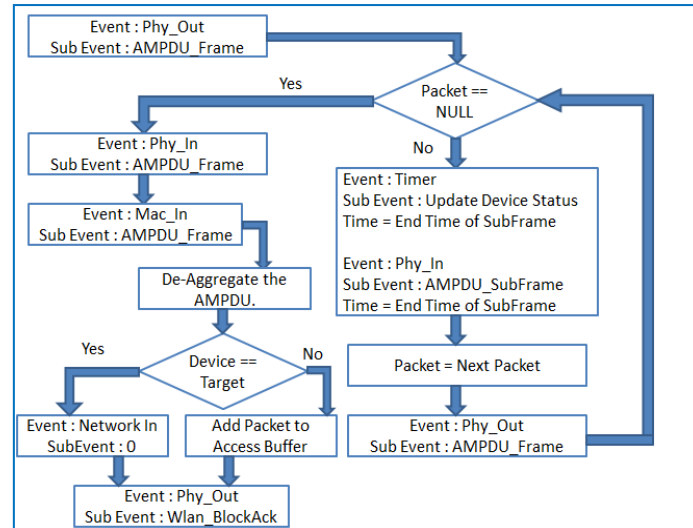
The medium is checked at the end of DIFS time period and a random time BackOff is calculated based on the Contention Window (CW). An Event MAC\_OUT with SubEvent Backoff is added at time Present\_Time + BackOff Time.

Once Backoff is successful, NetSim starts the transmission process wherein it gets the aggregated packet from the QOS buffer and stores it in the Retransmit buffer. If the A-MPDU size is > RTS Threshold, then it enables RTS/CTS mechanism which is an optional feature.



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 again as idle.



Events PHY\_OUT SubEvent AMPDU\_SubFrame, Timer Event SubEvent Update Device Status and Event PHY\_IN SubEvent AMPDU\_SubFrame are added in succession for each MPDU (Subframe of the aggregated packet). This is done for collision calculations. If two stations start transmission simultaneously, then some of the SubFrames may collide. Only those collided SubFrames will be retransmitted again. The same logic is followed for an Errored packet. 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 packet 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 packets are added to the access buffer of the receiver in addition to the packets which arrive from 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 till all packets are transmitted successfully or a retransmit limit is reached after which next set of packets are aggregated to be sent.

### 3.1.4 802.11ac MAC and PHY Layer Implementation

Improvements in 802.11ac compared to 802.11n

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

MAC layer improvements include only the increment of number of aggregated packets from 1 to 64. The MCS index for different modulation and coding rates are as follows:

| 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       |

Receiver sensitivity for different modulation schemes in 802.11ac (for a 20MHz Channel bandwidth) are as follows:

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

Number of subcarriers for different channel bandwidths

| PHY Standard                   | Subcarriers                      | Capacity relative to 20MHz in 802.11ac |
|--------------------------------|----------------------------------|----------------------------------------|
| <b>802.11n/802.11ac 20MHz</b>  | Total 56, 52 Usable (4 pilot)    | x1.0                                   |
| <b>802.11n/802.11ac 40MHz</b>  | Total 114, 108 Usable (6 pilot)  | x2.1                                   |
| <b>802.11n/802.11ac 80MHz</b>  | Total 242, 234 Usable (8 pilot)  | x4.5                                   |
| <b>802.11n/802.11ac 160MHz</b> | Total 484, 468 Usable (16 pilot) | x9.0                                   |

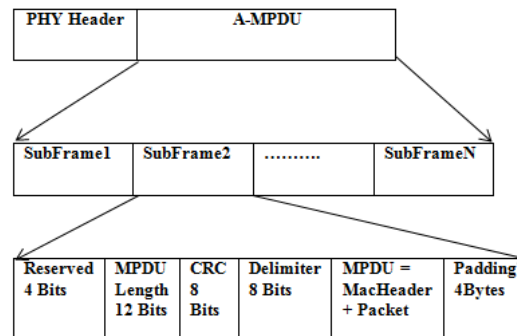
Now with the knowledge of MCS index and bandwidth of the channel data rate is set in the following manner

- 1 Get the number subcarriers that are usable for the given bandwidth of the medium.
- 2 Get the Number of Bits per Sub Carrier (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 (4micro sec for long GI and 3.6 micro sec for short GI)

### 3.1.5 MAC Aggregation in NetSim

Mac aggregation and block acknowledgement are two important enhancements to 802.11n standard. In the aggregation scheme, several MPDU's (MAC Protocol Data Units) are aggregated in to a single A-MPDU (Aggregated MPDU). The A-MPDU's are created before sending to PHY layer for transmission. The MAC does not wait for MPDU's before

aggregation. It aggregates the already present packets in the queue to form an A-MPDU. The maximum size of A-MPDU is 65535 bytes. The maximum size of each MPDU is 4KB. In A-MPDU, each MPDU has a delimiter of 32bits at the beginning and padding at the end. These padding bytes ensure that size of MPDU is a multiple of 4bytes.



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 64bits or 8 bytes. This bitmap can acknowledge up to 64 packets, 1bit for each packet.

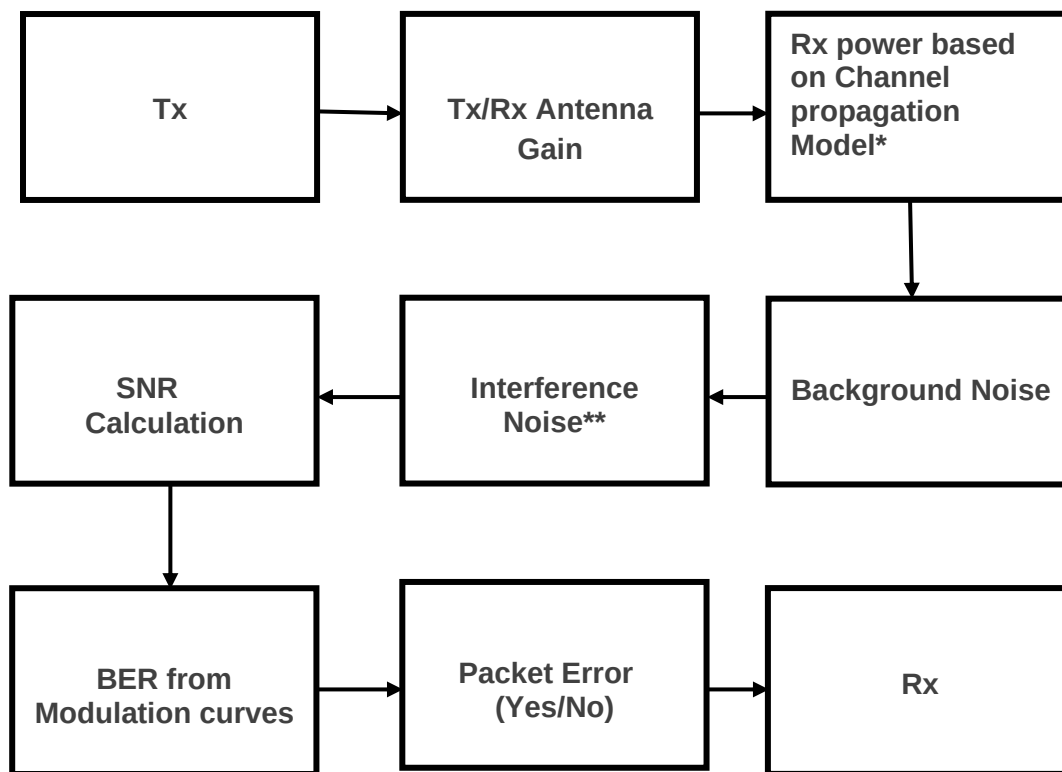
The value of a bitmap field is 1, if respective packet is received without error else it is 0. Only the error packets are resent until a retry limit is reached. The number of packets in an A-MPDU is restricted to 64 since the size of block ack bitmap is 64bits.

|               |              |    |    |            |                              |        |     |
|---------------|--------------|----|----|------------|------------------------------|--------|-----|
| Octets:2      | 2            | 6  | 6  | 2          | 2                            | 8      | 4   |
| Frame Control | Duration /ID | RA | TA | BARControl | BA Starting Sequence Control | BitMap | FCS |

### Block Ack Control Packet

- NetSim aggregates packets in terms of numbers and not size.
- A user can vary the number of packets to aggregate by changing the appropriate parameters in the GUI.
- NetSim ignores the padding bytes added to the MPDU since its effect is negligible.
- NetSim aggregates packets to the same receiver id and not to the destination ID.

### 3.1.6 Radio Tx-Rx Execution for one Transmission



\* Propagation model covers path loss, fading and shadowing

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

### 3.1.7 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.

### 3.1.8 IEEE802.11 Metrics

IEEE802.11 Metrics will be displayed in the results dashboard if the network scenario simulated consisted of at least one device with WLAN protocol enabled.

| Parameter    | Description                                                                 |
|--------------|-----------------------------------------------------------------------------|
| Device_Id    | It represents the Id's of the wireless devices which supports 802.11 (WLAN) |
| Interface_Id | It represents the interface Id's of the wireless nodes                      |

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Frame Sent</b>         | It is the Number of frames sent by Access Point                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Frame Received</b>     | It is the number of frames received by a wireless node                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>RTS Sent</b>           | It is the number of Request to send (RTS) packets sent by a Wireless Node.<br>RTS/CTS frames are sent prior to transmission when the packet size exceeds 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. |
| <b>RTS Received</b>       | It is the number of RTS packets received by an Access Points                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>CTS Sent</b>           | It is the number of Clear to send (CTS) packets sent by an Access Points                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>CTS Received</b>       | It is the number of CTS packets received by Wireless Nodes                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Successful Backoff</b> | 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 |
| <b>Failed Backoff</b> | It is the number of failed backoffs at wireless node                                                                      |

## 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.

Note that the switch 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 is set to a value that is greater than the time it takes for the spanning tree protocol to complete (typically of the order of a hundreds of milliseconds).

### 3.2.3 Model Limitations

1. The spanning protocol is only run at the beginning of simulation. If a link fails, the spanning 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.

### 3.2.4 GUI Configuration of Switch

OSPF properties can be set by right clicking on a switch --> Properties --> Interface\_x (Ethernet).

The screenshot shows the 'L2\_Switch' configuration window. On the left, a sidebar lists 'GENERAL', 'INTERFACE\_1 (ETHERNET)' (selected), and 'INTERFACE\_2 (ETHERNET)'. The main area is titled 'DATALINK\_LAYER' and contains the following settings:

| Property                | Value         |
|-------------------------|---------------|
| Protocol                | ETHERNET      |
| MAC_Address             | AF1D00000501  |
| Buffer Size(MB)         | 1             |
| STP Status              | TRUE          |
| Switch Priority         | 1             |
| Switch ID               | 1AF1D00000501 |
| Spanning Tree           | IEEE 802.1D   |
| STP Cost                | 90            |
| Switching Mode          | Store_Forward |
| Switching Latency(μsec) | 21.7          |
| VLAN_Status             | FALSE         |

Below the 'DATALINK\_LAYER' section is a collapsed 'PHYSICAL\_LAYER' section. At the bottom of the window are 'OK' and 'Reset' buttons.

The properties that can be set are:

| Parameter               | Type * | Range          | Description                                                                                                                                                                                                                                                                                                                                                       |
|-------------------------|--------|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>MAC ADDRESS</b>      | 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).                                                                                                                                                                       |
| <b>Buffer Size (MB)</b> | 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.                                                                                                             |
| <b>STP Status</b>       | Fixed  | TRUE           | Spanning Tree Protocol is set to “True” in the Switches by default.                                                                                                                                                                                                                                                                                               |
| <b>Switch Priority</b>  | Local  | 1-61440        | This is the priority that can be assigned to the Switch. Priority is involved in deciding the root bridge for STP.                                                                                                                                                                                                                                                |
| <b>Switch ID</b>        | Fixed  | 1-15           | 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 address of the connected ports is taken while forming the unique ID.                                                                                                 |
| <b>Spanning Tree</b>    | 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.                                                                                                                                                                                                                                                                   |
| <b>STP Cost</b>          | Local | 0-1000                     | Cost used by the switch to calculate spanning tree. The cost assigned to each port is based on its data rate.                                                                                                                                                                                                                                                                                                     |
| <b>Switching Mode</b>    | Local | Store Forward, Cut Through | <p>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.</p> <p>Cut through: Switch forwards the incoming frames to its appropriate outgoing port immediately after receipt of destination address of the frame.</p> |
| <b>Switching Latency</b> | Local | 0-10000                    | Switching latency is defined as the time it takes for a switch to forward a packet from its ingress port to its egress port. The lower the latency, the less time the packet needs to stay in the switch waiting to be processed, the faster the switch, the quicker the packets can be sent to the intended destination host.                                                                                    |
| <b>VLAN Status*</b>      | Local | TRUE, FALSE                | To enable/disable VLAN                                                                                                                                                                                                                                                                                                                                                                                            |

\*Requires Component 3 Advanced Routing and switching license

## 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 internal 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 re-transmission 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

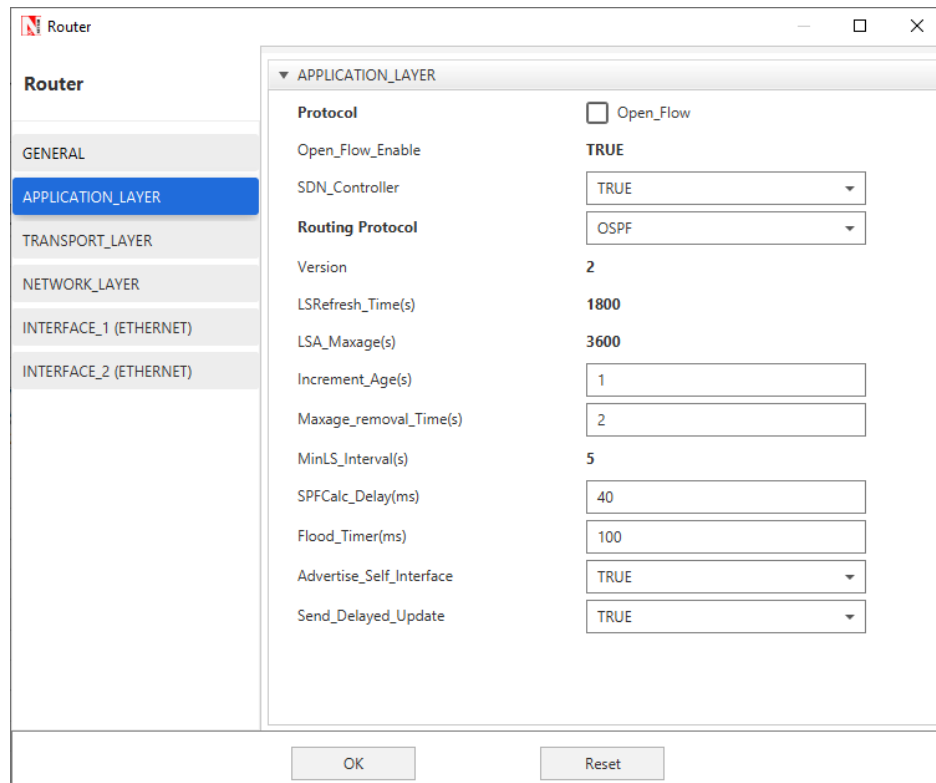
### 3.3.3 Excluded Features

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 multi-point

### 3.3.4 GUI Configuration of OSPF

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



The properties that can be set are:

| Parameter                 | Type * | Range | Description                                                                                                                                                                                                                                                                                                                              |
|---------------------------|--------|-------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Version</b>            | Global | Fixed | OSPF Version 2 as per RFC 2328 for IPv4.                                                                                                                                                                                                                                                                                                 |
| <b>LSRefresh_Time (s)</b> | 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.                                                                                                                                                                                                                                                 |
| <b>LSA_Maxage (s)</b>          | 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           |
| <b>Increment_Age (s)</b>       | 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 |
| <b>Maxage_removal_Time (s)</b> | Global | 0 - 9999 | This variable decides the time when the LSA is removed from the MaxAgeLSA List                                                                                                                                                                                                               |
| <b>MinLS_Interval (s)</b>      | Global | Fixed    | The minimum time between distinct originations of any particular LSA. The value of MinLSInterval is set to 5 seconds                                                                                                                                                                         |
| <b>SPFCalc_Delay (ms)</b>      | 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                                                                                                          |
| <b>Flood_Timer (ms)</b>        | 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 ISSENDDelayUpdate variable setting                                                                                                 |
| <b>Advertise_Self_Interface</b> | 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 |
| <b>Send_Delayed_Update</b>      | 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 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.

|             |
|-------------|
| Application |
| <b>TCP</b>  |
| IP          |
| MAC         |
| PHY         |

Fig: Protocol Layering

### 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 GUI configuration of TCP

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

**Wirednode**

GENERAL

APPLICATION\_LAYER

**TRANSPORT\_LAYER**

NETWORK\_LAYER

INTERFACE\_1 (ETHERNET)

**TRANSPORT\_LAYER**

**Protocol** ☒ TCP

Congestion\_Control\_Algorithm NEW\_RENO

Max\_SYN\_Retries 5

Acknowledgement\_Type Undelayed

MSS(bytes) 1460

Initial\_SSThreshold(bytes) 65535

Time\_Wait\_Timer(s) 120

Selective\_ACK FALSE

Window\_Scaling FALSE

Sack\_Permitted FALSE

Timestamp\_Option FALSE

**Protocol** ☒ UDP

Segment\_Size(bytes) 1472

OK Reset

The properties that can be set are:

| Parameter                           | Type * | Range                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------|--------|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Congestion Control Algorithm</b> | Local  | OLD TAHOE, TAHOE, RENO, NEW RENO, BIC, CUBIC | <p>Congestion control algorithm is used to control the network congestion.</p> <p>Old Tahoe is the combination of slow start and congestion avoidance algorithm.</p> <p>The Fast-retransmit algorithms 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.</p> |

|                             |       |                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------|-------|--------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                             |       |                    | <p>Reno implements fast recovery in case of three duplicate acknowledgements.</p> <p>New Reno improves retransmission during the fast-recovery phase of TCP Reno.</p> <p>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.</p> <p>CUBIC is an implementation of TCP with an optimized congestion control algorithm for high bandwidth networks with high latency.</p> |
| <b>Max SYN Retries</b>      | Local | 1-10               | <p>Maximum number of TCP SYN ACK packets that can be retransmitted. The value should in the range of 1 to 10.</p>                                                                                                                                                                                                                                                                                                                                      |
| <b>Acknowledgement Type</b> | Local | Delayed, Undelayed | <p>If set to delayed, ACK response will be delayed improving network performance. If set to Un delayed, ACK will be sent immediately without delay.</p>                                                                                                                                                                                                                                                                                                |
| <b>MSS(bytes)</b>           | Local | 64-1460            | <p>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.</p> <p><math>MSS = MTU - (\text{Network and Transport layer protocol headers})</math>.</p>                                                                                                                                                                                                              |

|                                    |       |             |                                                                                                                                                                                                                                                          |
|------------------------------------|-------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Initial SSThreshold(bytes )</b> | Local | 5840-65535  | The server-initial-ss-threshold should be in the range between 5840 and 65535 bytes.                                                                                                                                                                     |
| <b>Time Wait Timer(s)</b>          | 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.                                                                                       |
| <b>Selective ACK</b>               | Local | TRUE, FALSE | In Selective Acknowledgment (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.                                |
| <b>Window Scaling</b>              | 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.                                                                                            |
| <b>Sack Permitted</b>              | 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.                                                                       |
| <b>Timestamp Option</b>            | 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

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:

| Parameter                    | Description                                                                                                                                                                                                 |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Source</b>                | It displays the name with ID of the source device which generates TCP packets                                                                                                                               |
| <b>Destination</b>           | It displays the name with ID of the destination device which receives TCP packets                                                                                                                           |
| <b>Local Address</b>         | It displays the local IP address with port number of the device present in source column                                                                                                                    |
| <b>Remote Address</b>        | It represents the remote IP address with port number for the source and destination                                                                                                                         |
| <b>Syn Sent</b>              | It is the number of syn packets sent by the source                                                                                                                                                          |
| <b>Syn-Ack Sent</b>          | It is the number of syn ack packets sent by the destination                                                                                                                                                 |
| <b>Segment Sent</b>          | It is the number of segments sent by a source                                                                                                                                                               |
| <b>Segment Received</b>      | It is the number of segments received by a destination                                                                                                                                                      |
| <b>Segment Retransmitted</b> | It is the number of segments retransmitted by the source                                                                                                                                                    |
| <b>Ack Sent</b>              | 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 |
| <b>Ack Received</b>          | 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         |
| <b>Duplicate segment received</b>    | It is the number of duplicate segments received by destination     |
| <b>Out of order segment received</b> | It is the number of out of ordered packets received by destination |
| <b>Duplicate ack received</b>        | It is the number of duplicate acknowledgements received by source  |
| <b>Times RTO expired</b>             | 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 Selective Acknowledgment (SACK) for TCP
9. RFC 7323: TCP Extensions for High Performance
10. [https://research.csc.ncsu.edu/netsrv/sites/default/files/cubic\\_a\\_new\\_tcp\\_2008.pdf](https://research.csc.ncsu.edu/netsrv/sites/default/files/cubic_a_new_tcp_2008.pdf)
11. <https://research.csc.ncsu.edu/netsrv/sites/default/files/bitcp.pdf>
12. [https://research.csc.ncsu.edu/netsrv/sites/default/files/hystart\\_techreport\\_2008.pdf](https://research.csc.ncsu.edu/netsrv/sites/default/files/hystart_techreport_2008.pdf)

## 3.5 User Datagram Protocol (UDP)

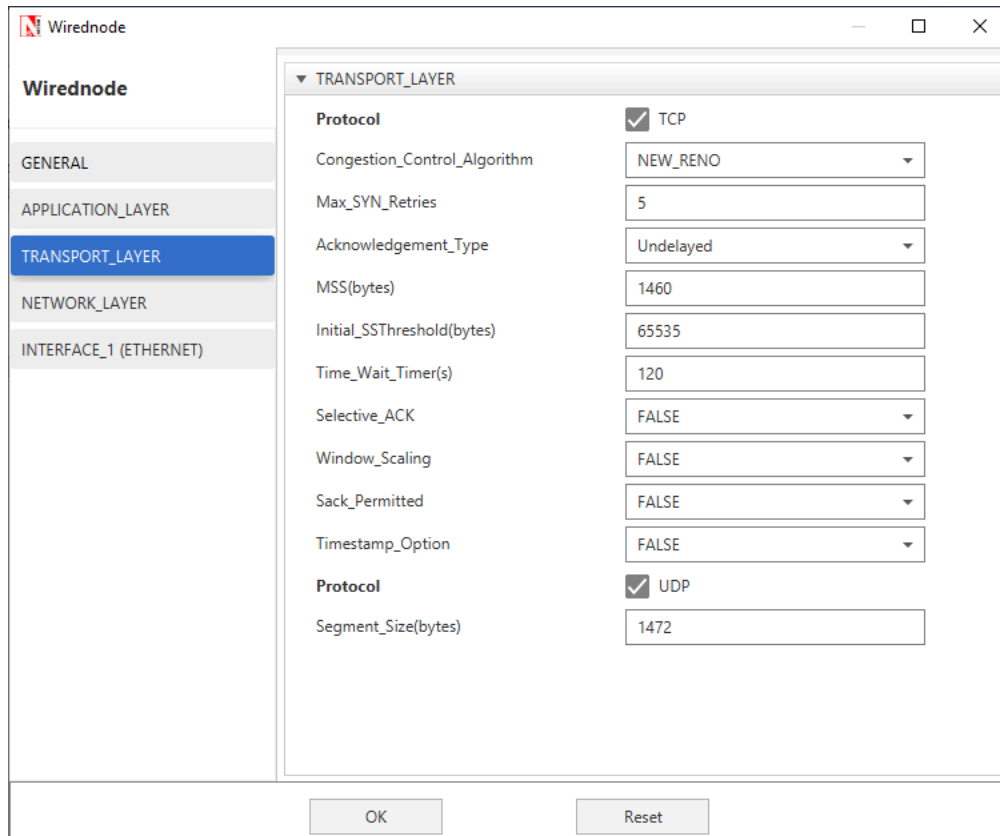
### 3.5.1 UDP Overview

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 ordered reliable delivery of streams of data should use the Transmission Control Protocol (TCP).

### 3.5.2 GUI Configuration of UDP

The UDP parameters can be accessed by right clicking on a node and selecting Properties - > Transport Layer



The properties that can be set are:

| Parameter                   | Type * | Range   | Description                                                                                                                                                                                                                             |
|-----------------------------|--------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Segment Size (bytes)</b> | Local  | 0-65535 | <p>The maximum amount of data that a single message may contain. The Segment Size is the maximum data size and does not include the size of the header.</p> <p>Segment Size = MTU – (Network and Transport layer protocol headers).</p> |

### 3.5.3 UDP Performance Metrics

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:

| 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

1. RFC 768: User Datagram Protocol

## 3.6 IP Protocol

### 3.6.1 IP Performance Metrics

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:

| Parameter | Description                                 |
|-----------|---------------------------------------------|
| Device_Id | It displays the Id's of the Layer_3 devices |

|                         |                                                                                                                                                                                  |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Packet sent</b>      | It is the number of packets sent by a source, intermediate devices (Router or L3 switch)                                                                                         |
| <b>Packet forwarded</b> | It is the number of packets forwarded by intermediate devices (Router or L3 switch)                                                                                              |
| <b>Packet received</b>  | It is the number of data packets received by destination, intermediate devices (routing packets (OSPF, RIP etc.) received by Routers)                                            |
| <b>Packet discarded</b> | It is the number of data packets that are discarded after their TTL value is expired.                                                                                            |
| <b>TTL expired</b>      | 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 |
| <b>Firewall blocked</b> | It is the number of packets blocked by firewall at routers                                                                                                                       |

## 3.7 Links

### 3.7.1 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)

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

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

$$PER = BER * L$$

## 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.

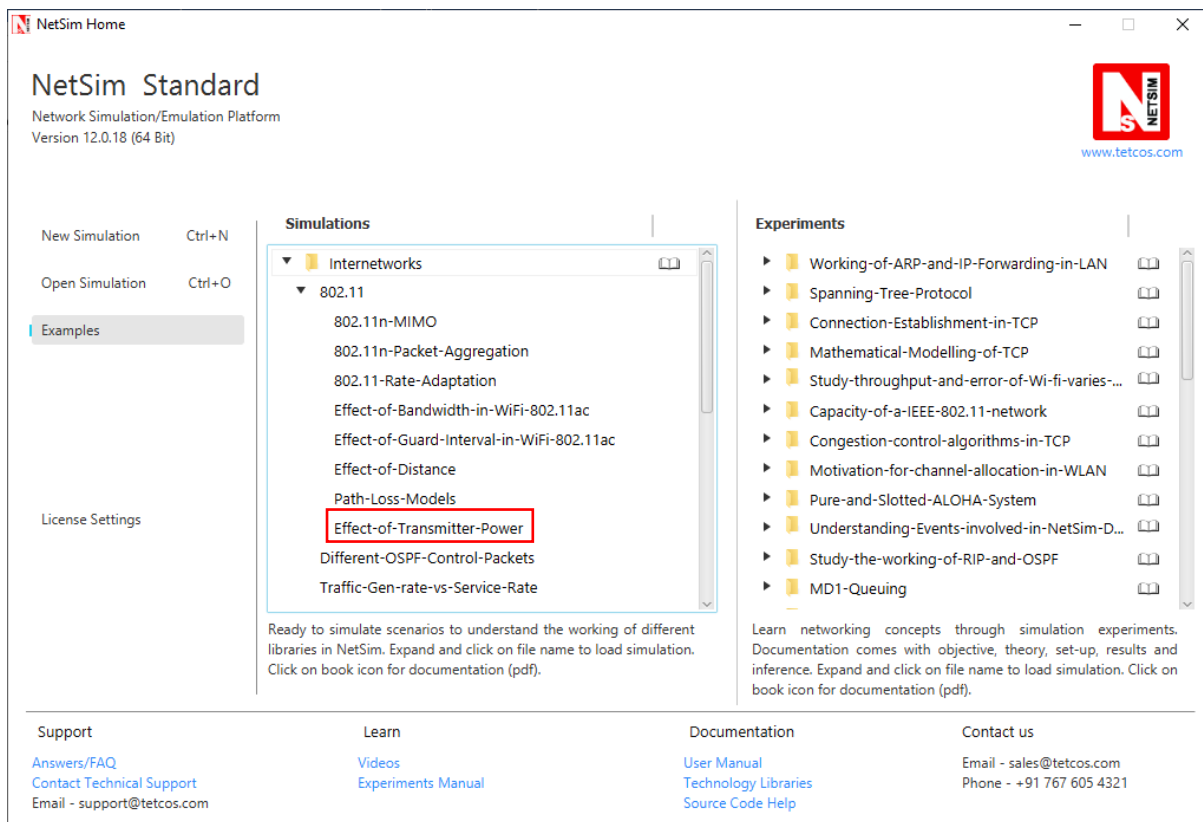
### 4.1 Factors affecting WLAN PHY Rate

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

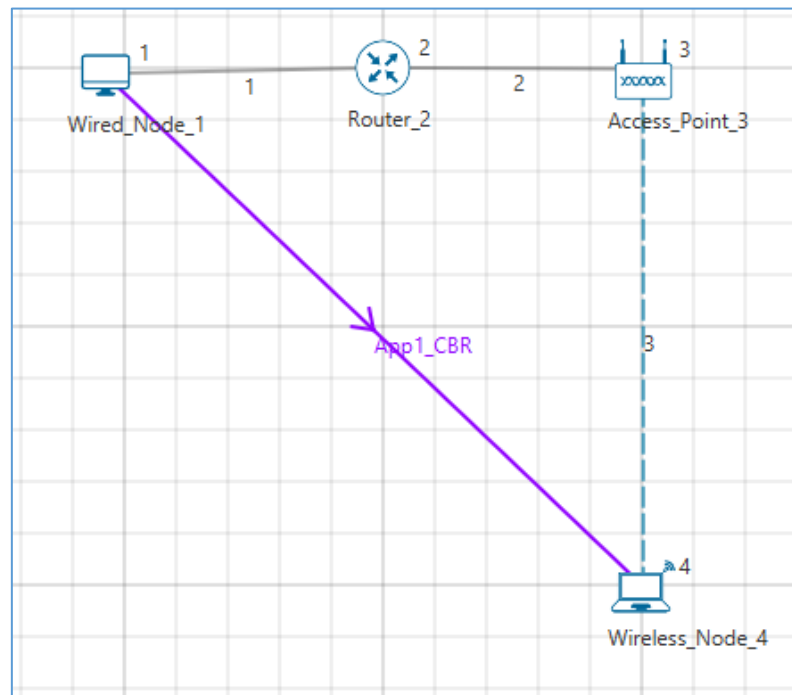
- Transmitter power (More Tx power implies higher throughput)
- Channel Path loss (Higher path loss exponent leads to lower throughput)
- Receiver sensitivity (Lower Rx sensitivity leads to higher data rate)
- Distance (Higher distance between nodes leads to lower throughput)

#### Example1: (Effect of Transmitter power)

Open NetSim and click **Examples > Internetworks > 802.11 > Effect-of-Transmitter-Power** as shown below:



The following network diagram illustrates, what the NetSim UI displays when you open the example configuration file.



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, higher throughputs and lesser error.

### Settings done in the Network:

1. Distance between Access Point and the Wireless Node is set to 210m
2. TCP is disabled in Wired Node (i.e. Source)
3. Set transmitter power to 100mW under Interface Wireless > Physical layer properties of Access point
4. Channel Characteristics: Path Loss Only, Path Loss Model: Log Distance, Path Loss Exponent: 2.5
5. Application Generation Rate: 10Mbps (Packet Size: 1460, Inter Arrival Time: 1168 $\mu$ s)
6. Run simulation for 10s
7. Similarly, reset the Transmitter Power to 80, 60, 40, and 20 respectively and see that, there is a decrease in the Throughput gradually.

## Output:

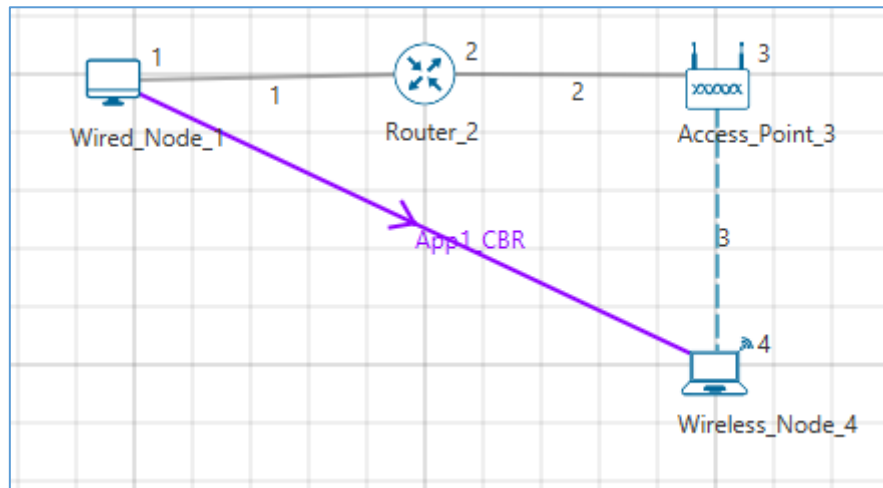
| Transmitter Power (mW) | Throughput (Mbps) |
|------------------------|-------------------|
| 100                    | 5.83              |
| 80                     | 5.17              |
| 60                     | 3.79              |
| 40                     | 3.48              |
| 20                     | 1.67              |

## Example2: (Effect-of-Distance)

Open NetSim and click **Examples > Internetworks > 802.11 > Effect-of-Distance**

The screenshot displays the NetSim Standard interface. On the left, there are buttons for 'New Simulation' (Ctrl+N), 'Open Simulation' (Ctrl+O), 'Examples' (highlighted), and 'License Settings'. The main area is divided into 'Simulations' and 'Experiments'. Under 'Simulations', the 'Internetworks' folder is expanded, showing a list of 802.11-related simulations. 'Effect-of-Distance' is highlighted with a red box. Below this list, a note states: 'Ready to simulate scenarios to understand the working of different libraries in NetSim. Expand and click on file name to load simulation. Click on book icon for documentation (pdf)'. The 'Experiments' section on the right lists various networking experiments, each with a book icon for documentation. At the bottom, there are four columns: 'Support' (with links to Answers/FAQ, Contact Technical Support, and Email), 'Learn' (with links to Videos and Experiments Manual), 'Documentation' (with links to User Manual, Technology Libraries, and Source Code Help), and 'Contact us' (with Email and Phone information).

The following network diagram illustrates, what the NetSim UI displays when you open the example configuration file.



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 the channel can handle also drops.

#### Settings done in the Network:

1. Distance between Access Point and the Wireless Node is set to 50m
2. TCP is disabled in Wired Node (i.e. Source)
3. Channel Characteristics: Path Loss Only, Path Loss Model: Log Distance, Path Loss Exponent: 3.2
4. Application Generation Rate: 2Mbps (Packet Size: 1460, Inter Arrival Time: 5840 $\mu$ s)
5. Run simulation for 10s
6. Similarly, increase the Distance from 50m to 100, 150, and 200 respectively and see that, there is a decrease in the Throughput.

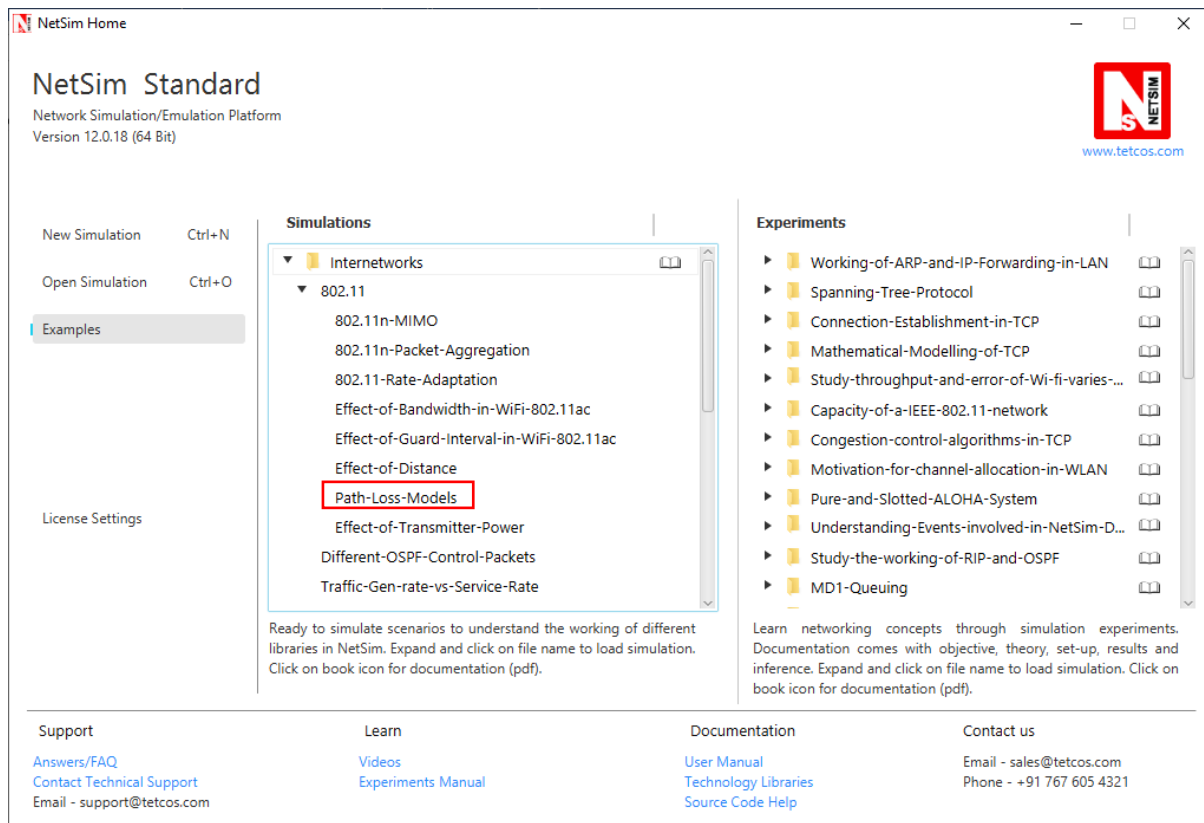
#### Output:

| Distance (m) | Throughput (Mbps) |
|--------------|-------------------|
| 50           | 1.99              |
| 100          | 1.67              |

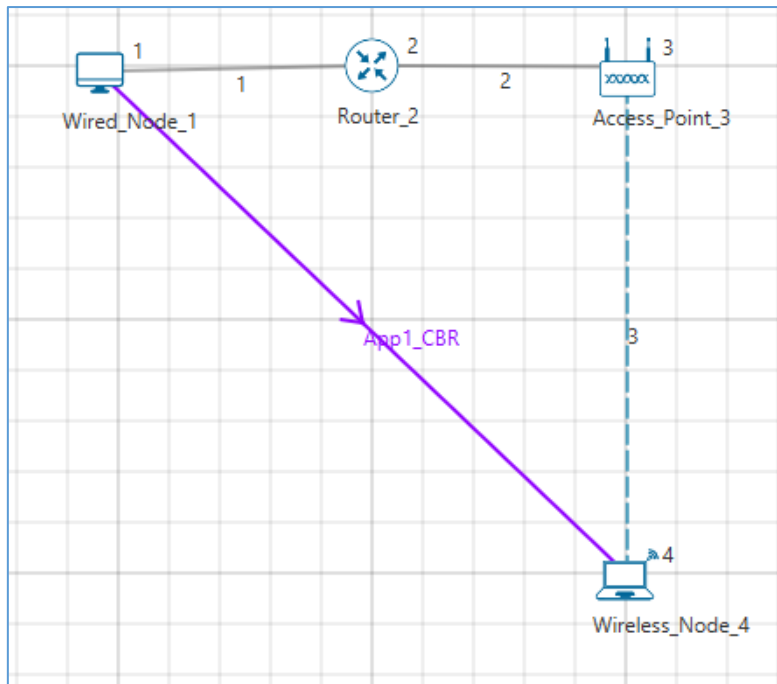
|     |      |
|-----|------|
| 150 | 1.67 |
| 200 | 1.67 |

### Example3: (Path Loss Models)

Open NetSim and click **Examples > Internetworks > 802.11 > Path-Loss-Models**



The following network diagram illustrates, what the NetSim UI displays when you open the example configuration file.



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.

#### Settings done in the Network:

1. Distance between Access Point and the Wireless Node is set to 100m
2. TCP is disabled in Wired Node (i.e. Source)
3. Channel Characteristics: Path Loss Only, Path Loss Model: Log Distance, Path Loss Exponent: 3.1
4. Application Generation rate: 2Mbps (Packet Size: 1460, Inter Arrival Time: 5840 $\mu$ s)
5. Run simulation for 10s
6. Similarly, increase the Path Loss Exponent from 3.1 to 3.2, 3.3, 3.4, and 3.5 respectively and see that, there is a decrease in the Throughput.

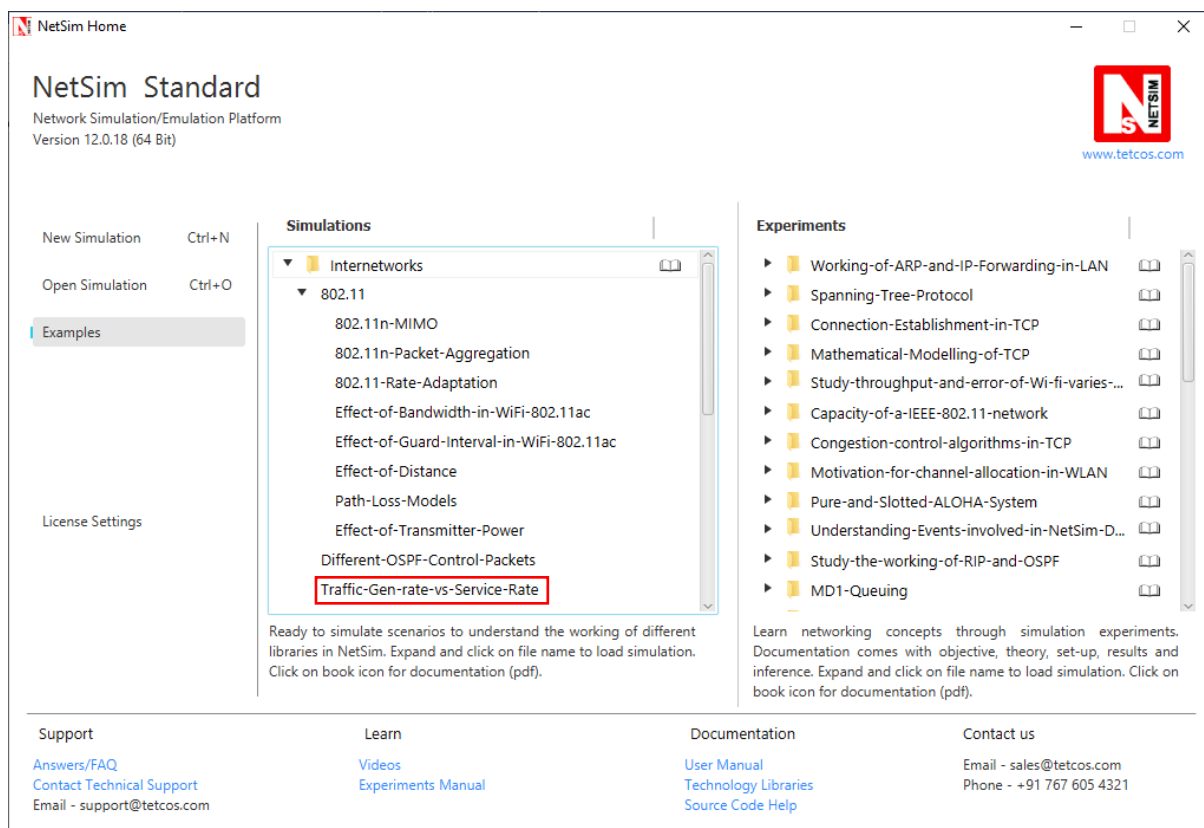
#### Output:

| Path loss Exponent | Throughput (Mbps) |
|--------------------|-------------------|
| 3.1                | 1.99              |

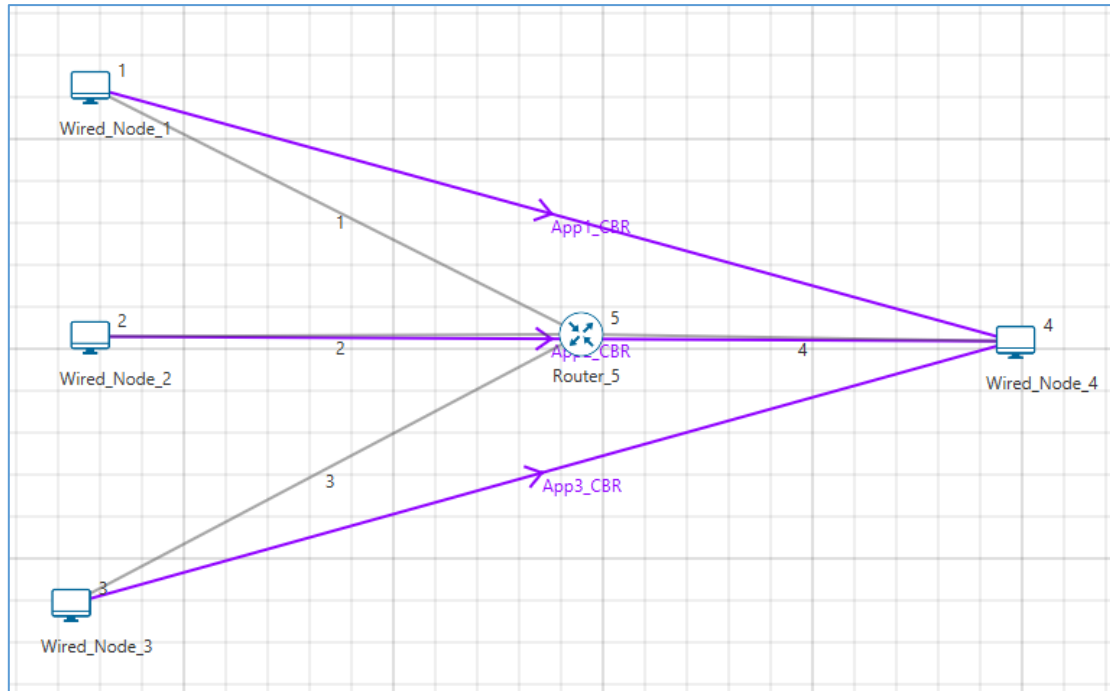
|     |      |
|-----|------|
| 3.2 | 1.67 |
| 3.3 | 1.67 |
| 3.4 | 1.67 |
| 3.5 | 1.67 |

## 4.2 Traffic Generation Rate and Service Rate

Open NetSim and click **Examples > Internetworks > Traffic-Gen-Rate-vs-Service-Rate** as shown below:



The following network diagram illustrates, what the NetSim UI displays when you open the example configuration file.



### Settings done in the Network:

- 1 TCP is disabled in all the Wired Nodes
- 2 Generation rate = 10Mbps for each application (Packet Size: 1460, Inter Arrival Time: 1168μs)

$$\text{Generation Rate (Mbps)} = (\text{Packet size (bytes)} * 8) / \text{Inter arrival time (}\mu\text{s)}$$

- 3 The traffic generation rate can be modified by changing application properties. Note that the generation rate should be less than or equal to service rate for steady-state simulation, where the service rate is defined as the data rate supported by the Bottle-neck link. In this case, there is no bottle neck link since all links support up to 100 Mbps
- 4 Packet Trace is Enabled
- 5 Simulate for 100s and note down the throughput
- 6 Go back to the scenario and change the link speed (both Uplink and Downlink Speed) between Router\_5 and Wired\_Node\_4 from the default 100 Mbps to 25 Mbps. In this case, the link between Router\_5 and Wired\_Node\_4 becomes a Bottle-neck link, since the link rate (i.e. service rate) is less than the generation rate of 30 Mbps (10 \* 3).

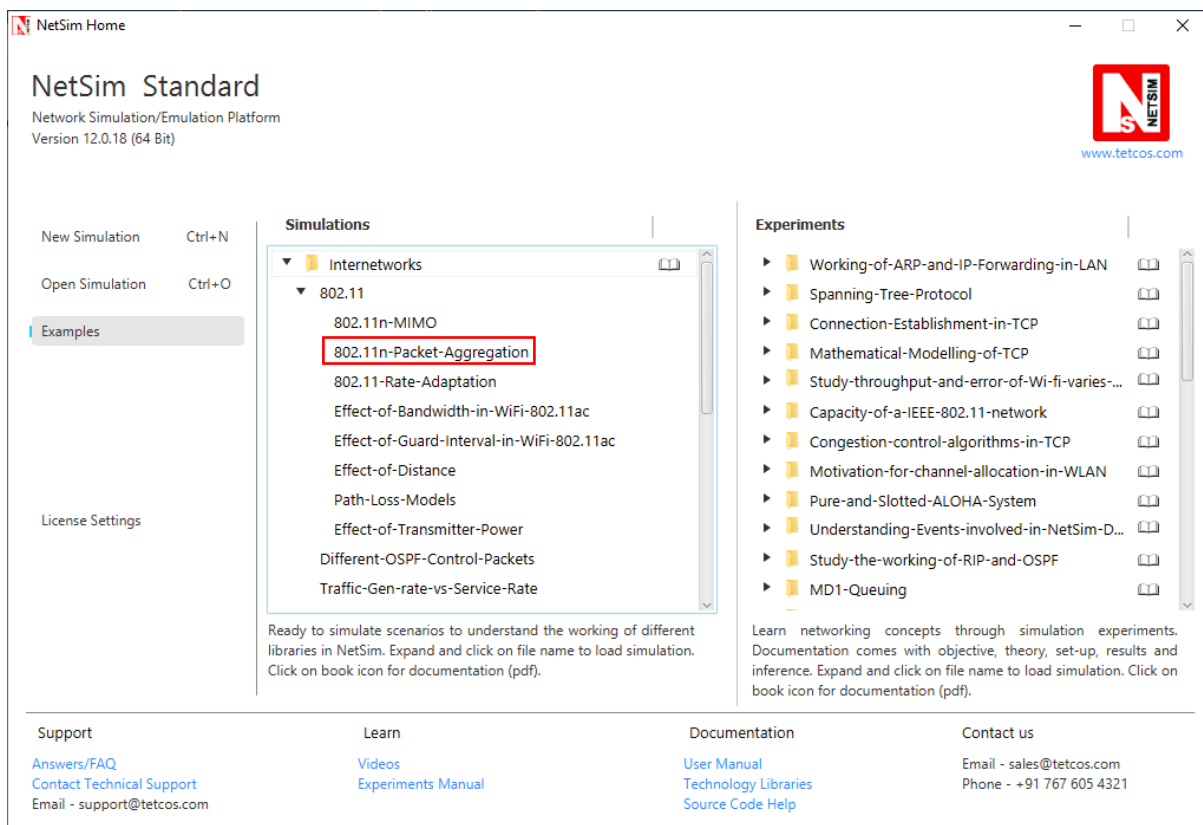
### Inference:

**Sample1:** In this scenario, router receives packets from three links at the rate of 10 Mbps each, a total of 30 Mbps. And the router-node link supports 100 Mbps. Hence there is no queuing / packet drop in the Router. The application throughput would be approximately equal to the generation rate.

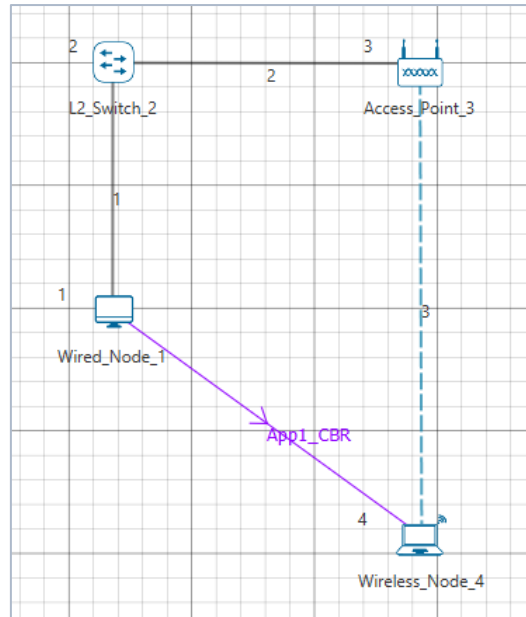
**Sample2:** In this case, the bottleneck link supports only 25 Mbps. Due to this, packets get accumulated in the router's buffer, which overflows after reaching its limit and hence router starts dropping the packets. Application throughput would be approximately equal to the bottle neck link capacity.

### 4.3 Packet aggregation in 802.11n

Open NetSim and click **Examples > Internetworks > 802.11 > 802.11n-Packet-Aggregation** as shown below:



The following network diagram illustrates, what the NetSim UI displays when you open the example configuration file.



### Settings done in the Network:

1. In the Environment Settings, Grid length is set to 50m \* 50m
2. Distance between Access Point and the Wireless Node is 20m
3. Packet Trace option is enabled
4. TCP is disabled in both Wired and Wireless Nodes
5. WLAN Standard is set to 802.11n and No. of Packets to Aggregate is set to 1. (Right-Click Access Point > Properties > Interface Wireless > No. of Packet to Aggregate)
6. Channel Characteristics: Path Loss Only, Path Loss Model: Log Distance, and Path Loss Exponent: 3. (Wireless Link Properties)
7. CBR Application with 100 Mbps Generation Rate (Packet Size: 1460, Inter Arrival Time: 116μs)
8. Simulate for 10s and note down the throughput
9. Go back to the scenario and increase the No. of Packets to Aggregate to 5 and 10 respectively, and note down the application throughputs.

### Output:

| Number of Packets Aggregated | Application Throughput |
|------------------------------|------------------------|
| 1                            | 23.26 Mbps             |
| 5                            | 43.89 Mbps             |
| 10                           | 53.46 Mbps             |

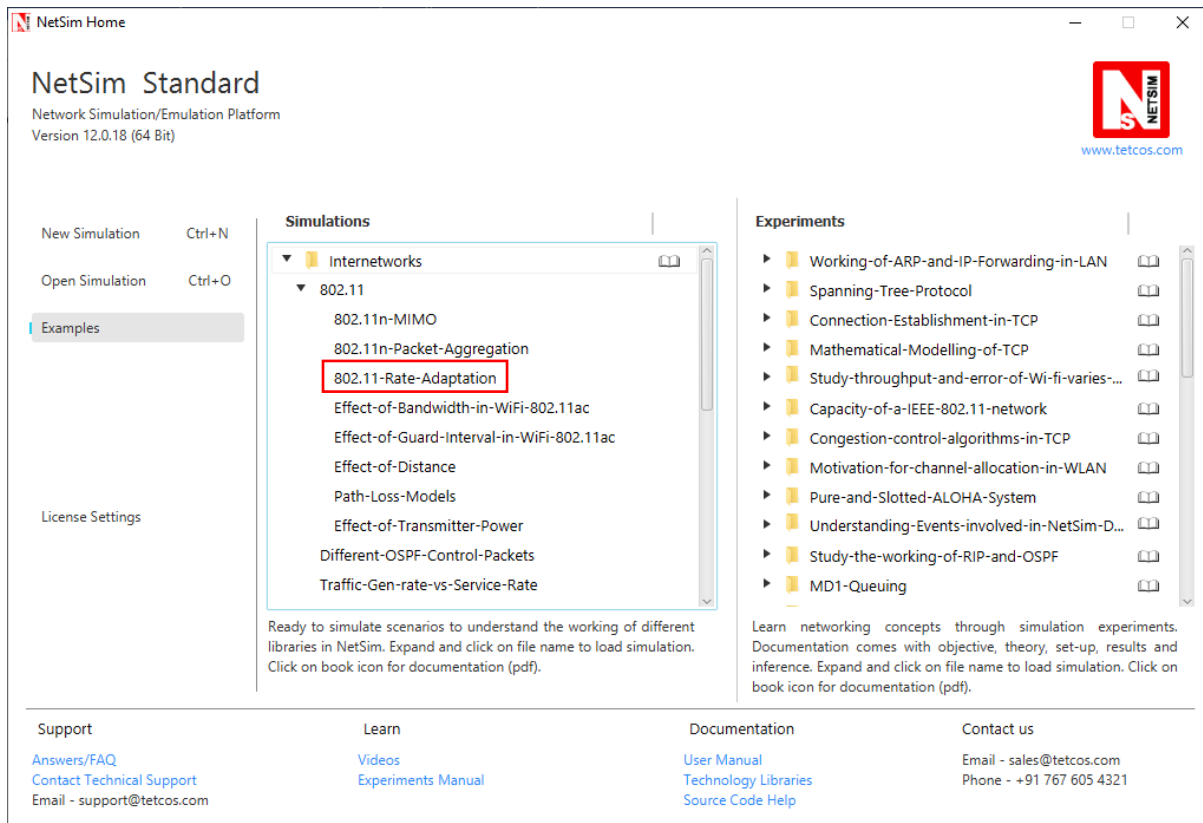
**Inference:**

- Packet aggregation is responsible for joining multiple MSDUs into a single MPDU that can be delivered to the physical layer as a single unit for transmission. As we increase the number of packets aggregated it results in lesser number of ack's. Hence, more data packets are transmitted per unit time leading to a higher application throughput.
- For No. of Packets to Aggregate is set to 5, we get five successive packets followed by a WLAN\_Block\_Ack (which is used to acknowledge that five packets are received successfully). Users can observe this in Packet Trace by filtering Tx\_ID as Access Point and Rx\_ID as Wireless Node
- Note that in the early stages of the simulation the AP would transmit whatever the number of frames/packets in its buffer. It will not wait for 5 packets to be aggregated (if say number of packets aggregated is 5). If Access Point buffer has more than 5 packets, it will aggregate 5 packets and then send. After sending 5 Packets it will receive one WLAN\_Block\_Ack.

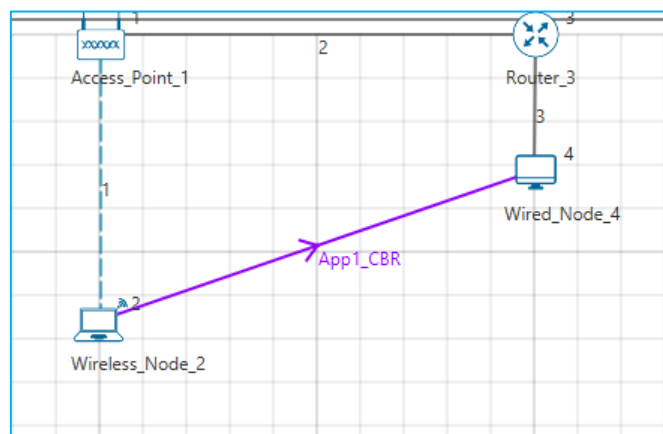
## 4.4 Rate Adaptation in 802.11b

Rate adaptation is the determination of the optimal data transmission rate most appropriate for current wireless channel conditions. It consists of assessing channel conditions and accordingly adjusting the rate. The Rate adaptation algorithm decides whether to increase or decrease the data rate based on consecutive transmission success/failures. In NetSim under the generic rate adaptation option, the data rate is decreased when four consecutive transmission attempts fail and is increased when twenty consecutive transmission attempts succeed.

Open NetSim, Select Examples->Internetworks->802.11->802.11-Rate-Adaptation as shown below:



The following network diagram illustrates what the NetSim UI displays when you open the example configuration file for Rate Adaptation.



#### Settings done in example config file:

1. Environment Grid length: 500m \* 500m
2. Distance between AP and Wireless Node is 65m
3. Enabled Packet Trace option
4. TCP → Disabled in Wireless Node
5. Set WLAN Standard → 802.11b

6. Propagation model → Path Loss only, Path Loss Model → Log Distance and Path loss Exponent → 3.25.
7. CBR application with 10Mbps generation rate (Set Packet Size: 1460 Bytes, Inter Arrival Time: 1168 micro sec)
8. Simulate for 10 sec.

### Output:

Open Packet Trace and filter Packet Type to CBR, Transmitter\_ID to Node-2 and then calculate Phy rate. Phy rate can be calculated using packet trace by using the formula shown below:

$$\text{Phy rate (802.11b)} = \text{Phy\_layer\_payload} * 8 / (\text{phy end time} - \text{phy arrival time} - 192)$$

192 micro seconds is the approximate preamble time for 802.11b

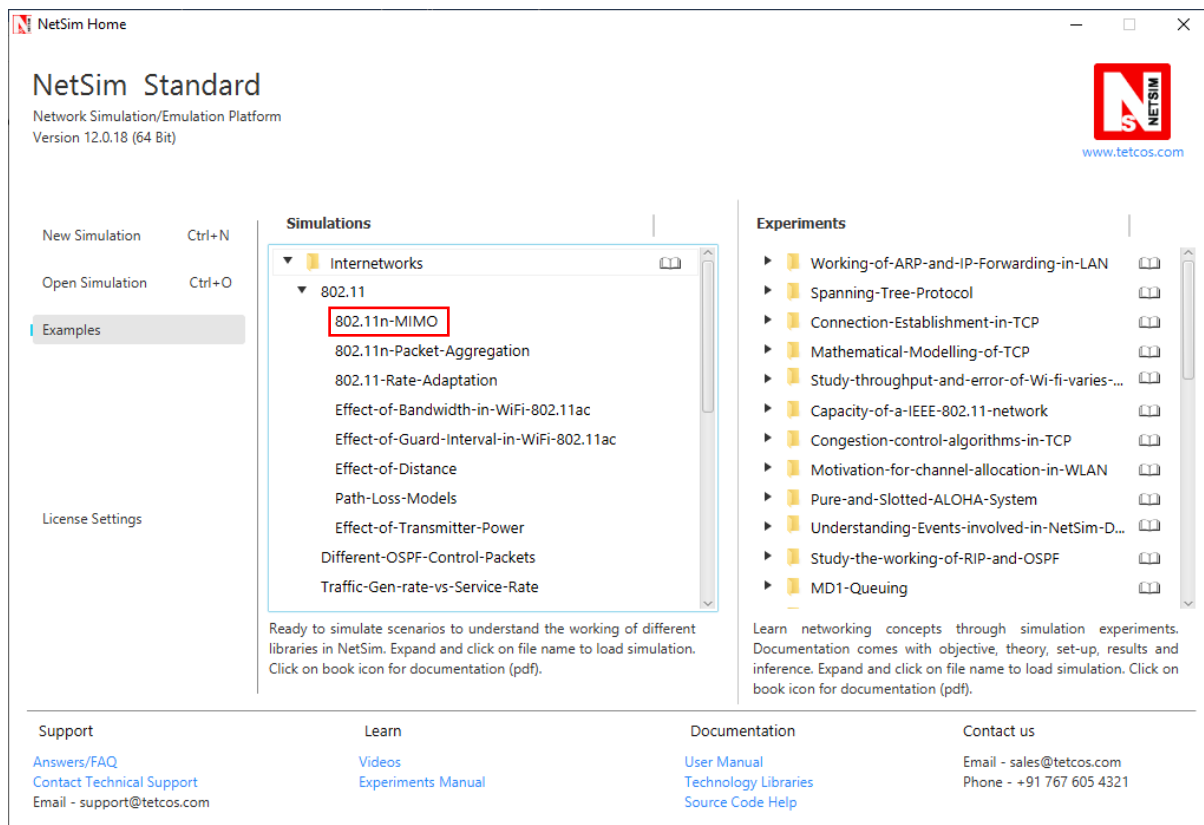
Calculate PHY rate for all the data packets coming from Wireless node to Access Point. For doing this please refer section 7.5.1 How to set filters to NetSim Packet Trace file from NetSim's User Manual.

|      | TRX_LAYER_PAYLOAD(Bytes) | NW_LAYER_PAYLOAD(Bytes) | MAC_LAYER_PAYLOAD(Bytes) | PHY_LAYER_PAYLOAD(Bytes) | PHY_LAYER_OVERHEAD(Bytes) | Column      | PACKET_STATUS | LOCAL_ADDRESS | REMOTE_ADDRESS |
|------|--------------------------|-------------------------|--------------------------|--------------------------|---------------------------|-------------|---------------|---------------|----------------|
| 9524 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 10.9927069  | Successful    | N/A           | N/A            |
| 9528 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 10.9927069  | Successful    | N/A           | N/A            |
| 9532 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 10.9927069  | Errored       | N/A           | N/A            |
| 9533 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 10.9927069  | Errored       | N/A           | N/A            |
| 9534 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 10.9927069  | Errored       | N/A           | N/A            |
| 9535 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 10.9927069  | Errored       | N/A           | N/A            |
| 9536 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9540 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9544 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9548 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9552 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9556 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9560 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9564 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9568 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9572 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9576 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9580 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9584 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9588 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9592 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9596 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9600 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9604 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9608 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9612 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 5.498850657 | Successful    | N/A           | N/A            |
| 9616 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 10.9927069  | Errored       | N/A           | N/A            |
| 9617 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 10.9927069  | Errored       | N/A           | N/A            |
| 9618 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 10.9927069  | Successful    | N/A           | N/A            |
| 9622 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 10.9927069  | Successful    | N/A           | N/A            |
| 9626 | 1460                     | 1468                    | 1488                     | 1528                     | 0                         | 10.9927069  | Successful    | N/A           | N/A            |

In the above screenshot, the Phy rate is reduced from 11Mbps to 5.5Mbps since there are four consecutive data error packets and again increased from 5.5Mbps to 11Mbps since there are twenty consecutive successful data packets.

## 4.5 802.11n MIMO

Open NetSim, Select **Examples → Internetworks → 802.11 → 802.11n-MIMO** as shown below:



### Settings done in sample network:

1. Environment Grid length: 50m \* 50m
2. Distance between AP and Node is 20m.
3. TCP → Disabled in Wireless and Wired Node.
4. WLAN Standard → 802.11n.
5. Propagation model → Path Loss only, Path Loss Model -> Log Distance and Path loss Exponent → 3.
6. CBR application with 50Mbps generation rate.(Set Inter Arrival Time : 233 (micro sec))
7. Simulate for 10 sec.

### Result:

| Number of Tx and Rx Antennae | Throughput |
|------------------------------|------------|
| 1 x 1                        | 23.26 Mbps |
| 2 x 2                        | 29.98 Mbps |
| 3 x 3                        | 32.13 Mbps |
| 4 x 4                        | 34.62 Mbps |

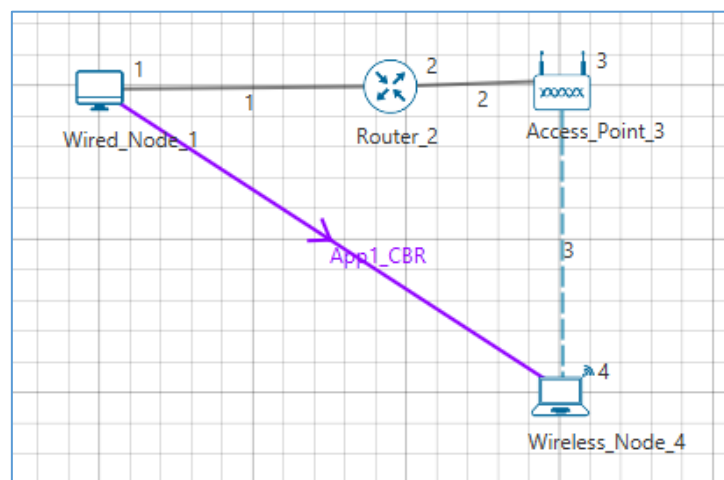
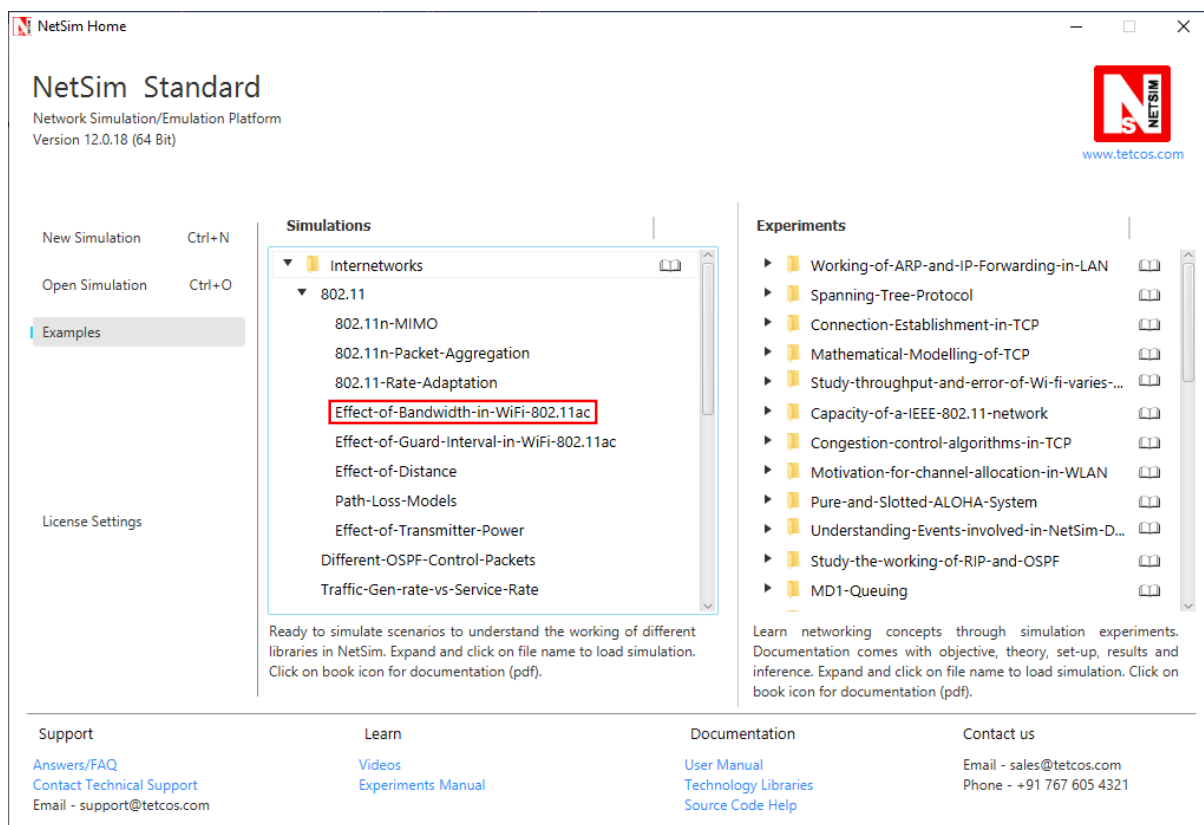
MIMO is a method for multiplying the capacity of a radio link using multiple transmit and receive antennas to exploit multipath propagation. Increasing the Transmitter and Receiver Antenna

count results in more PHY Data rate (link capacity) and hence leading to an increased application throughput.

## 4.6 Effect of Bandwidth and Guard Interval in WiFi 802.11ac

### Effect of Bandwidth:

Open NetSim and click **Examples > Internetworks > 802.11 > Effect-of-bandwidth-in-WiFi-802.11ac**.



### Settings done in the network:

1. Disabled TCP in Wired\_Node\_1
2. Set No\_Path\_Loss under Channel Characteristics
3. Set 802.11ac standard in Access Point and Wireless Node
4. Set Bit Error rate and Propagation delay to zero under wired link properties
5. Set Bandwidth to 20MHz under Wireless Interface->Physical Layer properties of access point
6. Enable packet trace
7. Set generation rate as 100 Mbps under Application properties (Packet Size = 1460 Bytes, Interarrival time = 116 micro seconds). Generation rate can be calculated by using the formula below:

$$\begin{aligned} \text{Generation Rate (Mbps)} &= \text{Packet Size (Bytes)} * 8 / \text{Interarrival time } (\mu\text{s}) \\ &= 1460 \text{ (Bytes)} * 8 / 116 (\mu\text{s}) \sim 100 \text{ Mbps} \end{aligned}$$

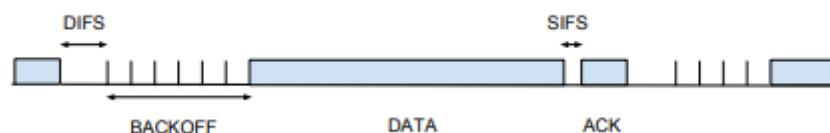
8. Run simulation for 10s and note down the throughput

### Theoretical calculations:

Average time to transmit a packet comprises of

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

The timing diagram is as shown below:



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

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

*Average time per packet (μs)*

$$= DIFS + Average Backoff time + Packet Transmission Time + SIFS + Ack Transmission Time$$

$$Packet Transmission Time (\mu s) = Preamble time + (MPDU Size / Data rate)$$

$$Average Backoff time (\mu s) = (CW_{min}/2) * Slot Time$$

$$Ack Transmission Time (\mu s) = Preamble time + (Ack Packet size / Ack data rate)$$

$$DIFS (\mu s) = SIFS + 2 * Slot Time$$

$$Average Backoff time (\mu s) = (CW_{min}/2) * Slot Time$$

where

$$Application payload = 1460 \text{ Bytes}$$

$$Average time per packet = 34 + 67.5 + 185.36 + 16 + 212.88 = 513.74 \mu s$$

$$SIFS = 16 \mu s$$

$$Slot time = 9 \mu s$$

$$CW_{min} = 15 \text{ slots for } 802.11ac$$

$$DIFS = SIFS + 2 * Slot Time = 16 \mu s + 2 * 9 \mu s = 34 \mu s$$

$$Average Backoff time = 67.5 \mu s$$

$$Packet Transmission Time = 44 \mu s + (1532 * 8 / 86.7 \text{ Mbps}) = 185.36 \mu s$$

$$Preamble time = 44 \mu s \text{ for } 802.11ac \text{ standard}$$

$$MPDU Size = 1460 + 8 + 20 + 44 = 1532 \text{ Bytes}$$

$$Ack Transmission Time = 44 \mu s + (152 \text{ Bytes} * 8 / 7.2 \text{ Mbps}) = 212.88 \mu s$$

$$Average throughput = 1460 * 8 / (513.74) = 22.7 \text{ Mbps}$$

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:

$$Phy \text{ rate } (802.11b) = Phy\_layer\_payload * 8 / (phy \text{ end time} - phy \text{ arrival time} - 44)$$

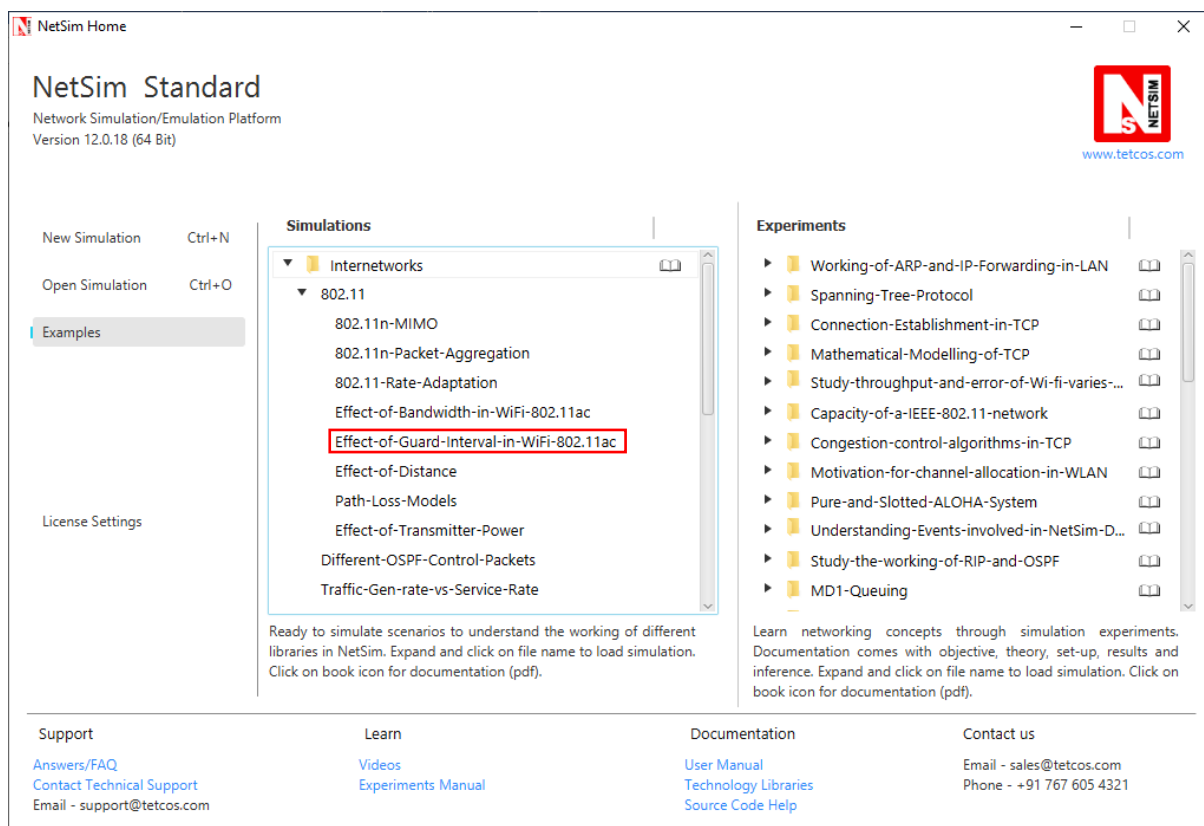
## Output:

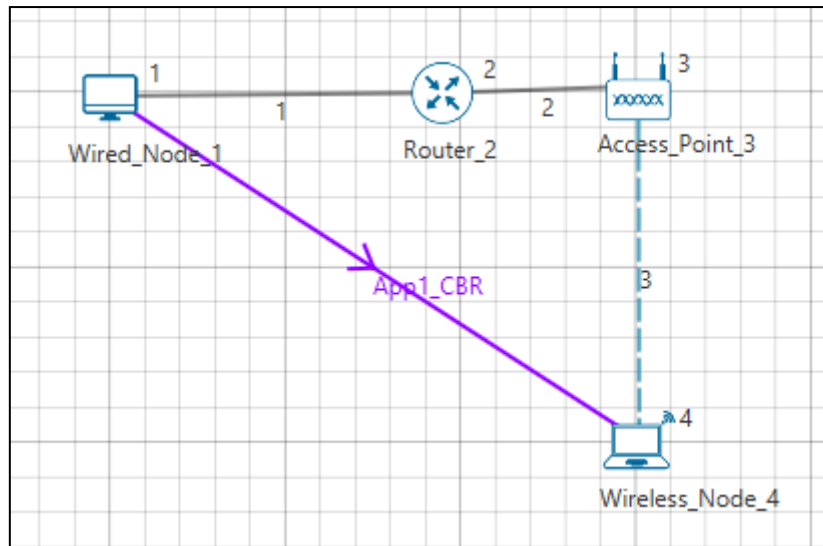
| Bandwidth (MHz) | Theoretical Throughput (Mbps) | Simulation Throughput (Mbps) |
|-----------------|-------------------------------|------------------------------|
| 20              | 22.70                         | 22.78                        |
| 40              | 33.77                         | 33.78                        |
| 80              | 43.39                         | 43.48                        |
| 160             | 49.35                         | 49.56                        |

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

## Effect of Guard Interval

Open NetSim and click **Examples > Internetworks > 802.11 > Effect-of-Guard-Interval-in-WiFi-802.11ac**.





### Settings done in the network:

1. Disabled TCP in Wired\_Node\_1
2. Set No\_Path\_Loss under Channel Characteristics
3. Set 802.11ac standard in Access Point and Wireless Node
4. Set Bit Error rate and Propagation delay to zero under wired link properties
5. Enable packet trace
6. Set Guard interval to 400ns under Wireless Interface->Physical Layer properties of access point
7. Set generation rate as 100 Mbps under Application properties (Packet Size = 1460 Bytes, Interarrival time = 116 micro seconds). Generation rate can be calculated by using the formula below:

$$\text{Generation Rate (Mbps)} = \text{Packet Size (Bytes)} * 8 / \text{Interarrival time } (\mu\text{s})$$

$$= 1460 \text{ (Bytes)} * 8 / 116 \text{ } (\mu\text{s}) \sim 100 \text{ Mbps}$$

8. Run simulation for 10s and note down the throughput

Calculate throughput theoretically as explained above and compare with Simulation throughput.

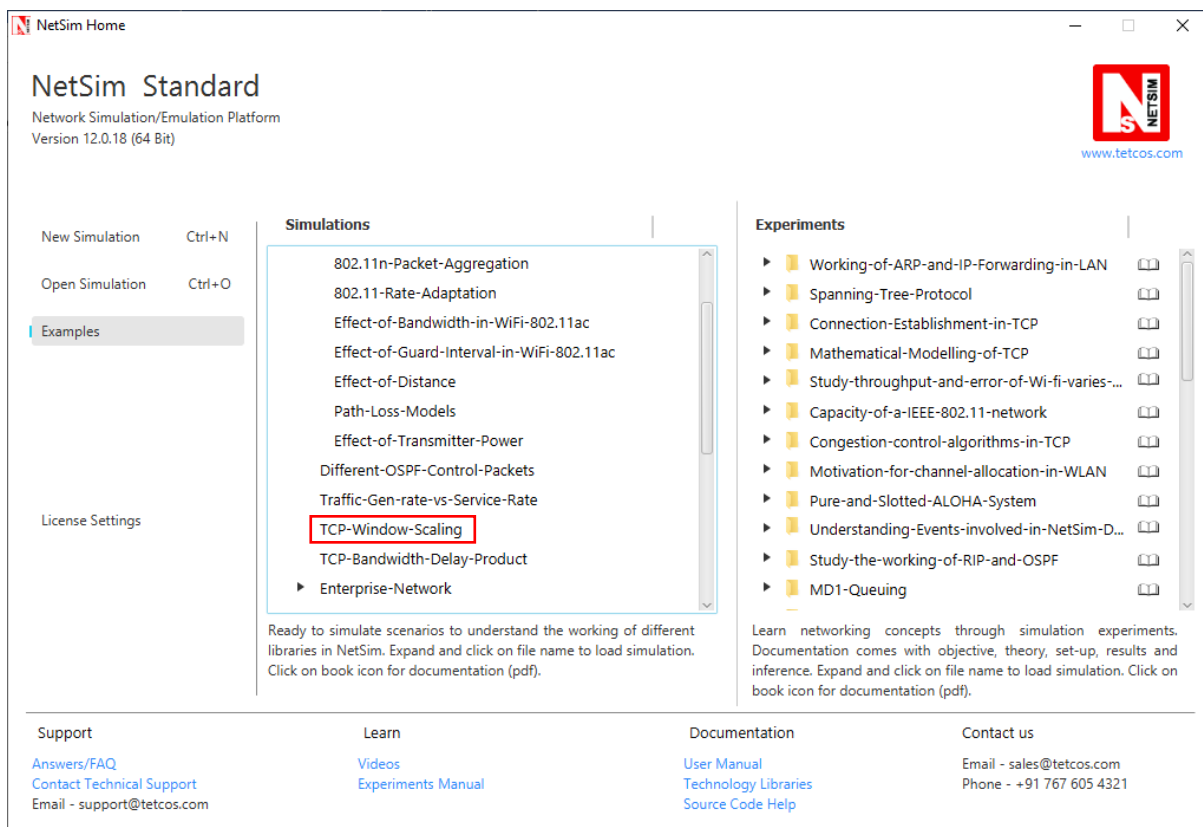
### Output:

| Guard Interval (ns) | Theoretical Throughput (Mbps) | Simulation Throughput (Mbps) |
|---------------------|-------------------------------|------------------------------|
| 400                 | 17.76                         | 22.78                        |

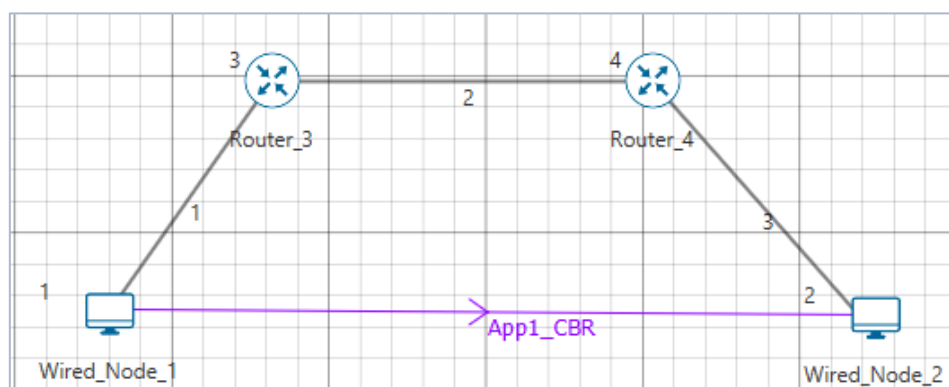
|     |       |       |
|-----|-------|-------|
| 800 | 16.87 | 21.33 |
|-----|-------|-------|

## 4.7 TCP Window Scaling

Open NetSim, Select Examples->Internetworks->TCP-Window-Scaling as shown below:



The following network diagram illustrates what the NetSim UI displays when you open the example configuration file for 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

(congestion control); the receive window tries not to exceed the capacity of the receiver to process data (flow control).

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.

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  $(1,500,000 * 0.513) = 769,500$  bits or about 96,187 bytes.

Using a maximum window size of 64 KB only allows the buffer to be filled to  $(65,535 / 96,187) = 68\%$  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 Mbit/second utilizing all of the available bandwidth.

#### Settings done in example config file:

1. Transport Layer TCP Window Scaling → FALSE (by default)
2. Application Generation rate → 10Mbps (Set Inter arrival time = 1168)
3. Bit error rate(Uplink and Downlink) → 0 in all wired links
4. Enabled Wireshark Capture in General Properties Wired Node 1 → Set as Offline
5. Link1 & Link3 Propagation delay (uplink and downlink) → 5(Microsec) (by default)
6. Change the Link2 speed → 10Mbps, Propagation delay(uplink and downlink) - >100000 (Microsec)
7. Simulate for 100sec and note down the throughput
8. Now change the Window Scaling → TRUE (for all wired nodes)
9. Simulate for 100s and note down the throughput

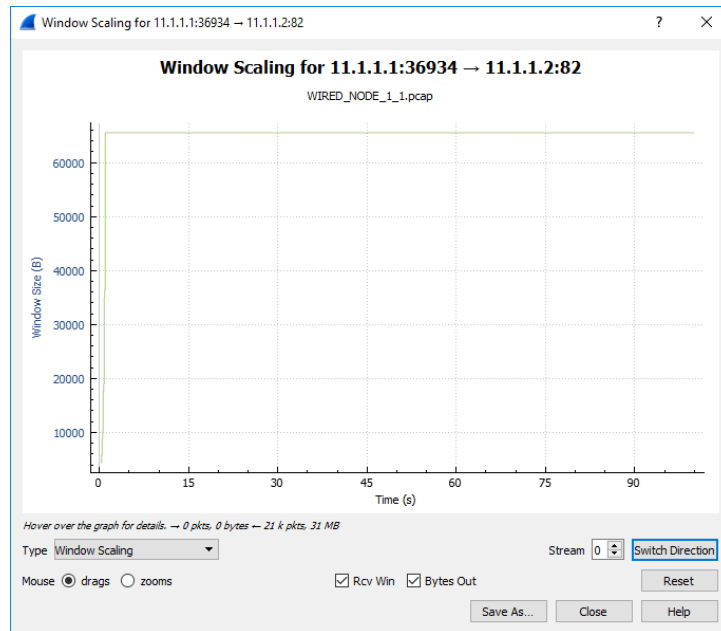
#### Output:

| Window Scaling | Application_Throughput (Mbps) |
|----------------|-------------------------------|
| FALSE          | 2.5                           |
| TRUE           | 8.7                           |

## Throughput calculation (Without Window Scaling)

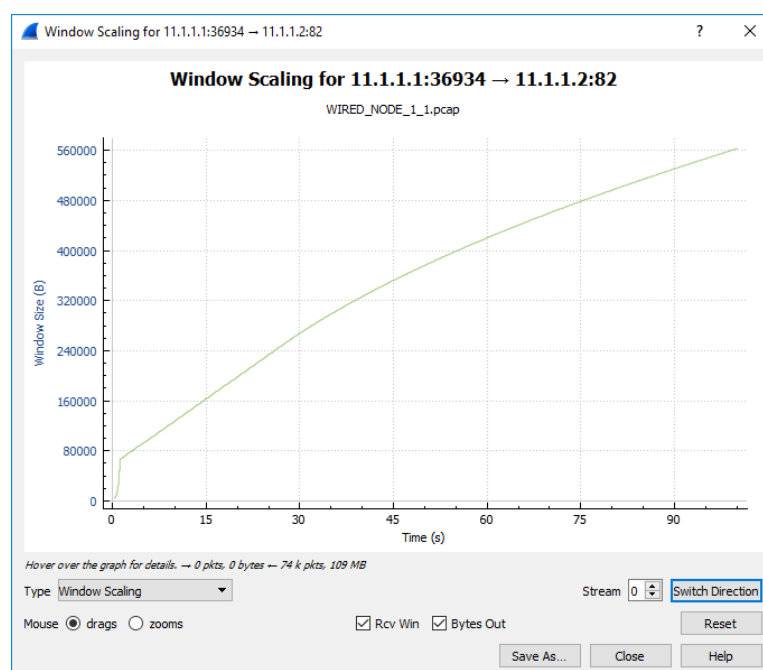
Theoretical Throughput = Window size / Round trip time =  $65535 \times 8 / 200\text{ms} = 2.62\text{Mbps}$

### Without Window Scaling:



In case 1 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. Users can notice the Window size in Wireshark. Please refer section 7.7.5 in User Manual for creating Wireshark graphs.

### With Window Scaling:



From the above screenshot users can notice that the window size grows up to 560192Bytes because of Window Scaling. This leads to a higher Application\_Throughput compared to the case without window scaling.

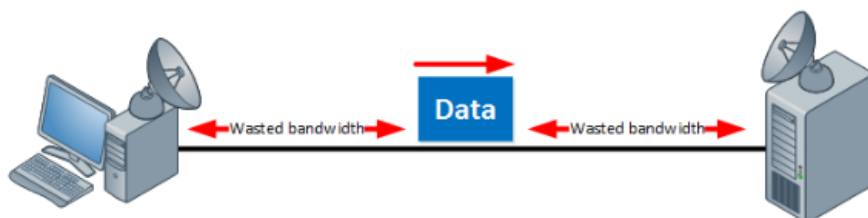
## 4.8 TCP Bandwidth Delay Product

The Bandwidth Delay Product determines the amount of data that can be in transit in the network. It is the product of the available bandwidth and the latency, or RTT.

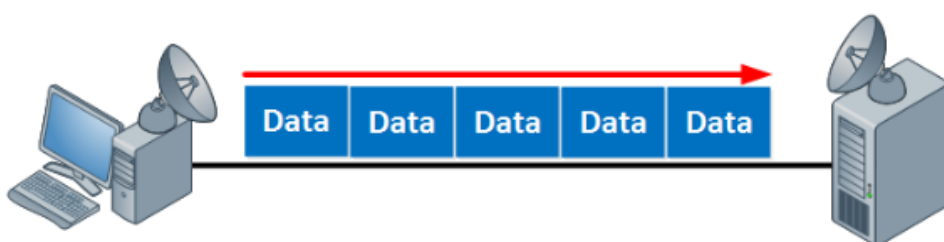
$$BDP(bits) = Total\ bandwidth\ \left(\frac{bits}{sec}\right) * Propagation\ Delay\ (sec)$$

Consider a **long fat networks** (LFN). A LFN is a network that offers a high bandwidth but also a very high delay. An example could be a satellite connection. These connections offer a high bandwidth and high delay of about 500-1000 ms.

The problem here is that when the sender sends some data, it has to wait a very long time for an acknowledgment of the receiver before it can send the next data. During the time we are waiting, nothing happens so we are not utilizing the full bandwidth of our link. The throughput of TCP is limited by the round-trip time of the link and the window size. We can't change the round-trip time but we can play with the window size. Take a look at the image below:

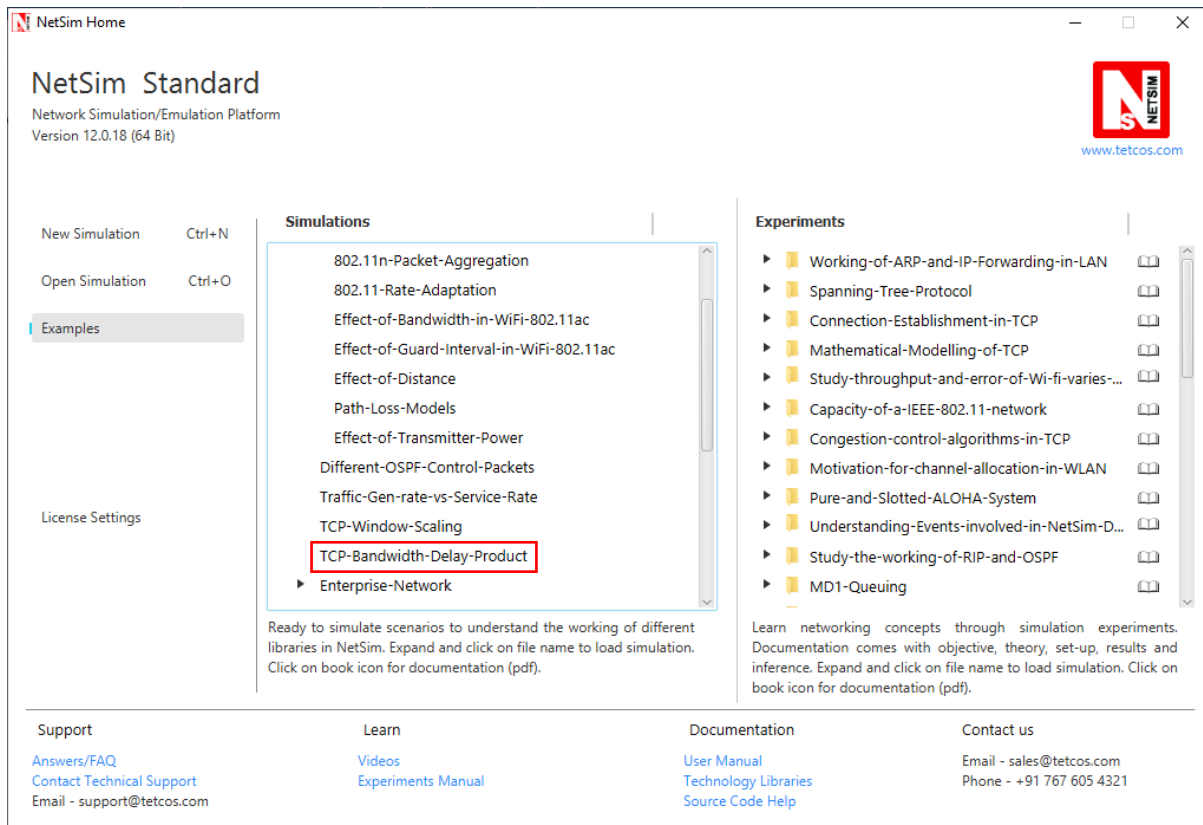


Imagine we send some data from the host to the server, when this piece of data is on its way we have to wait a long time before it reaches the server and for the acknowledgment to come back. A lot of bandwidth is wasted. This can be overcome with a large window size shown below:

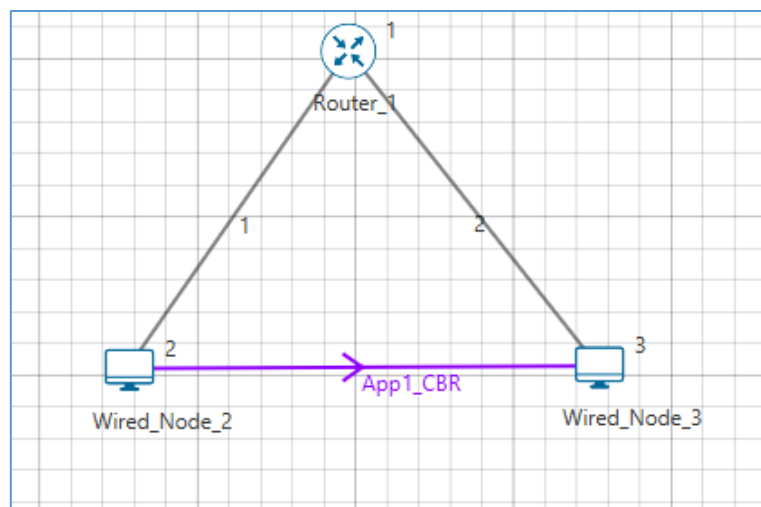


With a large window size, we can fill the entire “pipeline” with data. So there will be no wastage of bandwidth.

Open NetSim, Select Examples->Internetworks->TCP-Bandwidth-Delay-Product as shown below:



The following network diagram illustrates what the NetSim UI displays when you open the example configuration file for TCP Bandwidth Delay Product.



### Settings done in example config file:

1. Transport Layer TCP Window Scaling → FALSE (by default)
2. Application Generation rate → 10Mbps (Set Packet size = 1460 Bytes, Inter arrival time = 1168 μs)
3. Bit error rate (Uplink and Downlink) → 0 in all wired links
4. Enable Wireshark Capture in General Properties Wired Node 2 → Set as Offline
5. Change the Link speed → 10Mbps in all wired links
6. Set Propagation delay (Uplink and Downlink) → 100000 μs
7. Simulate for 100sec and note down the throughput
8. Now for Wired\_Node\_2 change the Window Scaling → TRUE for Wired Node-2 and set Scale shift count=2
9. Simulate for 100s and note down the throughput

### Theoretical calculations

#### Sample1:

In Sample1, BDP (bits) = total\_available\_bandwidth (bits/sec) x round\_trip\_time (sec)

$$= (10 \times 10^6) \times (400 \times 10^{-3}) = 4 \times 10^6 = 4 \text{ Mb}$$

Window Size = 65kB = 520kb = 0.52Mb

Throughput = Window Size/RTT = 0.52/0.4 = 1.25Mbps

#### Sample2:

In Sample2, BDP (bits) = Total\_available\_bandwidth (bits/sec) x Round\_Trip\_Time (sec)

$$= (10 \times 10^6) \times (400 \times 10^{-3}) = 4 \times 10^6 = 4 \text{ Mb}$$

Window Size = 65kB = 520Kb = (0.52) Mb\*(2^2) (2^2 represents 2^scale\_shift\_count)

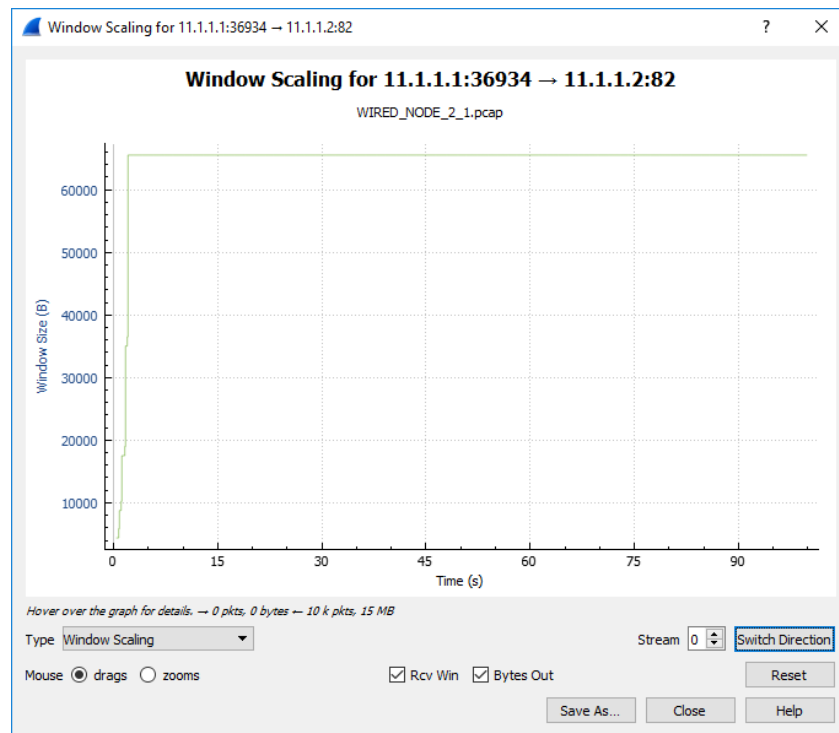
Throughput = Window Size/RTT = 0.52\*4/0.4 = 5.2Mbps

### Output:

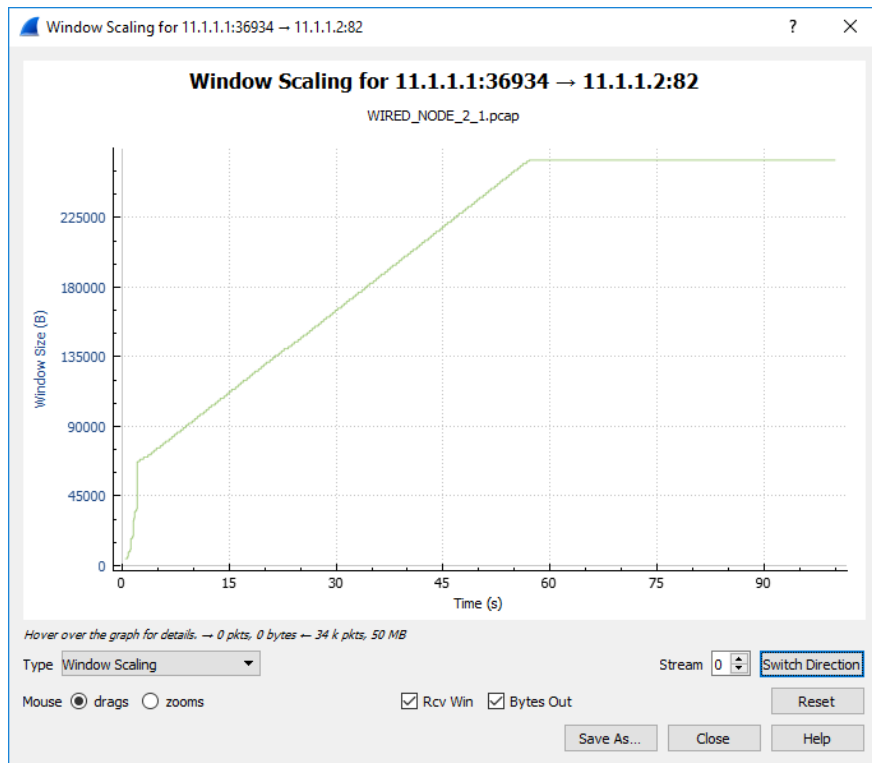
| Sample | Theoretical throughput<br>(Mbps) | NetSim throughput<br>(Mbps) |
|--------|----------------------------------|-----------------------------|
|--------|----------------------------------|-----------------------------|

|   |      |      |
|---|------|------|
| 1 | 1.25 | 1.25 |
| 2 | 5.2  | 4.01 |

**Sample1:** In this sample, even though we have a link capacity of 10 Mbps we are only getting 1.25 Mbps throughput because TCP window is limited to 65KB. Users can notice the Window size in Wireshark. Please refer section 7.7.5 in User Manual for creating Wireshark graphs



**Sample2:** In this sample also, Link capacity is 10 Mbps and window scaling is enabled. Window scaling allows till 5.2 Mbps, but since the BDP is limited to 4Mb, the throughput seen is 4 Mbps. From the below screenshot users can notice that the window size grows up to 2.08 Mb because of Window Scaling.



## 4.9 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 11.1.1.1 and 11.1.1.2 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 (10.0.0.0).

Similarly, if you create a network with a router and two wired nodes, the IP addresses are assigned as 11.1.1.2 and 11.2.1.2 for the two wired nodes. The subnet mask is default as in above case, i.e., 255.255.0.0. The IP address of the router is 11.1.1.1 and 11.2.1.1 respectively for the two interfaces. Both the nodes are in different networks (11.1.0.0 and 11.2.0.0) in this case.

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

## 4.10 Configuring Static Routing in NetSim

**Static Routing:**

Routers forward packets using either route information from route table entries that 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 unroutable packets are sent).

### **How to Setup Static Routes**

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

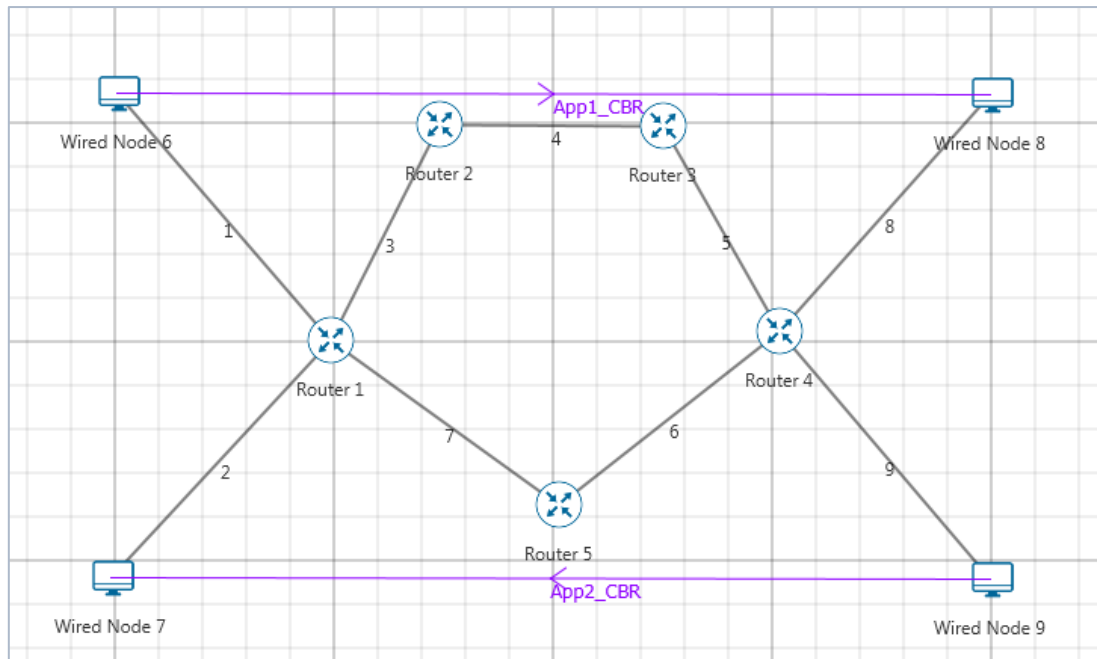
Static route configuration prior to simulation:

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

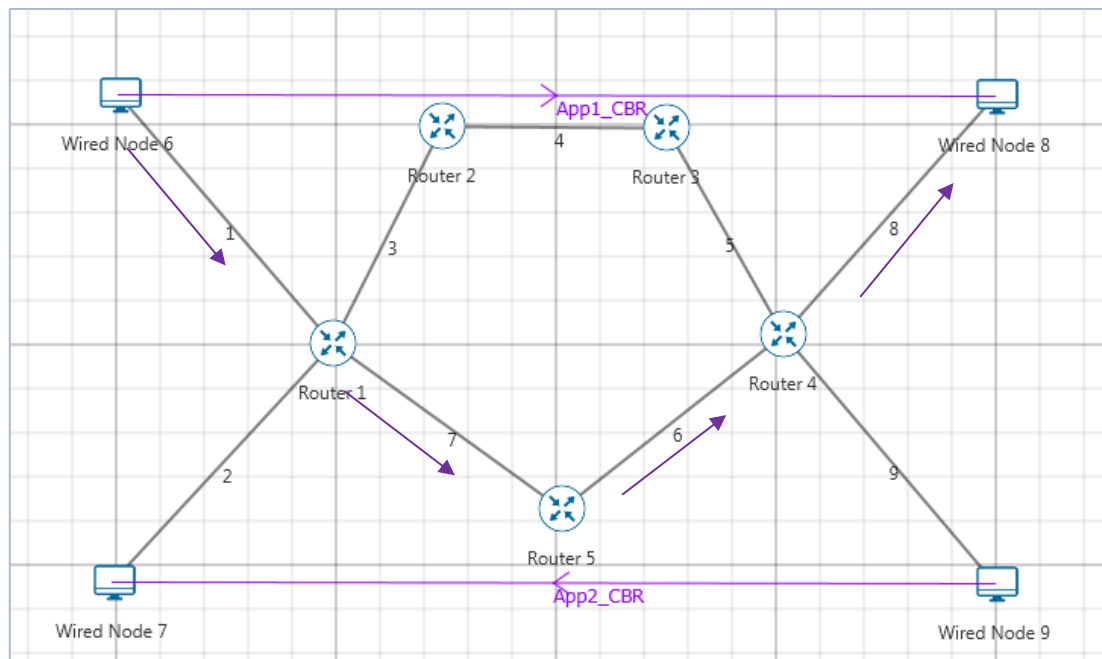
Static route configuration during the simulation:

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

### **Static route configuration via GUI – Static IP Routing Dialogue**



Create a scenario as per the screenshot above and disable TCP in all nodes. Run simulation for 10 seconds and open packet animation.



The default routing protocol is OSPF. So, packets will reach destination via Router 5. (Refer related experiment in experiment manual for more information).

**Static routing:**

Open Router 1 properties->Network\_Layer. Click on configure Static Route IP and set the properties as per the screenshot below and click on Add and then click on OK.

The screenshot shows a 'Static IP Routing Dialogue' window. It contains the following fields and controls:

- Network Destination:** Text box with '11.3.0.0'
- Subnet Mask:** Text box with '255.255.0.0'
- Interface ID:** Dropdown menu with '2' selected
- Gateway:** Text box with '11.4.1.2'
- Metrics:** Text box with '1'
- Buttons:** 'Default', 'Add' (highlighted with a red box), and 'Remove'
- Table:** A table with headers: 'Network Destination', 'Subnet Mask', 'Gateway', 'Metrics', and 'Interface ID'. The table body is empty, with the text 'No content in table' centered below the headers.
- Bottom Buttons:** 'OK' and 'Cancel'

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

#### Router 1:

```
route ADD 11.3.0.0 MASK 255.255.0.0 11.4.1.2 METRIC 1 IF 3
```

**route ADD destination\_ip 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

**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.

Similarly follow the same procedure for remaining entries of Router 1, Router 2, Router 3, Router 4 as per the following:

route ADD 11.9.0.0 MASK 255.255.0.0 11.4.1.2 METRIC 1 IF 3

route ADD 11.1.0.0 MASK 255.255.0.0 11.1.1.2 METRIC 1 IF 1

route ADD 11.2.0.0 MASK 255.255.0.0 11.2.1.2 METRIC 1 IF 2

#### **Router 2:**

route ADD 11.3.0.0 MASK 255.255.0.0 11.5.1.2 METRIC 1 IF 2

route ADD 11.9.0.0 MASK 255.255.0.0 11.5.1.2 METRIC 1 IF 2

route ADD 11.1.0.0 MASK 255.255.0.0 11.4.1.1 METRIC 1 IF 1

route ADD 11.2.0.0 MASK 255.255.0.0 11.4.1.1 METRIC 1 IF 1

#### **Router 3:**

route ADD 11.3.0.0 MASK 255.255.0.0 11.6.1.2 METRIC 1 IF 2

route ADD 11.9.0.0 MASK 255.255.0.0 11.6.1.2 METRIC 1 IF 2

route ADD 11.1.0.0 MASK 255.255.0.0 11.5.1.1 METRIC 1 IF 1

route ADD 11.2.0.0 MASK 255.255.0.0 11.5.1.1 METRIC 1 IF 1

#### **Router 4:**

route ADD 11.3.0.0 MASK 255.255.0.0 11.3.1.2 METRIC 1 IF 3

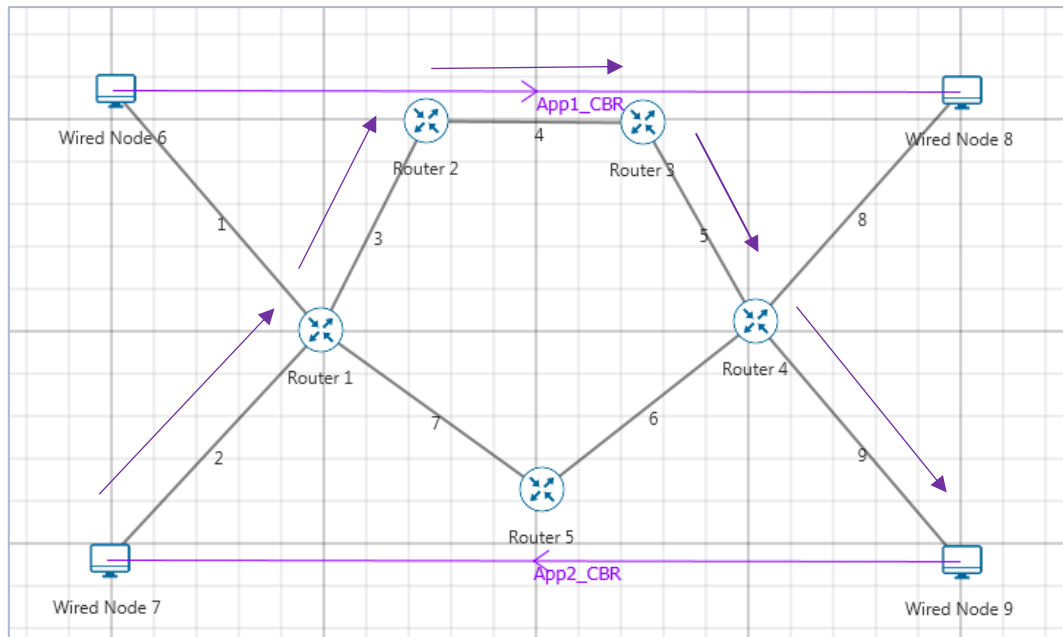
route ADD 11.9.0.0 MASK 255.255.0.0 11.3.1.2 METRIC 1 IF 4

route ADD 11.1.0.0 MASK 255.255.0.0 11.6.1.1 METRIC 1 IF 1

route ADD 11.2.0.0 MASK 255.255.0.0 11.6.1.1 METRIC 1 IF 1

After configuring the router properties, run simulation for 10 seconds and check packet animation. Now the packets will reach destination as per the above routes:

**Router 1→Router 2→Router 3→Router 4**



## Disabling Static Routing

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

## 4.11 Different OSPF Control Packets

There are five distinct OSPF packet types.

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

### 1. The Hello packet

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 packets 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 neighbour'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.

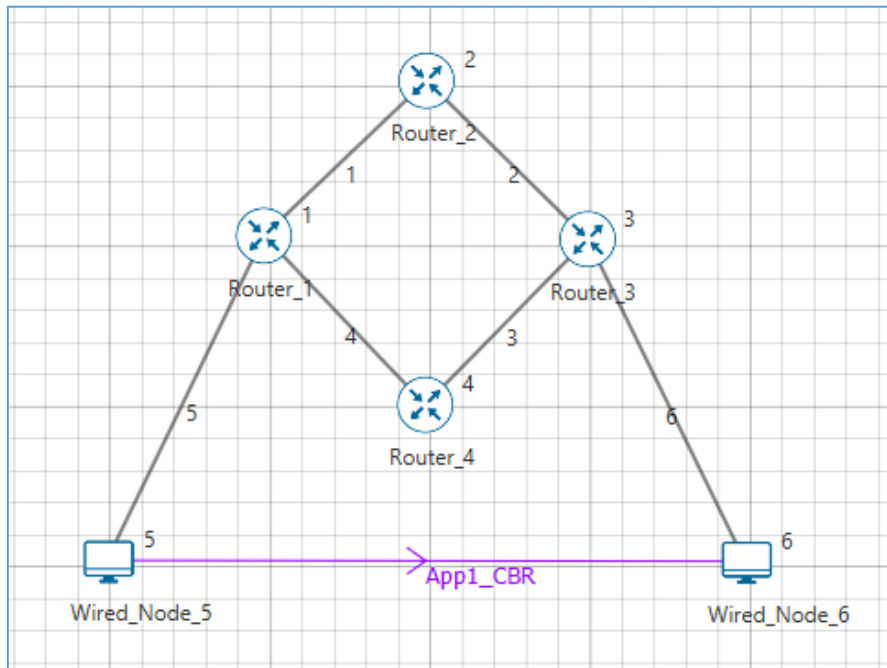
## **4. The Link State Update packet**

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

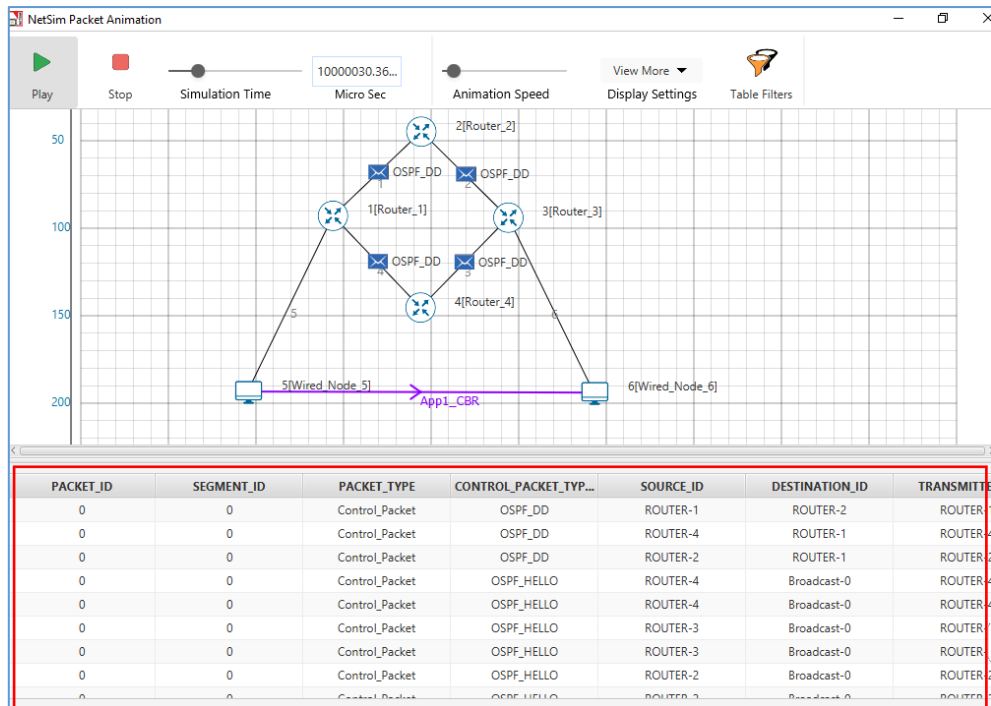


#### Settings done in sample network:

1. Set OSPF Routing protocol under Application\_Layer properties of a router
2. Configured CBR application with default properties
3. Enabled Packet Trace
4. Simulate for 100 sec.

#### Output:

Open Packet animation window and click on play button. Users can observe all the OSPF packets. 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.



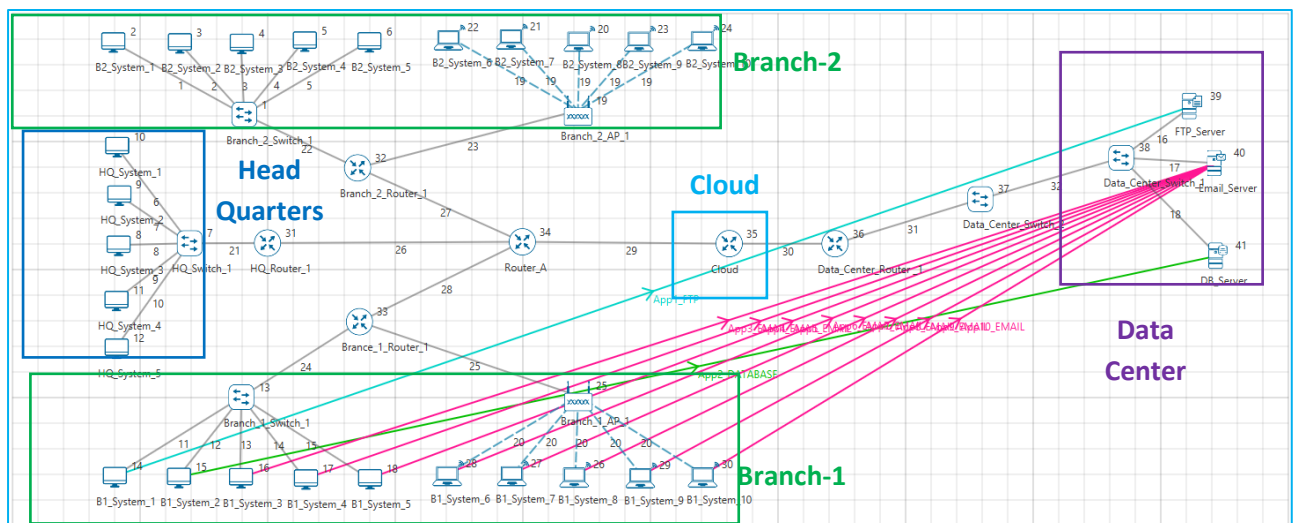
The same can be observed in Packet trace by filtering CONTROL\_PACKET\_TYPE/ APP\_NAME to OSPF\_HELLO, OSPF\_DD, OSPF\_LSACK, OSPF\_LSUPDATE and OSPF\_LSREQ packets as shown below:

| 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 |

## 4.12 Enterprise Network

Create a simple enterprise network, comprising of, two branches, head-quarters and a data center. Branches and headquarter are connected to the data center over the public cloud. In NetSim, users can model the network, by just adding the network elements, through click and drop, and renaming them suitably as shown below:



**Settings done in the network:**

**Sample1:**

Open NetSim, Select Examples->Internetworks->Enterprise-Network ->Sample-1

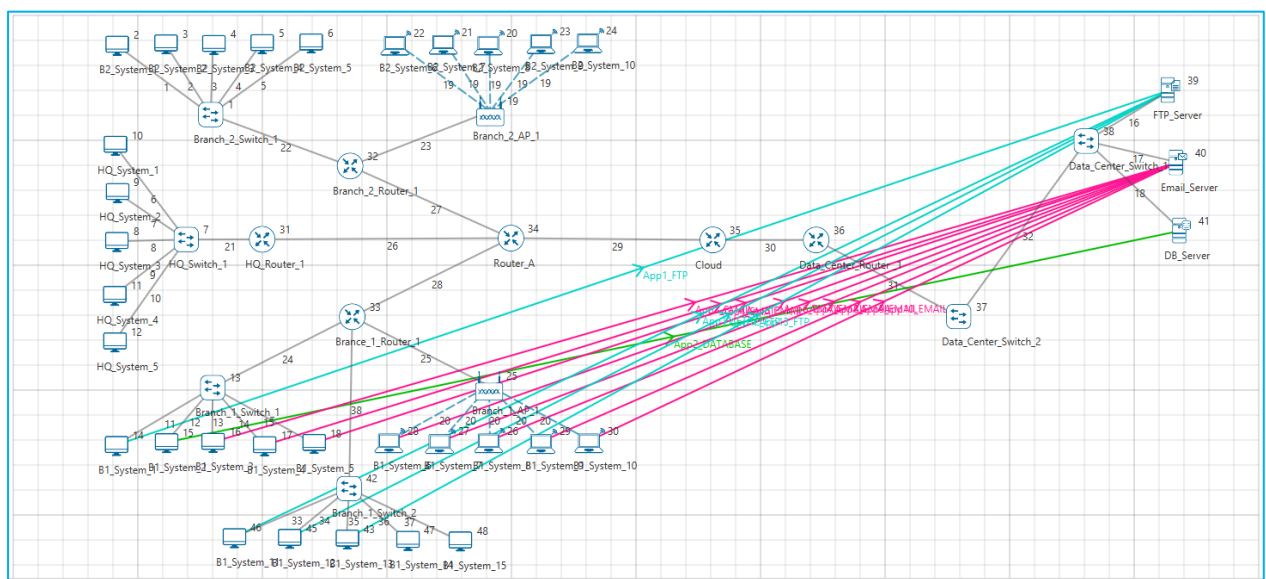
1. We have changed the link rate for the outbound link i.e. Link 28 from Branch 1 as 2Mbps

2. Configured one FTP application from 14 to the file server 39, a Database application from 15 to the Database server 41, and eight email applications running between 16, 17, 18, 26, 27, 28, 29, 30 and the Email server 40.
3. Simulated for 100s

### Sample2:

Open NetSim, Select Examples->Internetworks->Enterprise-Network ->Sample-2

1. In this sample, we have added more nodes via the switch and configured 3 FTP applications from systems 43, 45, 46 to FTP server 39 as shown in the below screenshot:



2. Simulated for 100 seconds

### Sample3:

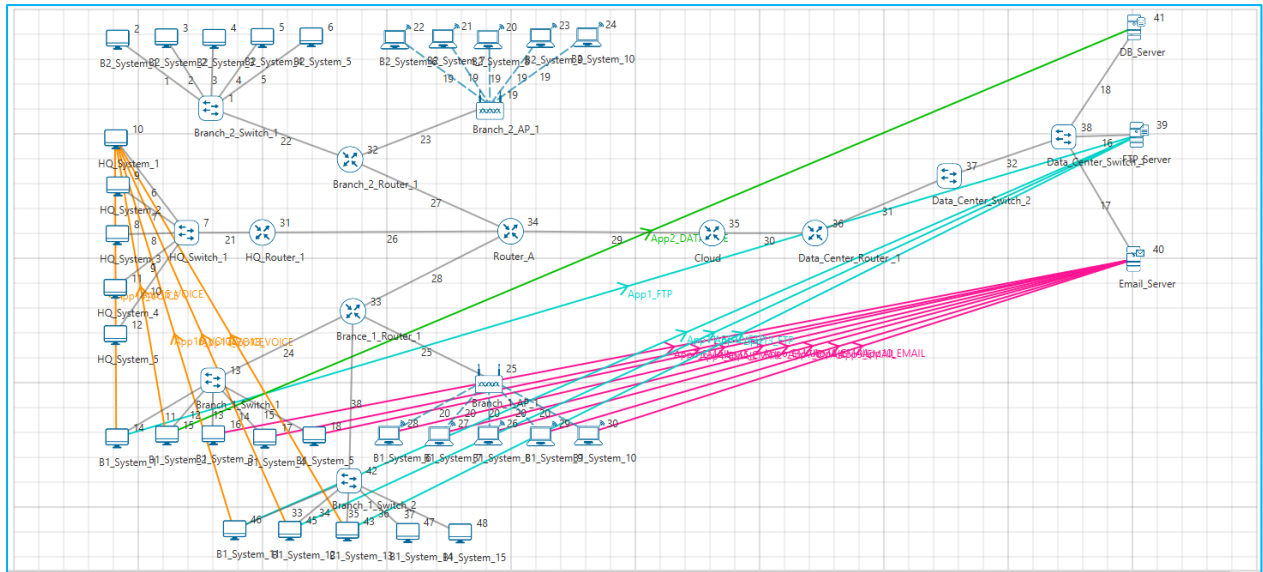
Open NetSim, Select Examples->Internetworks->Enterprise-Network ->Sample-3

1. In this sample, we have changed the outbound link speed i.e. Link 28 to 4Mbps and simulated for 100 seconds

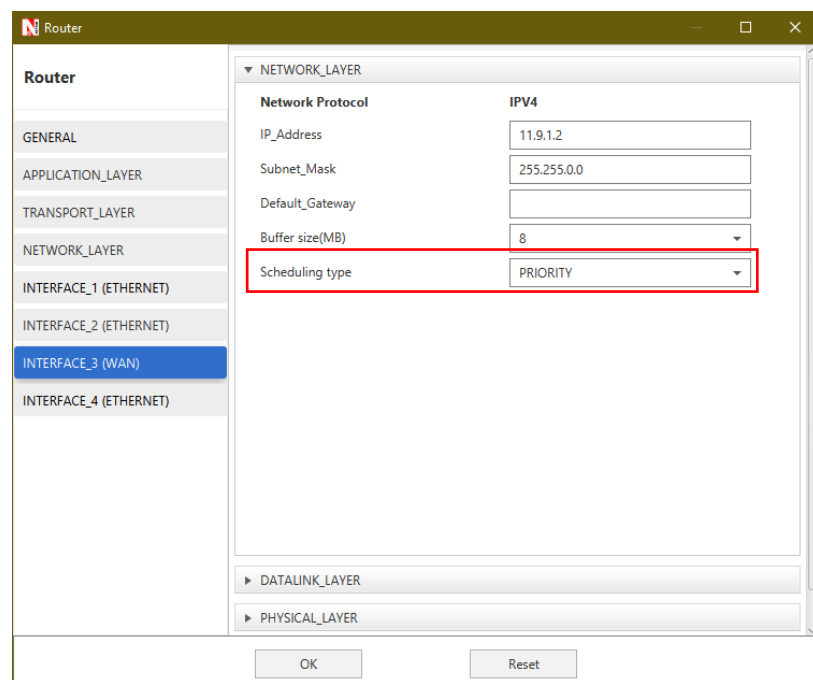
### Sample4:

Open NetSim, Select Examples->Internetworks->Enterprise-Network ->Sample-4

1. In this sample, we have changed the outbound link speed i.e. Link 28 to 2Mbps and configured Voice applications from 14, 15, 46, 45 and 43 to Head office 10 as shown in the below screenshot



2. Also changed Scheduling type to Priority under Network Layer Properties of Router33 Interface\_WAN properties as shown below:



3. Simulated for 100 seconds

### Sample5:

Open NetSim, Select Examples->Internetworks->Enterprise-Network ->Sample-5

1. In this sample, we have changed the start time for Voice and FTP applications to 40 seconds, email application to 30 seconds.
2. Enabled Plots and simulated for 100 seconds

## Output:

### Sample1:

Open metrics window and calculate the average delay for e-mail application present under Application properties shown below:

| Application_Metrics_Table                                  |                                             |                  |                  |                 |                   |                 |                  |
|------------------------------------------------------------|---------------------------------------------|------------------|------------------|-----------------|-------------------|-----------------|------------------|
| Application_metrics <input type="checkbox"/> Detailed View |                                             |                  |                  |                 |                   |                 |                  |
| Application Id                                             | Throughput Plot                             | Application Name | Packet generated | Packet received | Throughput (Mbps) | Delay(microsec) | Jitter(microsec) |
| 1                                                          | <a href="#">Application throughput plot</a> | App1_FTP         | 3850             | 3411            | 0.567298          | 4095837.077554  | 314102.983791    |
| 2                                                          | <a href="#">Application throughput plot</a> | App2_DATABASE    | 490              | 469             | 0.076622          | 1946525.824476  | 237844.681038    |
| 3                                                          | <a href="#">Application throughput plot</a> | App3_EMAIL       | 966              | 803             | 0.131120          | 4415240.588795  | 120645.225828    |
| 3                                                          | <a href="#">Application throughput plot</a> | App3_EMAIL       | 966              | 966             | 0.157714          | 1035908.886449  | 66435.364396     |
| 4                                                          | <a href="#">Application throughput plot</a> | App4_EMAIL       | 966              | 952             | 0.155429          | 875172.002020   | 20161.933874     |
| 4                                                          | <a href="#">Application throughput plot</a> | App4_EMAIL       | 966              | 966             | 0.157714          | 411224.920138   | 42968.398025     |
| 5                                                          | <a href="#">Application throughput plot</a> | App5_EMAIL       | 966              | 879             | 0.143550          | 2927376.105362  | 103504.527465    |
| 5                                                          | <a href="#">Application throughput plot</a> | App5_EMAIL       | 966              | 890             | 0.145335          | 2037412.228867  | 86213.944840     |
| 6                                                          | <a href="#">Application throughput plot</a> | App6_EMAIL       | 966              | 942             | 0.153810          | 1425117.898370  | 66003.535722     |
| 6                                                          | <a href="#">Application throughput plot</a> | App6_EMAIL       | 966              | 731             | 0.119358          | 2579269.360978  | 102179.402476    |
| 7                                                          | <a href="#">Application throughput plot</a> | App7_EMAIL       | 966              | 857             | 0.139929          | 1280611.183388  | 68174.784012     |
| 7                                                          | <a href="#">Application throughput plot</a> | App7_EMAIL       | 966              | 939             | 0.153310          | 1506358.739867  | 81189.837784     |
| 8                                                          | <a href="#">Application throughput plot</a> | App8_EMAIL       | 966              | 910             | 0.148571          | 1179612.450721  | 57133.640390     |
| 8                                                          | <a href="#">Application throughput plot</a> | App8_EMAIL       | 966              | 567             | 0.092597          | 11893518.337495 | 178419.793403    |
| 9                                                          | <a href="#">Application throughput plot</a> | App9_EMAIL       | 966              | 952             | 0.155429          | 1381537.985213  | 56555.288238     |
| 9                                                          | <a href="#">Application throughput plot</a> | App9_EMAIL       | 966              | 726             | 0.118574          | 2063831.923717  | 90708.044255     |

The delay experienced by the e-mail applications would be **3.20 seconds**

### Sample2:

In this Sample, the average delay for email applications has increased to **16.27 seconds** due to the impact of additional load on the network

### Sample3:

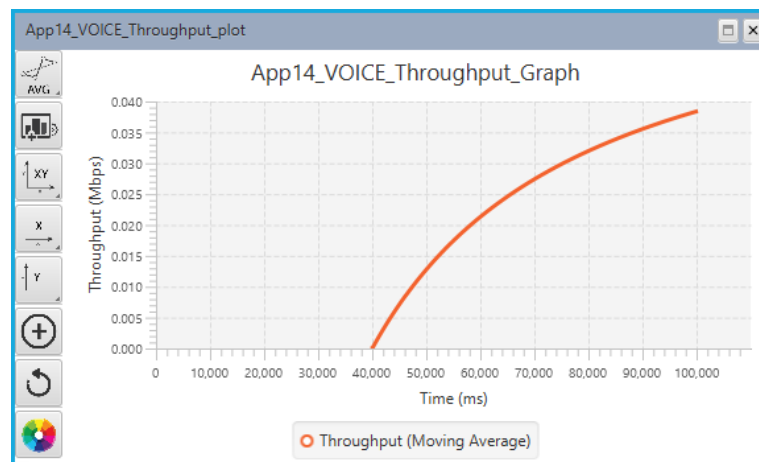
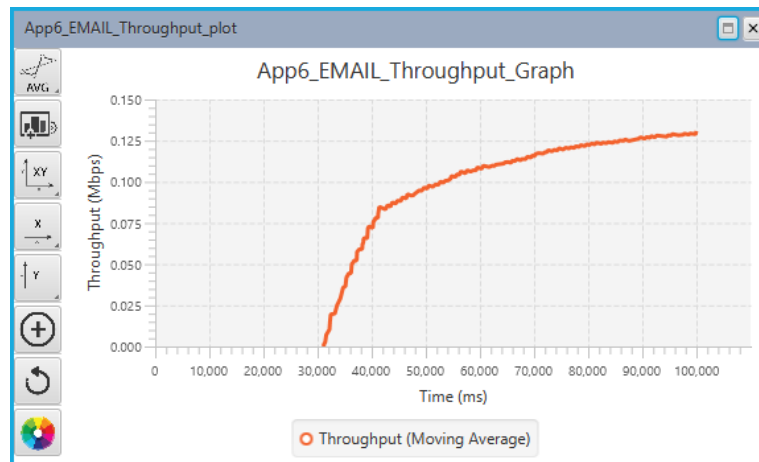
In this Sample, the average delay for e-mail applications has dropped down to **0.77 seconds**

### Sample4:

In this Sample, the average delay for the e-mail application has increased to **19.66 seconds** since voice has a higher priority over data, and the routers will first serve the voice packets in its queue and only then route the data packets

### Sample5:

In this Sample, users can notice that email application sees good throughput initially after which it is flat. On the other hand, the voice application throughput is 0 till 40 seconds, since it has no traffic, and starts picking up 40 seconds onwards. Note that these are moving average graphs and not instantaneous value plots



## 4.13 NetSim Interactive Simulation

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

**Working:** This section will demonstrate how to perform Interactive simulation for a simple network scenario.

Open NetSim and click **Examples > Simulations > Internetworks > Interactive-Simulation** as shown below:

# NetSim Standard

Network Simulation/Emulation Platform  
Version 12.0.18 (64 Bit)

www.tetcos.com

New Simulation Ctrl+N

Open Simulation Ctrl+O

**Examples**

License Settings

## Simulations

- 802.11-Rate-Adaptation
- Effect-of-Bandwidth-in-WiFi-802.11ac
- Effect-of-Guard-Interval-in-WiFi-802.11ac
- Effect-of-Distance
- Path-Loss-Models
- Effect-of-Transmitter-Power
- Different-OSPF-Control-Packets
- Traffic-Gen-rate-vs-Service-Rate
- TCP-Window-Scaling
- TCP-Bandwidth-Delay-Product
- Enterprise-Network
- Interactive-Simulation**

Ready to simulate scenarios to understand the working of different libraries in NetSim. Expand and click on file name to load simulation. Click on book icon for documentation (pdf).

## Experiments

- Working-of-ARP-and-IP-Forwarding-in-LAN
- Spanning-Tree-Protocol
- Connection-Establishment-in-TCP
- Mathematical-Modelling-of-TCP
- Study-throughput-and-error-of-Wi-fi-varies-...
- Capacity-of-a-IEEE-802.11-network
- Congestion-control-algorithms-in-TCP
- Motivation-for-channel-allocation-in-WLAN
- Pure-and-Slotted-ALOHA-System
- Understanding-Events-involved-in-NetSim-D...
- Study-the-working-of-RIP-and-OSPF
- MD1-Queuing

Learn networking concepts through simulation experiments. Documentation comes with objective, theory, set-up, results and inference. Expand and click on file name to load simulation. Click on book icon for documentation (pdf).

**Support**

[Answers/FAQ](#)

[Contact Technical Support](#)

Email - support@tetcos.com

**Learn**

[Videos](#)

[Experiments Manual](#)

**Documentation**

[User Manual](#)

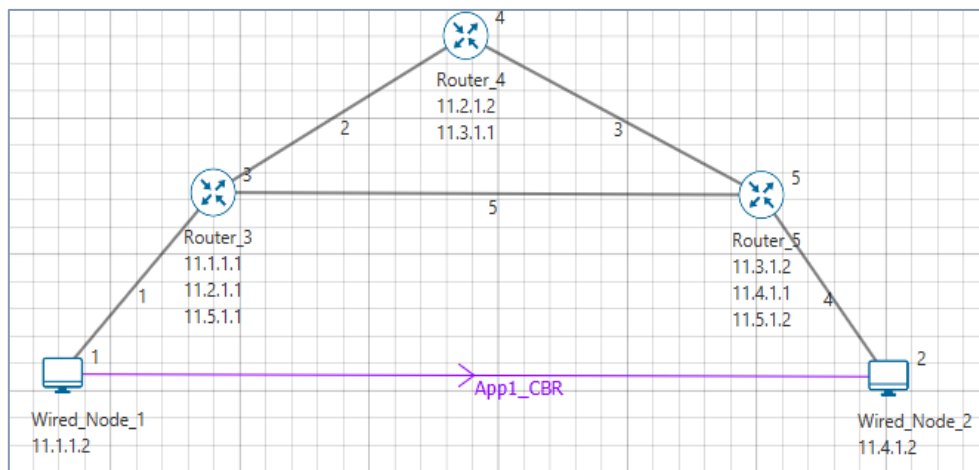
[Technology Libraries](#)

[Source Code Help](#)

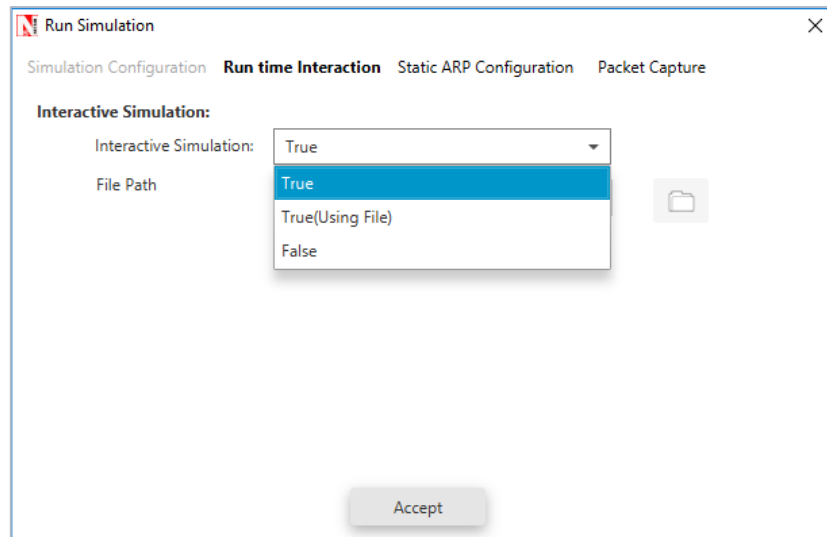
**Contact us**

Email - sales@tetcos.com

Phone - +91 767 605 4321



- Click on Application icon present in the top ribbon and set the Application type as CBR. The Source\_Id is 1 and Destination\_Id is 2.
- Set Start Time as 30 Sec
- Enable Packet trace
- Click on run simulation option and In the Run time Interaction tab set Interactive Simulation as True and click on Accept



- Click on run simulation and set Simulation Time as 500 sec. **(It is recommended to specify a longer simulation time to ensure that there is sufficient time for the user to execute the various commands and see the effect of that before Simulation ends)** and click OK
- Simulation (NetSimCore.exe) will start running and will display a message “waiting for first client to connect” as shown below:

```

C:\Program Files\NetSim Pro\bin\NetSimCore.exe
License Manager Output. Product>Edition>Maj_ver>Min_ver>Lic_type>Components>
netsim>pro>11>0>rlm_hw>00000000>000>

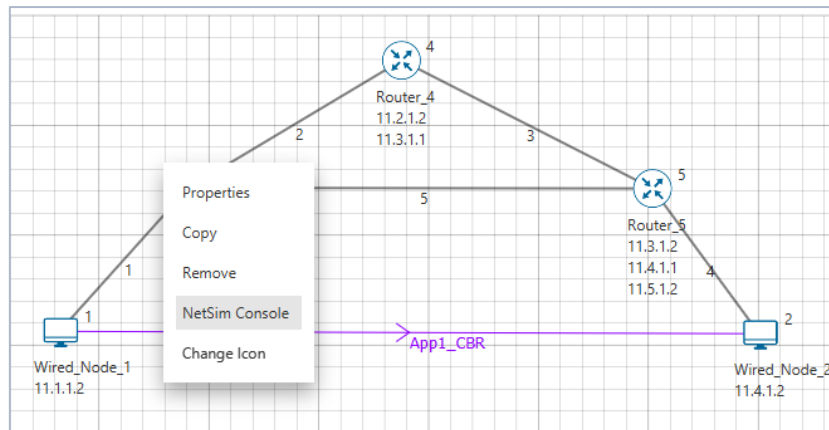
NetSim License Manager Start. Checking for licenses available (this may take upto 2 min) -

License Manager Output. Product>Edition>Maj_ver>Min_ver>Lic_type>Components>
netsim>pro>11>0>rlm_hw>11111111>110>
NetSim license validated
Installing heart-beat...
Heartbeat status = 0 (0 indicates successful)
NetworkStack loaded from path- C:\Program Files\NetSim Pro\bin\NetworkStack.dll

***
NetSim start
Network Stack loaded
Initializing simulation
Config file reading complete
License re-validation complete
Protocol binaries loaded
Stack variables initialized
Could Not Find C:\Users\TETC05~1\AppData\Local\Temp\NetSim\Plot_*
Metrics variables initialized
Protocol variables initialized
Executing command --- DEL "C:\Users\TETC05~1\AppData\Local\Temp\NetSim\*.pcap"
Could Not Find C:\Users\TETC05~1\AppData\Local\Temp\NetSim\*.pcap
Emulation is disabled
CLI mode is enable...
Waiting for first client to connect. Press ctrl+c to stop connection.

```

- After Simulation window opened goto Network scenario and right click on Router\_3 or any other node and select NetSim Console option



- Now Client (NetSimCLI.exe) will start running and it will try to establish connection with NetSimCore.exe. After connection is established the window will look similar like this shown below:

```

C:\Program Files\NetSim Pro\bin\NetSimCLI.exe

Initialising Winsock...Initialised.
Connecting to device DESKTOP-LC53CTS.
Connection attempt: 1
Connection established.

NetSim>

```

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

**Note: Commands are not a case sensitive**

### 1. Simulation specific (Not applicable for file based interactive simulation)

1. Pause
2. PauseAt
3. Continue
4. Stop
5. Exit
6. Reconnect

**Pause:** To pause the currently running simulation

**PauseAt:** To pause the currently running simulation with respect to particular time (Ex: To Pause simulation at 70.2 sec use command as **PauseAt 70.2**)

**Continue:** To start the currently paused simulation

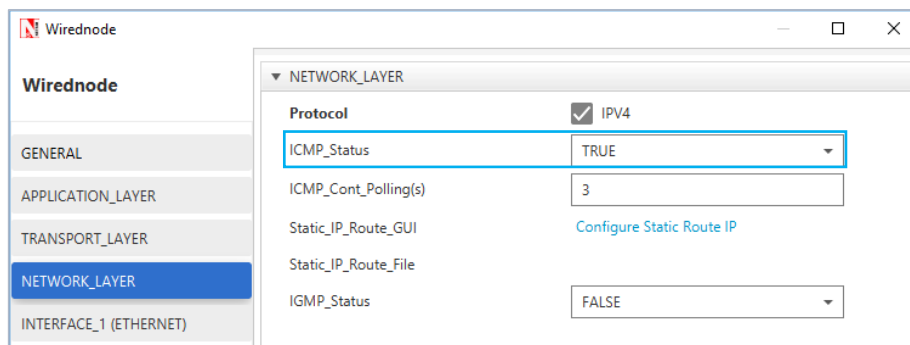
**Stop:** To stop the currently running simulation (NetSimCore.exe)

**Exit:** To exit from the client (NetSimCLI.exe)

**Reconnect:** To reconnect client (NetSimCLI.exe) to simulation (NetSimCore.exe) when we rerun simulation again

## 2. Ping Command

- The ping command is one of the most often used networking utilities for troubleshooting network problems
- You can use the ping command to test the availability of a networking device (usually a computer) on a network
- When you ping a device, you send that device a short message, which it then sends back (the echo)
- If you receive a reply then the device is in Network, if you don't then the device is faulty, disconnected, switched off, incorrectly configured
- You can use the **ping** command with an IP address or Device name
- **ICMP\_Status** should be set as True in all nodes(Wired\_Node and Router)



- Right click on Wired\_Node\_1 and go to properties. Under General properties enable Wireshark Capture option as “Online”

**Ping <IP address> e.g. ping 11.4.1.2**

**Ping <Node Name> e.g. ping Wired\_Node\_2**

## Ping Command Results:

```
C:\Program Files\NetSim Standard\bin\NetSimCLI.exe
Initialising Winsock...Initialised.
Connecting to device DESKTOP-I7H5EJ7.
Connection attempt: 1
Connection established.

NetSim>ping 11.4.1.2
Reply from 11.4.1.2: bytes 32 time=43us TTL=255
Reply from 11.4.1.2: bytes 32 time=43us TTL=255
Reply from 11.4.1.2: bytes 32 time=43us TTL=255
Reply from 11.4.1.2: bytes 32 time=43us TTL=255

NetSim>ping Wired_Node_2
Reply from 11.4.1.2: bytes 32 time=43us TTL=255
Reply from 11.4.1.2: bytes 32 time=43us TTL=255
Reply from 11.4.1.2: bytes 32 time=43us TTL=255
Reply from 11.4.1.2: bytes 32 time=43us TTL=255

NetSim>
```

- After simulation open packet trace and filter ICMP\_EchoRequest and ICMP\_EchoReply from CONTROL\_PACKET\_TYPE/APP\_NAME column

|      | A         | B          | C              | D                            | E         | F              | G              | H           |
|------|-----------|------------|----------------|------------------------------|-----------|----------------|----------------|-------------|
|      | PACKET_ID | SEGMENT_ID | PACKET_TYPE    | CONTROL_PACKET_TYPE/APP_NAME | SOURCE_ID | DESTINATION_ID | TRANSMITTER_ID | RECEIVER_ID |
| 915  | 0         | N/A        | Control_Packet | ICMP_EchoRequest             | NODE-1    | ROUTER-3       | NODE-1         | ROUTER-3    |
| 916  | 0         | N/A        | Control_Packet | ICMP_EchoRequest             | NODE-2    | ROUTER-5       | NODE-2         | ROUTER-5    |
| 917  | 0         | N/A        | Control_Packet | ICMP_EchoReply               | ROUTER-3  | NODE-1         | ROUTER-3       | NODE-1      |
| 918  | 0         | N/A        | Control_Packet | ICMP_EchoReply               | ROUTER-5  | NODE-2         | ROUTER-5       | NODE-2      |
| 1822 | 0         | N/A        | Control_Packet | ICMP_EchoRequest             | NODE-1    | ROUTER-3       | NODE-1         | ROUTER-3    |
| 1823 | 0         | N/A        | Control_Packet | ICMP_EchoRequest             | NODE-2    | ROUTER-5       | NODE-2         | ROUTER-5    |
| 1824 | 0         | N/A        | Control_Packet | ICMP_EchoReply               | ROUTER-3  | NODE-1         | ROUTER-3       | NODE-1      |
| 1825 | 0         | N/A        | Control_Packet | ICMP_EchoReply               | ROUTER-5  | NODE-2         | ROUTER-5       | NODE-2      |
| 2726 | 0         | N/A        | Control_Packet | ICMP_EchoRequest             | NODE-1    | ROUTER-3       | NODE-1         | ROUTER-3    |
| 2727 | 0         | N/A        | Control_Packet | ICMP_EchoRequest             | NODE-2    | ROUTER-5       | NODE-2         | ROUTER-5    |
| 2728 | 0         | N/A        | Control_Packet | ICMP_EchoReply               | ROUTER-3  | NODE-1         | ROUTER-3       | NODE-1      |

- Open Wireshark and apply filter ICMP. we can see the ping request and reply packets in Wireshark

| No.  | Time      | Source   | Destination | Protocol | Length | Info                                                        |
|------|-----------|----------|-------------|----------|--------|-------------------------------------------------------------|
| 305  | 3.000000  | 11.1.1.2 | 11.1.1.1    | ICMP     | 28     | Echo (ping) request id=0x0000, seq=0/0, ttl=2 (reply in 3.. |
| 307  | 3.000018  | 11.1.1.1 | 11.1.1.2    | ICMP     | 28     | Echo (ping) reply id=0x0000, seq=0/0, ttl=255 (request ..   |
| 608  | 6.000000  | 11.1.1.2 | 11.1.1.1    | ICMP     | 28     | Echo (ping) request id=0x0000, seq=0/0, ttl=2 (reply in 6.. |
| 610  | 6.000018  | 11.1.1.1 | 11.1.1.2    | ICMP     | 28     | Echo (ping) reply id=0x0000, seq=0/0, ttl=255 (request ..   |
| 910  | 9.000000  | 11.1.1.2 | 11.1.1.1    | ICMP     | 28     | Echo (ping) request id=0x0000, seq=0/0, ttl=2 (reply in 9.. |
| 912  | 9.000018  | 11.1.1.1 | 11.1.1.2    | ICMP     | 28     | Echo (ping) reply id=0x0000, seq=0/0, ttl=255 (request ..   |
| 1034 | 10.201252 | 11.1.1.2 | 11.1.1.1    | ICMP     | 28     | Echo (ping) request id=0x0000, seq=0/0, ttl=255 (no respo.. |
| 1035 | 10.201319 | 11.3.1.2 | 11.1.1.2    | ICMP     | 28     | Echo (ping) reply id=0x0000, seq=0/0, ttl=253               |
| 1136 | 11.201252 | 11.1.1.2 | 11.1.1.1    | ICMP     | 28     | Echo (ping) request id=0x0000, seq=0/0, ttl=255 (no respo.. |
| 1137 | 11.201319 | 11.3.1.2 | 11.1.1.2    | ICMP     | 28     | Echo (ping) reply id=0x0000, seq=0/0, ttl=253               |
| 1216 | 12.000000 | 11.1.1.2 | 11.1.1.1    | ICMP     | 28     | Echo (ping) request id=0x0000, seq=0/0, ttl=2 (reply in 1.. |
| 1218 | 12.000018 | 11.1.1.1 | 11.1.1.2    | ICMP     | 28     | Echo (ping) reply id=0x0000, seq=0/0, ttl=255 (request ..   |
| 1240 | 12.201252 | 11.1.1.2 | 11.1.1.1    | ICMP     | 28     | Echo (ping) request id=0x0000, seq=0/0, ttl=255 (no respo.. |

### 3. Route Commands

- route print
- route delete
- route add

In order to view the entire contents of the IP routing table, use following commands **route print**

route print

```

C:\Program Files\NetSim Standard\bin\NetSimCLI.exe
NetSim>route print
=====
IP Route Table
=====

```

| Metric | Network | Destination Type | Netmask/Prefix                  | Gateway  | Interface |
|--------|---------|------------------|---------------------------------|----------|-----------|
| 200    |         | OSPF             | 11.2.1.2 255.255.0.0            | 11.2.1.2 | 11.2.1.1  |
| 200    |         | OSPF             | 11.3.1.1 255.255.0.0            | 11.2.1.2 | 11.2.1.1  |
| 200    |         | OSPF             | 11.3.1.2 255.255.0.0            | 11.5.1.2 | 11.5.1.1  |
| 200    |         | OSPF             | 11.5.1.2 255.255.0.0            | 11.5.1.2 | 11.5.1.1  |
| 300    |         | LOCAL            | 11.5.0.0 255.255.0.0            | on-link  | 11.5.1.1  |
| 300    |         | LOCAL            | 11.2.0.0 255.255.0.0            | on-link  | 11.2.1.1  |
| 300    |         | LOCAL            | 11.1.0.0 255.255.0.0            | on-link  | 11.1.1.1  |
| 306    |         | MULTICAST        | 224.0.0.1 255.255.255.255       | on-link  | 11.1.1.1  |
| 306    |         | MULTICAST        | 224.0.0.0 240.0.0.0             | on-link  | 11.1.1.1  |
| 999    |         | BROADCAST        | 255.255.255.255 255.255.255.255 | on-link  | 11.1.1.1  |

```

=====

```

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

```
route delete destination_network
```

- So, to delete the route with destination network 11.5.0.0, all we'd have to do is type this command

```
route delete 11.5.1.2
```

```

NetSim>route delete 11.5.1.2
OK!
NetSim>

```

- To check whether route has been deleted or not check again using **route print** command

- To add a static route to the table, you will type a command using the following syntax

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

- So, for example, if you wanted to add a route specifying that all traffic bound for the 11.5.1.2 subnet went to a gateway at 11.5.1.1

```
route ADD 11.5.1.2 MASK 255.255.0.0 11.5.1.1 METRIC 100 IF 2
```

- If you were to use the **route print** command to look at the table now, you would see your new static route

```

C:\Program Files\NetSim Pro\bin\NetSimCLI.exe
NetSim>route ADD 11.5.1.2 MASK 255.255.0.0 11.5.1.1 METRIC 100 IF 2
OK!
NetSim>route print
=====
IP Route Table
=====

```

| Type      | Network Destination | Netmask/Prefix  | Gateway  | Interface | Metric |
|-----------|---------------------|-----------------|----------|-----------|--------|
| STATIC    | 11.5.1.2            | 255.255.0.0     | 11.5.1.1 | 11.2.1.1  | 100    |
| OSPF      | 11.2.1.2            | 255.255.0.0     | 11.2.1.2 | 11.2.1.1  | 200    |
| OSPF      | 11.3.1.1            | 255.255.0.0     | 11.2.1.2 | 11.2.1.1  | 200    |
| OSPF      | 11.3.1.2            | 255.255.0.0     | 11.5.1.2 | 11.5.1.1  | 200    |
| OSPF      | 11.5.0.0            | 255.255.0.0     | on-link  | 11.5.1.1  | 300    |
| LOCAL     | 11.2.0.0            | 255.255.0.0     | on-link  | 11.2.1.1  | 300    |
| LOCAL     | 11.1.0.0            | 255.255.0.0     | on-link  | 11.1.1.1  | 300    |
| LOCAL     | 224.0.0.1           | 255.255.255.255 | on-link  | 11.1.1.1  | 306    |
| MULTICAST | 224.0.0.0           | 240.0.0.0       | on-link  | 11.1.1.1  | 306    |
| MULTICAST | 255.255.255.255     | 255.255.255.255 | on-link  | 11.1.1.1  | 999    |
| BROADCAST |                     |                 |          |           |        |

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

### 3. ACL Configuration:

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

#### Commands to configure ACL:

- To view ACL syntax use: **acl print**
- Before using ACL's we must first verify that acl option enabled. A common way to enable ACL use command: **ACL Enable**
- Enters configuration mode of ACL using: **aclconfig**
- To view ACL Table: **Print**
- To exit from ACL configuration use command : **exit**
- To disable ACL use command: **ACL Disable** (use this command after exit from acl configuration)

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

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

### Step to Configure ACL:

- To create a new rule in the ACL use command as shown below to block UDP packet in Interface\_2 and Interface\_3 of the Router\_3
- Disable TCP in all nodes(Wired Node and Router)
- Click on run simulation option and In the Run time Interaction tab set Interactive Simulation as True and click on Accept
- Set the Simulation Time as 500 sec or more. Click Ok
- Right click on Router\_3 and select NetSim Console. Use the command as follows:

```
NetSim>acl enable
ACL is enable
NetSim>aclconfig
ROUTER_3/ACLCONFIG>acl print
Usage: [PERMIT, DENY] [INBOUND, OUTBOUND, BOTH] PROTO SRC DEST
SPORT DPORT IFID
ROUTER_3/ACLCONFIG>DENY BOTH UDP ANY ANY 0 0 2
OK!
ROUTER_3/ACLCONFIG>DENY BOTH UDP ANY ANY 0 0 3
OK!
ROUTER_3/ACLCONFIG>print
DENY BOTH UDP ANY/0 ANY/0 0 0 2
ROUTER_3/ACLCONFIG>exit
```

```
NetSim>acl disable
```

```
ACL is disable
```

```
NetSim>
```

```
C:\Program Files\NetSim Pro\bin\NetSimCLI.exe
Initialising Winsock...Initialised.
Connecting to device DESKTOP-LC53CTS.
Connection attempt: 1
Connection established.

NetSim>acl enable
ACL is enable

NetSim>aclconfig
ROUTER_3/ACLCONFIG>acl print
Usage: [PERMIT,DENY] [INBOUND,OUTBOUND,BOTH] PROTO SRC DEST SPORT DPORT IFID
ROUTER_3/ACLCONFIG>DENY BOTH UDP ANY ANY 0 0 2
OK!
ROUTER_3/ACLCONFIG>print
DENY BOTH UDP ANY/0 ANY/0 0 0 2
ROUTER_3/ACLCONFIG>exit

NetSim>acl disable
ACL is disable

NetSim>
```

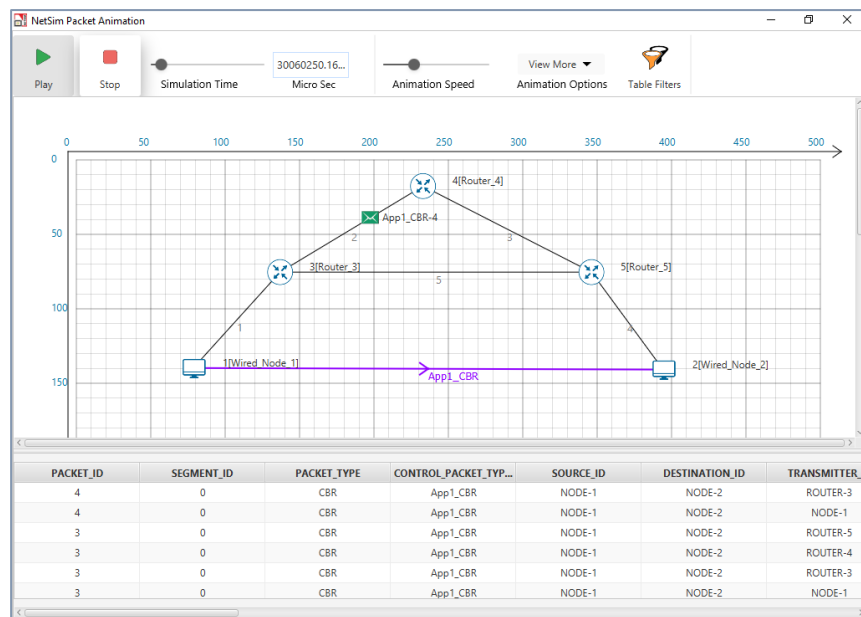
## ACL Results:

The impact of ACL rule applied over the simulation traffic can be observed in the IP\_Metrics\_Table in the simulation results window, In Router\_3 no of packets blocked by firewall has been shown below

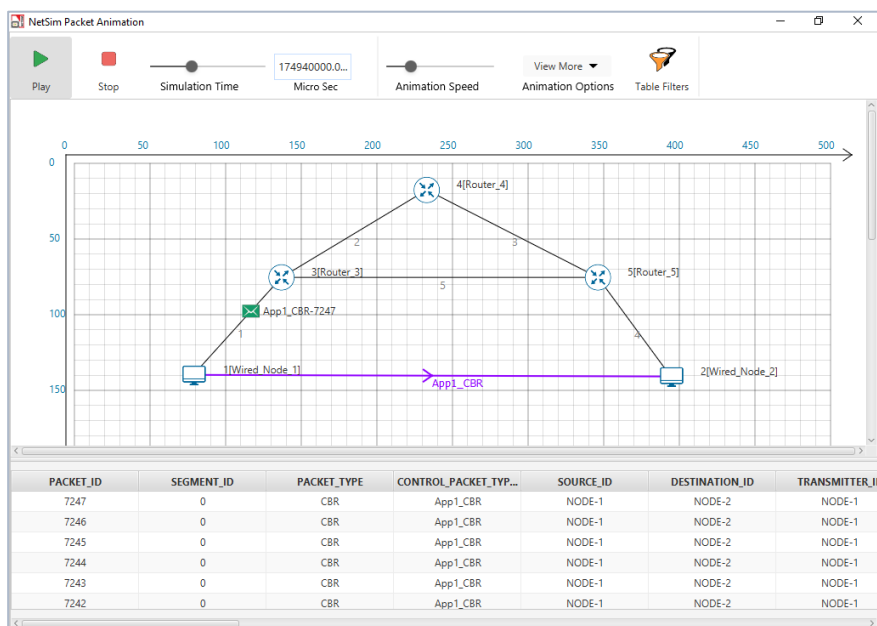
**Note: Results will vary based on time of ACL command are executed**

| IP_Metrics_Table                                             |             |                  |                 |                  |             |                  |  |
|--------------------------------------------------------------|-------------|------------------|-----------------|------------------|-------------|------------------|--|
| IP_Metrics <input checked="" type="checkbox"/> Detailed View |             |                  |                 |                  |             |                  |  |
| Device Id                                                    | Packet sent | Packet forwarded | Packet received | Packet discarded | TTL expired | Firewall blocked |  |
| 1                                                            | 11090       | 0                | 0               | 0                | 0           | 0                |  |
| 2                                                            | 83          | 0                | 2911            | 0                | 0           | 0                |  |
| 3                                                            | 3066        | 10995            | 64              | 0                | 0           | 8079             |  |
| 4                                                            | 2981        | 2915             | 66              | 0                | 0           | 0                |  |
| 5                                                            | 3061        | 2915             | 66              | 0                | 0           | 0                |  |

- Check Packet animation window whether packets has been blocked in Router\_3 or not after entering ACL command to deny UDP traffic
- Before applying ACL rule there is packet flow from Wired\_Node\_1 to Wired\_Node\_2



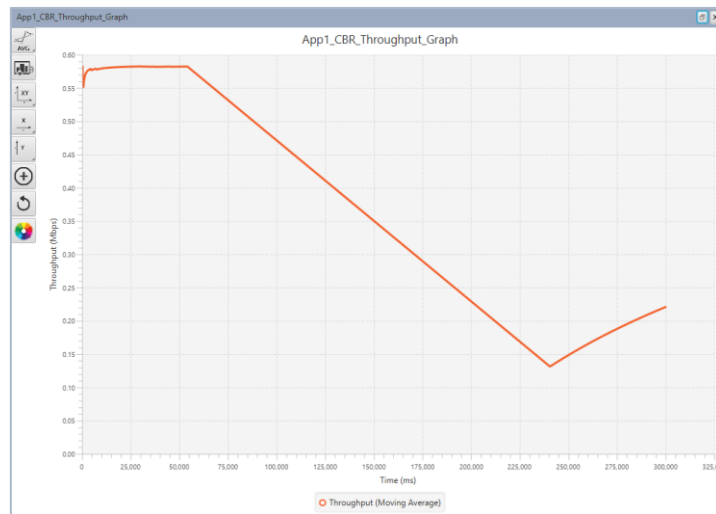
- After applying ACL rule Packet flows up to Router\_3 only



The impact of ACL rule applied over the simulation traffic can be observed in the Application throughput plot. Throughput graph will show a drop after ACL is set. If ACL is disabled after a while, application packets will start flowing across the router. The Application throughput plot will show a drop and increase(Moving throughput graph) in throughput after setting ACL and disabling ACL respectively.

**Following is an example:**

ACL rule applied at around 50sec user can see the drop in throughput in the graph, since router blocks UDP packets in the plot. Once ACL has been disabled at around 240sec router permits packets and hence throughput can be observed in the plot shown below



## 5 Reference Documents

1. IEEE 802.3 standard for Ethernet
2. IEEE 802.11 standard 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)

## 6 Latest FAQs

Up to date FAQs on NetSim's Internetworks library is 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>