NetSim™

Simulation Platform for Network R & D

Experiments Manual

TETCOS

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

This manual is for informational purposes only. TETCOS MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Warning! DO NOT COPY

Contact us at

TETCOS

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

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

E-Mail: sales@tetcos.com

Visit: www.tetcos.com

LIST OF EXPERIMENTS

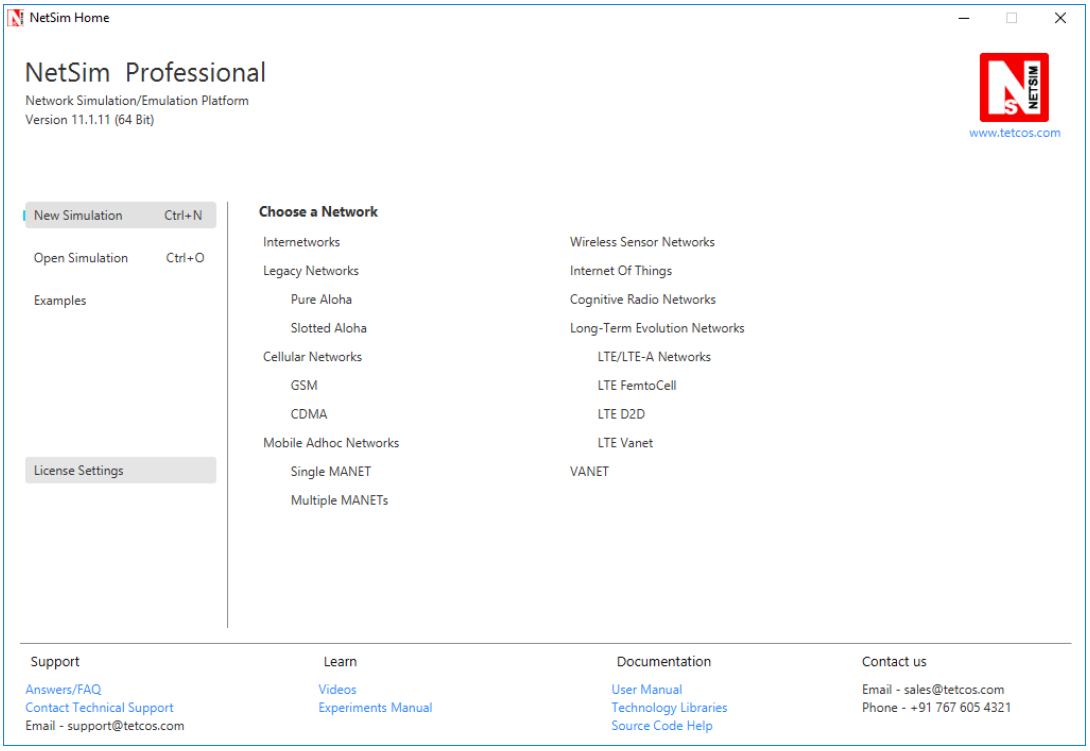
1. Introduction to NetSim

1.1 Introduction to network simulation with NetSim, NetSim feature list and NetSim Simulation environment

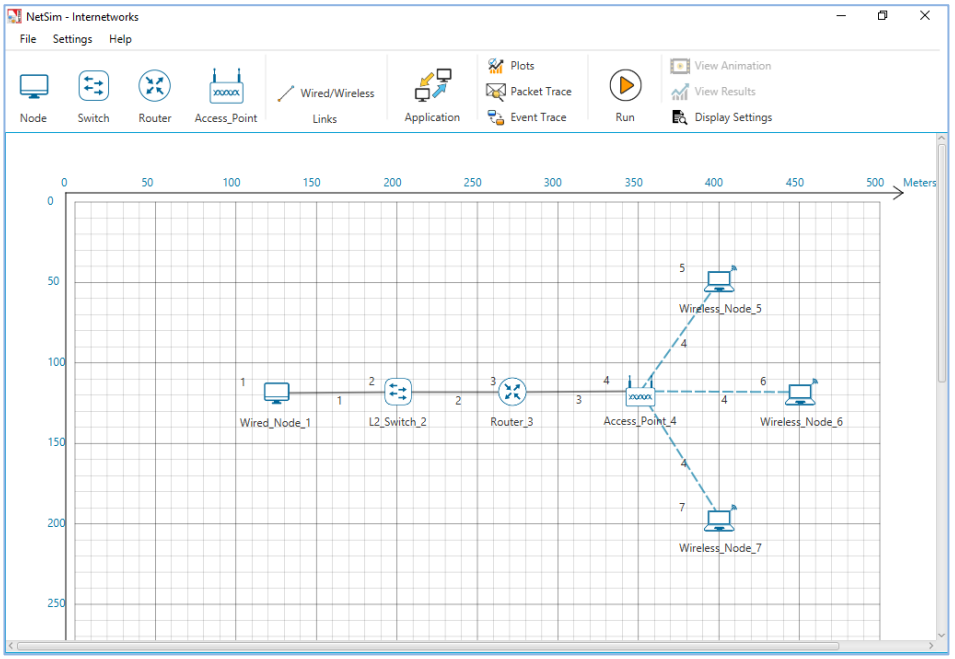
NetSim is a network simulation tool that allows you to create network scenarios, model traffic, design protocols and analyze network performance. Users can study the behavior of a network by test combinations of network parameters. The various network technologies covered in NetSim include:

- Internetworks - Ethernet, WLAN, IP, TCP
- Legacy Networks - Aloha, Slotted Aloha
- Cellular Networks - GSM, CDMA
- Mobile Adhoc Networks - DSR, AODV, OLSR, ZRP
- Wireless Sensor Networks - 802.15.4
- Internet of Things - 6LoWPAN gateway, 802.15.4 MAC / PHY, RPL
- Cognitive Radio Networks - 802.22
- Long-Term Evolution Networks - LTE, LTE Adv.
- Software Defined Networking
- Advanced Routing and Switching - VLAN, IGMP, PIM, L3 Switch, ACL and NAT

NetSim home screen will appear as shown below:



- **Network Design Window:** NetSim network design window enables users to model a network comprising of network devices switches, routers, nodes, etc., connect them through links and model application traffic to flow in the network. The network devices shown are specific to the network/technology chosen by the user.



Description:

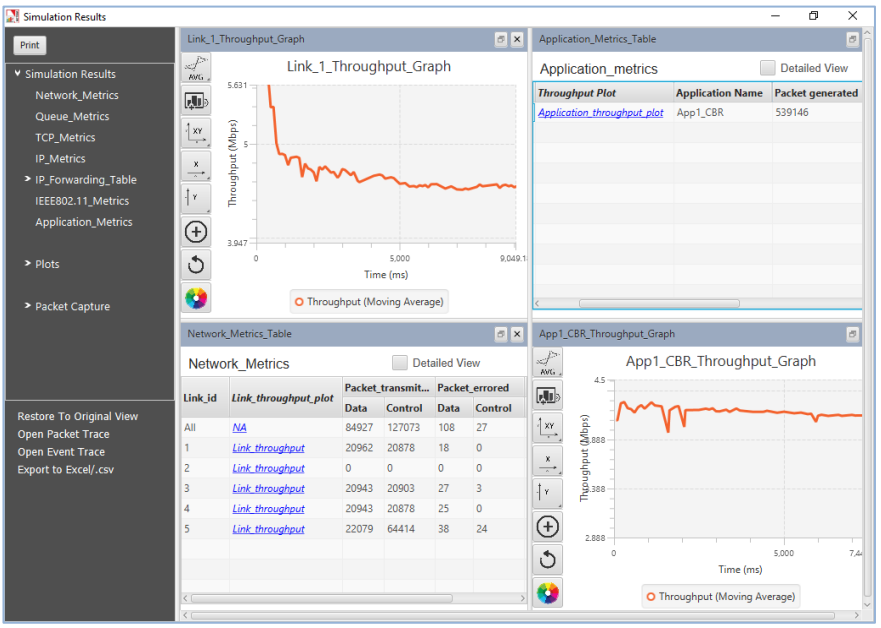
1. Click on File to Save or Save As or Close the simulation in the current workspace.
2. Go to Settings → Environmental Settings and choose the type of environment. Here we have chosen the Environment in the form of a Grid.
3. Under Help users can find various resources like manuals, source code help, technology libraries, tutorials and other options like raise a ticket etc.

Below the menu options, the entire region constitutes the Ribbon from which users can do the following:

- Click and drop network devices and right click to edit properties
- Click on Wired/Wireless links to connect the devices to one another. It automatically detects whether to use a Wired/Wireless link based on the devices we are trying to connect
- Click on Application to configure different types of applications and generate traffic
- Click on Plots, Packet Trace, and Event Trace and click on the enable check box option which appears in their respective windows to generate additional metrics to further analyze the network performance.
- Click on Run to perform the simulation and specify the simulation time in seconds.

- Next to Run, we have View Animation and View Results options. Both the options remain hidden before we run the simulation or if the respective windows are already open.
- Display Settings option is mainly used to display various parameters like Device Name, IP, etc., to provide a better understanding especially during the design and animation.

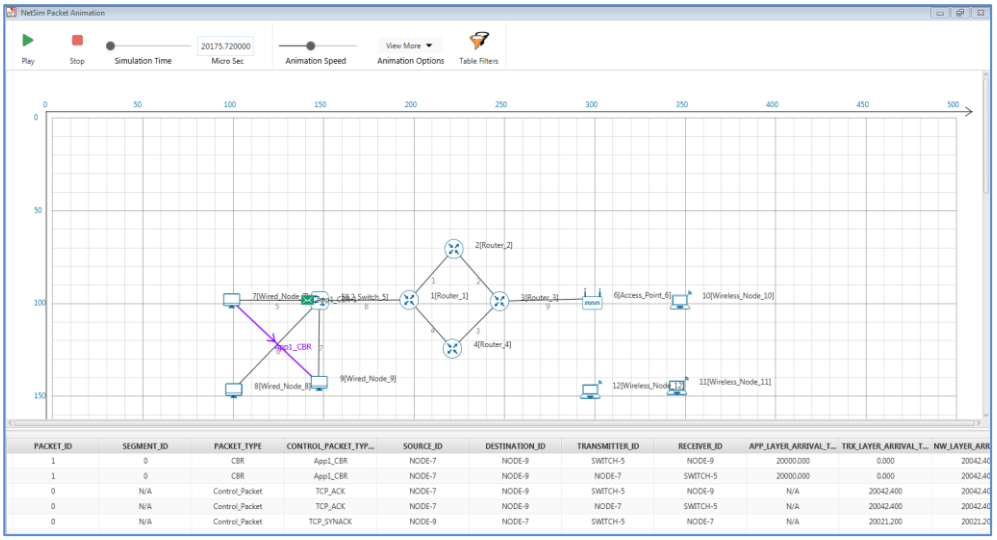
- **Results Window:** Upon completion of simulation, Network statistics or network performance metrics reported in the form of graphs and tables. The report includes metrics like throughput, simulation time, packets generated, packets dropped, collision counts etc.



Description:

1. Below Simulation Results, Clicking on a particular metrics will display the respective metrics window.
2. Clicking on links in a particular metrics will display the plot in a separate window
3. Enabling Detailed View by clicking on it will display the remaining properties
4. Clicking on Restore to Original View will get back to the original view
5. Click on Open Packet Trace / Open Event Trace to open the additional metrics which provide in depth analysis on each Packets / Events.

- **Packet Animation Window:** When we click on run simulation, we have the option to record / play & record animation. If this is enabled, users can view the animation during the run time or upon completion of the simulation users can see the flow of packets through the network. Along with this, more than 25+ fields of packet information is available as a table at the bottom. This table contains all the fields recorded in the packet trace. In addition, animation options are available for viewing different graphs, IP Addresses, Node movement etc.



Description:

1. Click on Play to view the animation. You can Pause the animation at any interval and Play again.
2. Click on Stop to stop the animation. Now click on Play to start the animation from the beginning.
3. Next to that we also have speed controllers to increase/decrease Simulation Time and Animation Speed
4. View More option enables the user to view Plots, Throughputs, and IP Tables during the animation
5. Table Filters are used to filter the packet information's shown in the below table during simulation as per user requirement
6. While setting more than one application, it is differentiated using different color indications
7. Packets are indicated using different color combinations say, blue color indicates control packets, green color indicates data packets and red color indicates error packets.

1.2 Typical sequence of steps to do experiments in this manual

The typical steps involved in doing experiments in NetSim are,

- **Network Set up:** Drag and drop devices, and connect them using wired or wireless links
- **Configure Properties:** Configure device, protocol or link properties by right clicking on the device or link and modifying parameters in the properties window.
- **Model Traffic:** Click on the Application icon present on the ribbon and set traffic flows.

- **Enable Trace/Plots (optional):** Click on packet trace, event trace and Plots to enable. Packet trace logs packet flow, event trace logs each event (NetSim is a discrete event simulator) and the Plots button enables charting of various throughputs over time.
- **Save/Save As/Open/Edit:** Click on File → Save / File → Save As to save the experiments in the current workspace. Saved experiments can then opened from NetSim home screen to run the simulation or to modify the parameters and again run the simulation.
- **View Animation/View Results:** Visualize through the animator to understand working and to analyze results and draw inferences.

Note: Example Configuration files for all experiments would available where NetSim has been installed. This directory is (<NetSim_Install_Directory>\Docs\Sample_Configuration\NetSim_Experiment_Manual)

2. Understand working of ARP, and IP forwarding within a LAN and across a router

2.1 Theory

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

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

2.2 ARP protocol Description

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

2.3 ARP Frame Format

<table>
<tr><td colspan="2">Hardware Type</td><td>Protocol Type</td></tr>
<tr><td>Hardware Address Length</td><td>Protocol address length</td><td>Opcode</td></tr>
<tr><td colspan="3">Sender Hardware Address</td></tr>
<tr><td colspan="2">Sender Protocol Address(1-2)</td><td>Sender Protocol Address(3-4)</td></tr>
<tr><td colspan="3">Target hardware Address</td></tr>
<tr><td colspan="3">Target Protocol Address</td></tr>
</table>

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

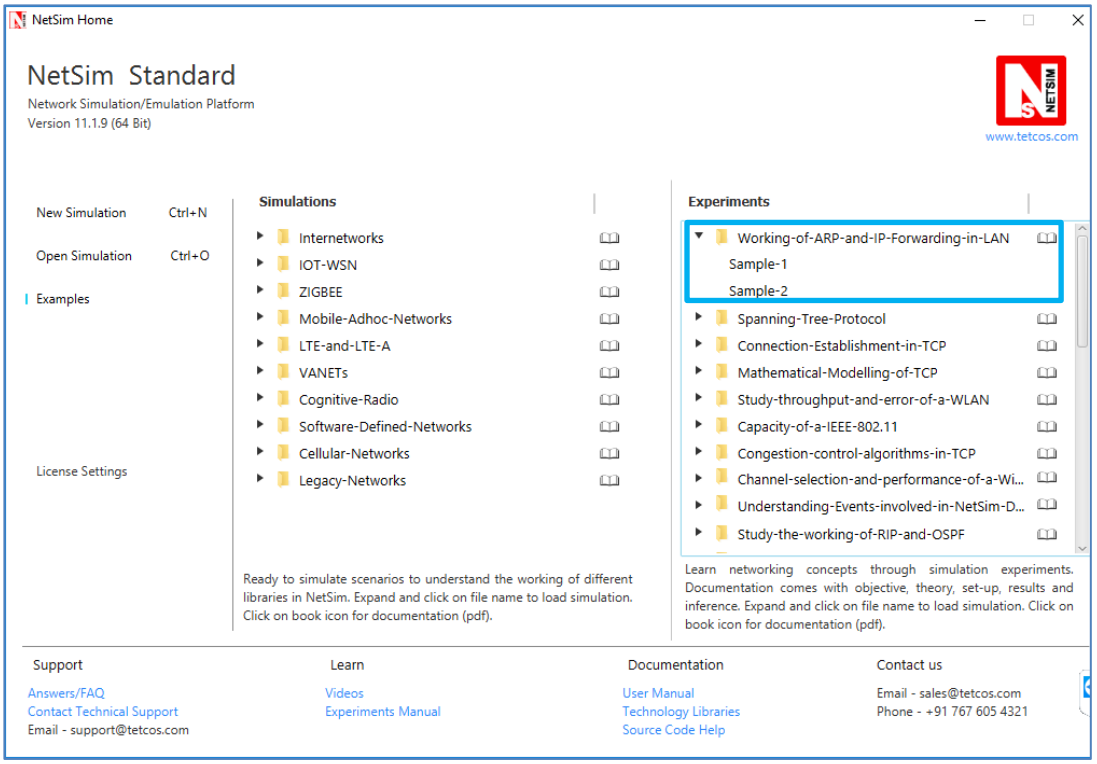
2.4 IP Forwarding Description

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

2.5 Network Set up

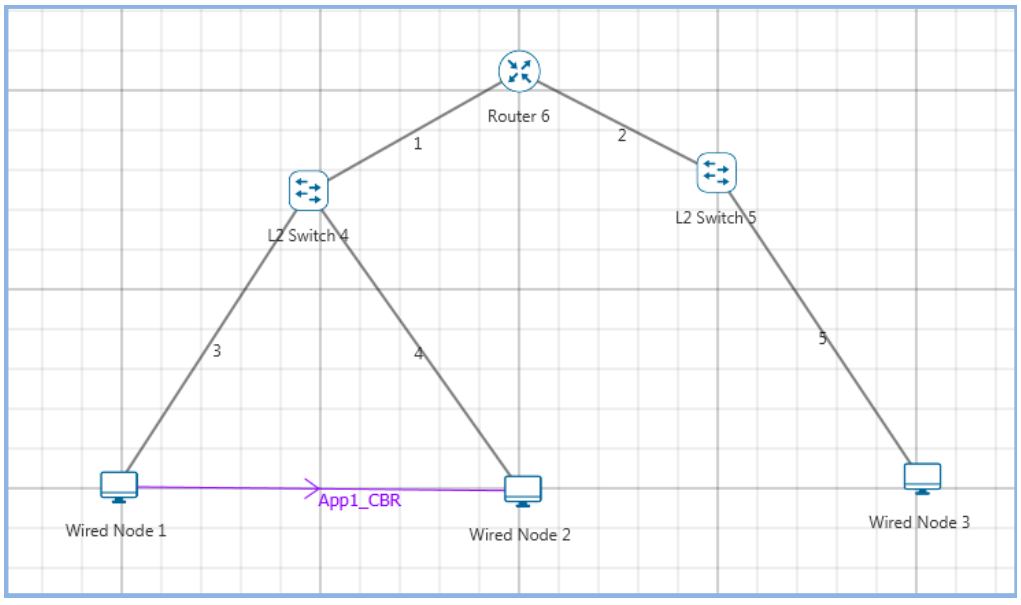
Step 1:

Open Examples →

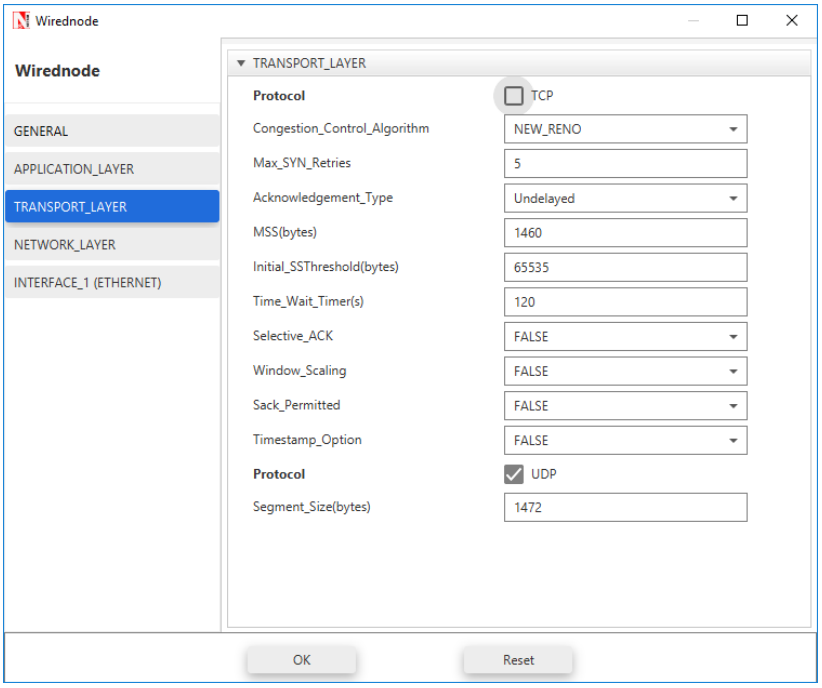


Step 2:

Click & drop Wired Nodes, Switches and Router onto the Simulation Environment as shown below and connect wired/wireless links between devices.



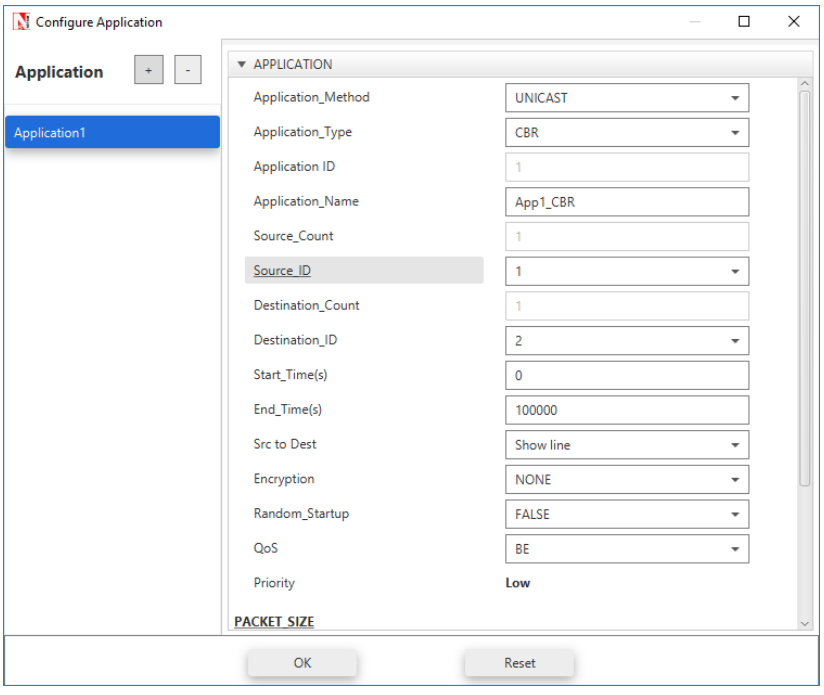
Node properties: Disable TCP in all nodes in Transport layer as shown below. This is because if TCP is enabled, the ARP table will get updated due to the transmission of TCP control packets thereby eliminating the need for ARP to resolve addresses.



Step 3: Create the Sample as follows:

2.6 Sample 1:

To run the simulation, click on Application icon present in the ribbon and set the Source_Id and Destination_Id as 1 and 2 respectively.

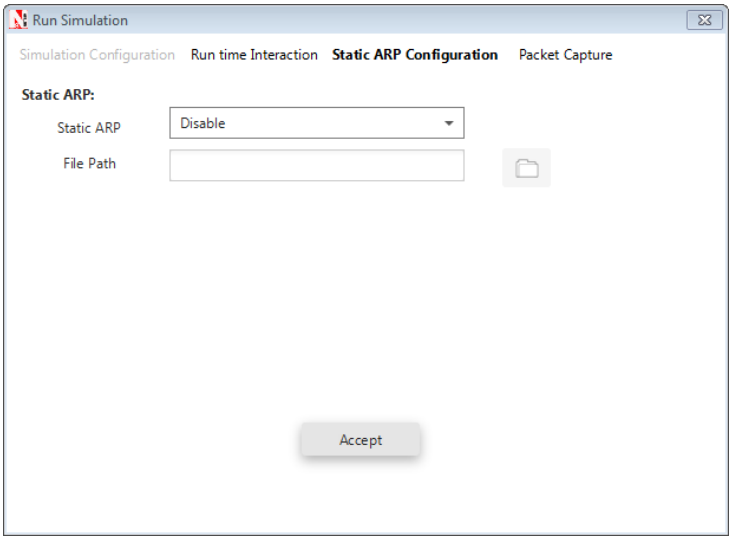


Enabling the packet trace:

- Click Packet Trace icon in the ribbon. This is used to log the packet details.
- Click on Enable Packet Trace check box. By default all attributes are selected.
- Click on OK button. Once the simulation is completed, the file gets stored in the location specified.

Step 4:

Click Run Simulation. Do not click on OK. Go to the Static ARP Configuration tab and set Static ARP as Disable and click Accept. If Static ARP is enabled then NetSim automatically creates the ARP table for each node. To see the working of the ARP protocol users should disable Static ARP. When disabled ARP request would be sent to the destination to find out the destinations MAC Address. Set simulation time as 10 seconds and click on OK to simulate.



2.7 Output – I:

After simulation, open Packet Trace from Simulation Results window.

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

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

2.8 Inference:

Intra-LAN-IP-forwarding:

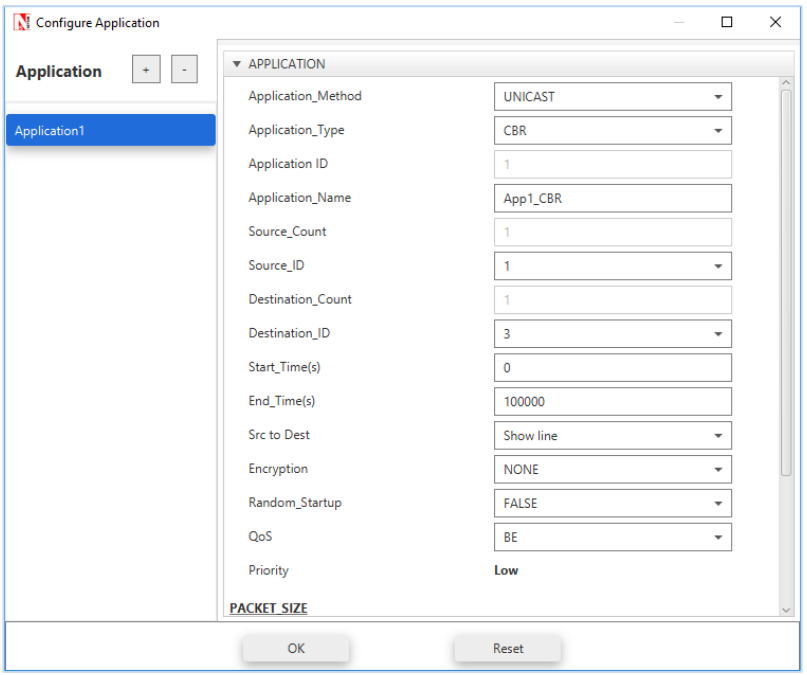
ARP PROTOCOL- WORKING

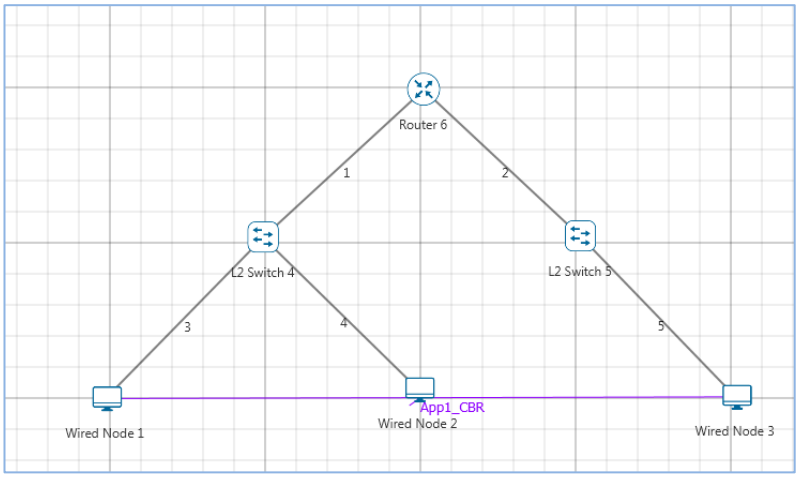
A Brief Explanation: NODE-1 broadcasts ARP_Request which is then broadcasted by SWITCH-4. NODE -2 sends the ARP_Reply to NODE-1 via SWITCH-4. After this step, datagrams are transmitted from NODE-1 to NODE-2. Notice the DESTINATION_ID column for ARP_Request type packets.

Step 5: Follow all the steps till Step 2 and perform the following sample:

2.9 Sample 2:

Reconfigure the application with SOURCE_ID as 1 and DESTINATION_ID as 3



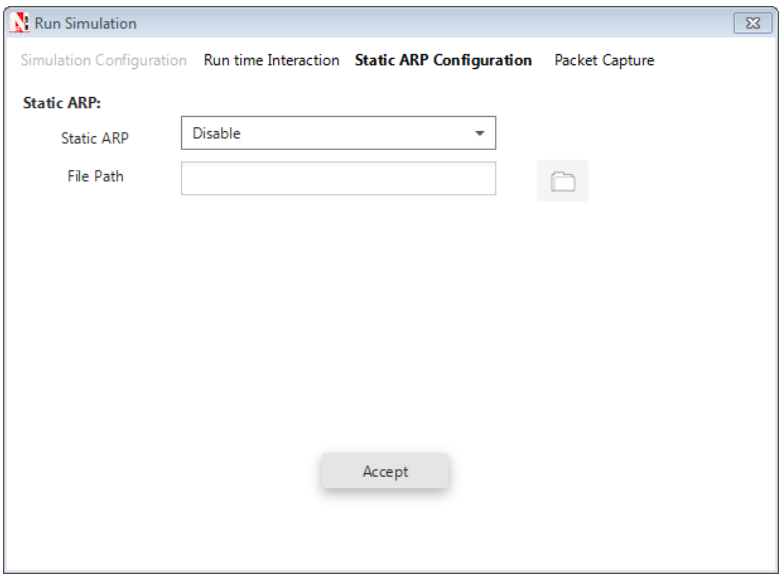


Enabling the packet trace:

- Click Packet Trace icon in the ribbon. This is used to log the packet details.
- Click on Enable Packet Trace check box. By default all attributes are selected.
- And Click on OK button. Once the simulation is completed, the file gets stored in the location specified.

Step 6:

Click Run Simulation. Do not click on OK. Go to **Static ARP Configuration** tab and set Static ARP as Disable and Click Accept.



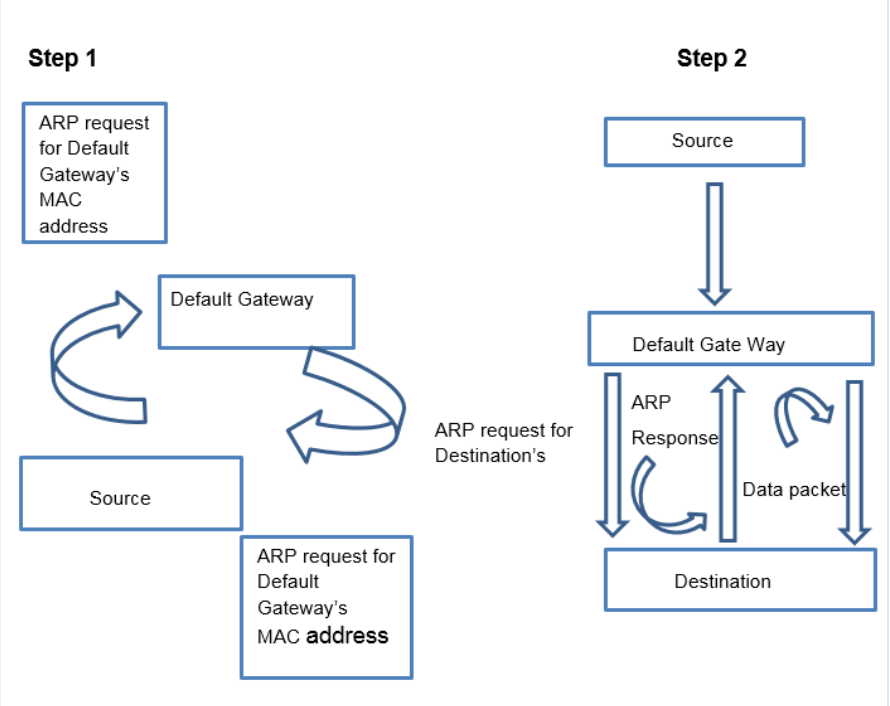
Set simulation time as 10 seconds and click on OK to simulate.

2.10 Output – II:

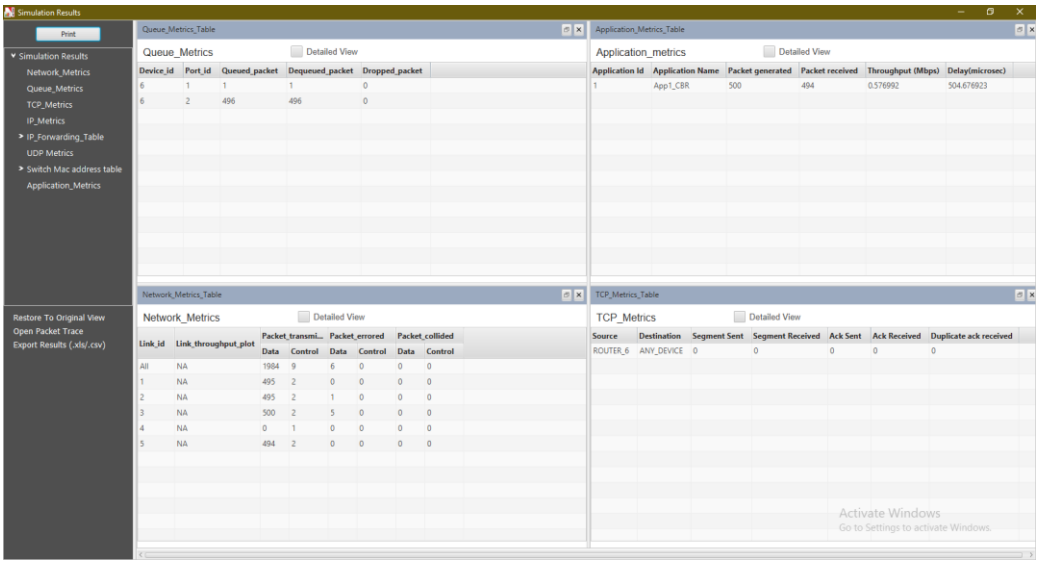
PACKET TRACE Analysis

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACK	SOURCE_ID	DESTINATION_I	TRANSMITTER_I	RECEIVER_ID
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	NODE-1	SWITCH-4
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	SWITCH-4	ROUTER-6
0	N/A	Control_Packet	ARP_Request	NODE-1	Broadcast-0	SWITCH-4	NODE-2
0	N/A	Control_Packet	ARP_Reply	ROUTER-6	NODE-1	ROUTER-6	SWITCH-4
0	N/A	Control_Packet	ARP_Reply	ROUTER-6	NODE-1	SWITCH-4	NODE-1
1	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	SWITCH-4
1	0	CBR	App1_CBR	NODE-1	NODE-3	SWITCH-4	ROUTER-6
0	N/A	Control_Packet	ARP_Request	ROUTER-6	Broadcast-0	ROUTER-6	SWITCH-5
0	N/A	Control_Packet	ARP_Request	ROUTER-6	Broadcast-0	SWITCH-5	NODE-3
0	N/A	Control_Packet	ARP_Reply	NODE-3	ROUTER-6	NODE-3	SWITCH-5
0	N/A	Control_Packet	ARP_Reply	NODE-3	ROUTER-6	SWITCH-5	ROUTER-6
1	0	CBR	App1_CBR	NODE-1	NODE-3	ROUTER-6	SWITCH-5
1	0	CBR	App1_CBR	NODE-1	NODE-3	SWITCH-5	NODE-3
2	0	CBR	App1_CBR	NODE-1	NODE-3	NODE-1	SWITCH-4

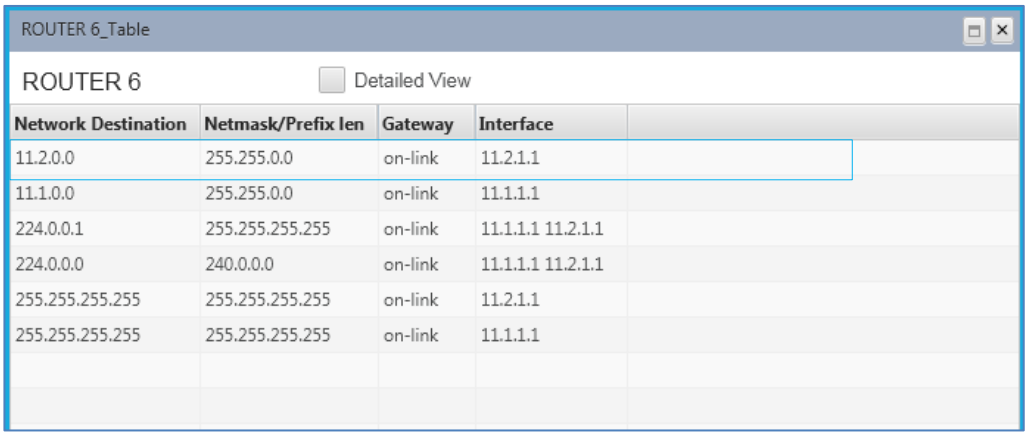
Across-Router-IP-forwarding



The IP forwarding table formed in the router can be accessed from the IP_Forwarding_Table list present in the Simulation Results window as shown below:



Router forwards packets intended to the subnet 11.2.0.0 to the interface with the IP 11.2.1.1 based on the first entry in its routing table.



2.11 Inference:

NODE-1 transmits ARP_Request which is further broadcasted by SWITCH-4. ROUTER-6 sends ARP_Reply to NODE-1 which goes through SWITCH-4. Then NODE-1 starts sending datagrams to NODE-3. If router has the MAC address of NODE-3 in its ARP table, then ARP ends here and router starts forwarding the datagrams to NODE-3 by consulting its forwarding table. In the other case, Router sends ARP_Request to appropriate subnet and after getting the MAC address of NODE-3, it then forwards the datagrams to NODE-3 using its forwarding table.

3. Simulate and study the spanning tree protocol

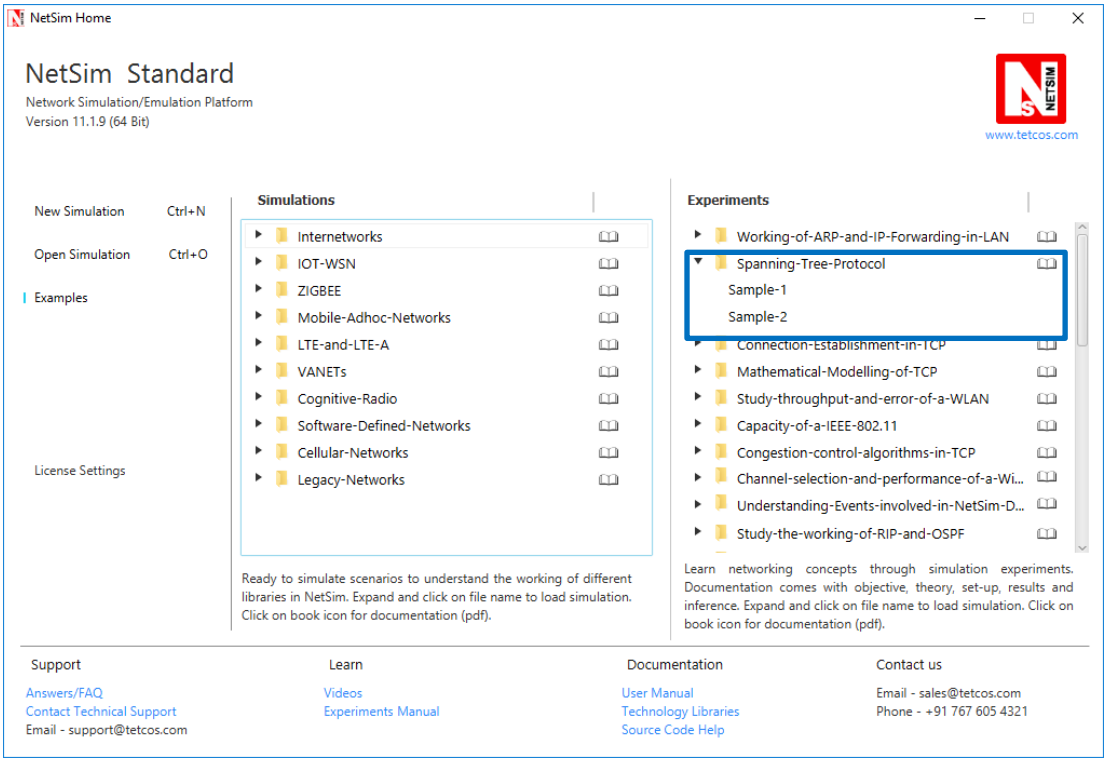
3.1 Introduction

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

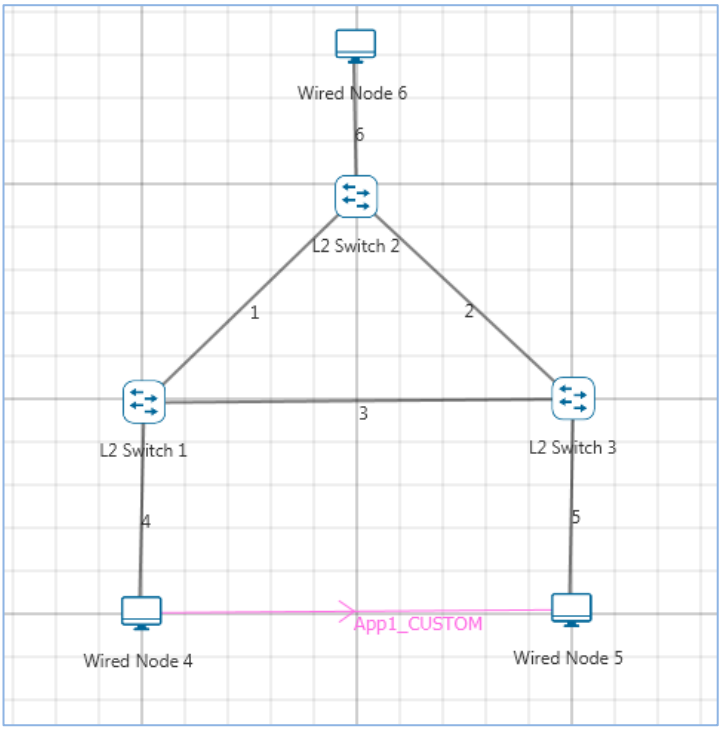
(***Reference:*** *A good reference for this topic is Section 3.1.4: Bridges and LAN switches, of the book, Computer Networks, 5th Edition by Peterson and Davie)*

3.2 Network Set up

Open Examples → Spanning-Tree-Protocol as shown below:



Create a network scenario as shown below,



(Note: At least three L2_Switches are required in the network to analyze spanning tree formation)

Sample Inputs: Inputs for the Sample experiments are given below,

Sample 1: Application properties:

Application Properties		
Traffic Type	**Custom**	
Source_Id	**4**	
Destination_Id	**5**	
Packet Size	**Distribution**	**Constant**
	Packet Size (bytes)	**1460**
Packet Inter Arrival Time	**Distribution**	**Constant**
	Packet Inter Arrival Time (µs)	**20000**

Wired Node 4 is sending data to Wired Node 5. The node properties are default.

(Note: Wired Node 6 is not generating Traffic to any other Wired Nodes)

L2_Switch Priority: L2_Switch priority is interpreted as weights associated with each interface of a L2_switch. A higher value indicates a higher priority.

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

(Note: Switch Priority has to be changed for all the interfaces of L2_Switch.)

Simulation Time - 10 Seconds

(Note: The Simulation Time can be selected only after doing the following two tasks,

- ***Set the properties of Nodes and L2_Switches***
- ***Then click on Run Simulation button).***

Sample 2: Set all properties as above and change properties of Switch as follows:

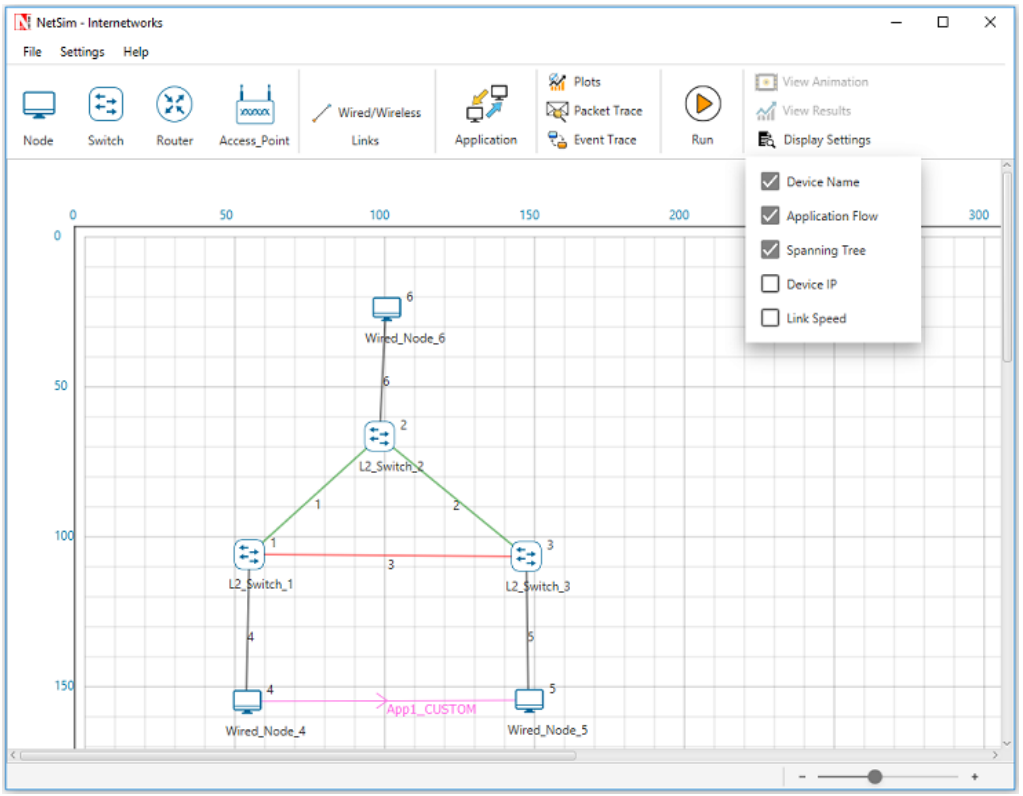
L2_Switch Properties	L2_Switch 1	L2_Switch 2	L2_Switch 3
Switch Priority	**1**	**2**	**3**

Simulation Time - 10 Seconds

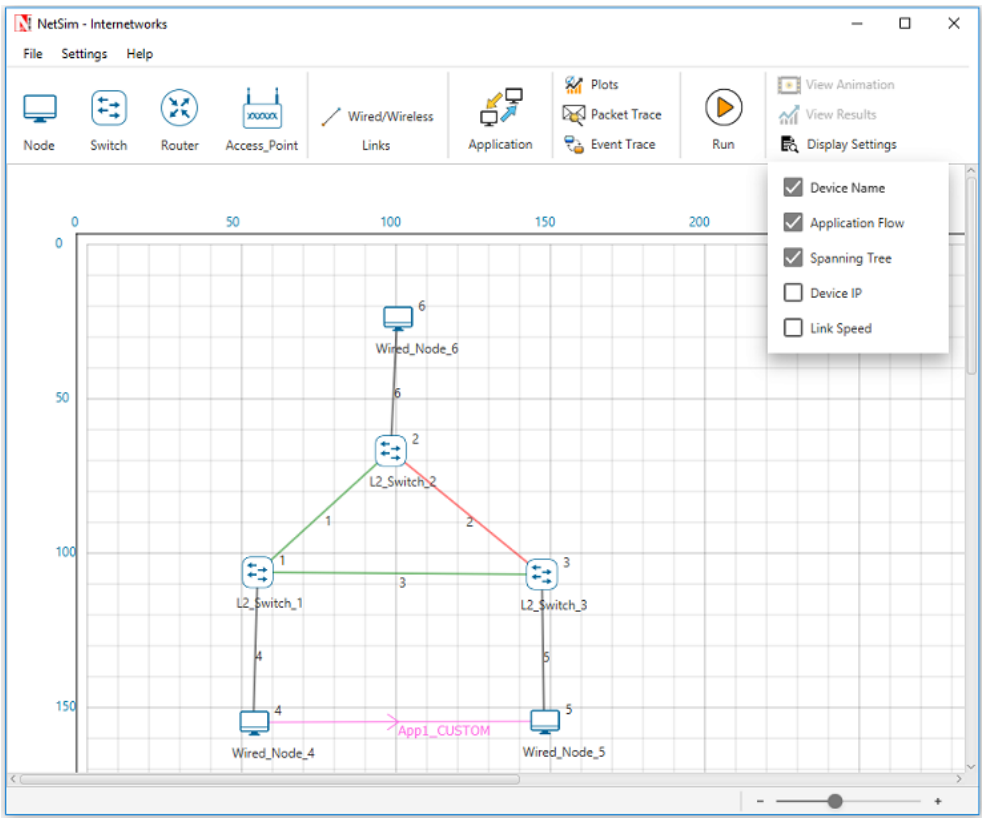
3.3 Output:

To view the output, click on Display Settings→Spanning Tree.

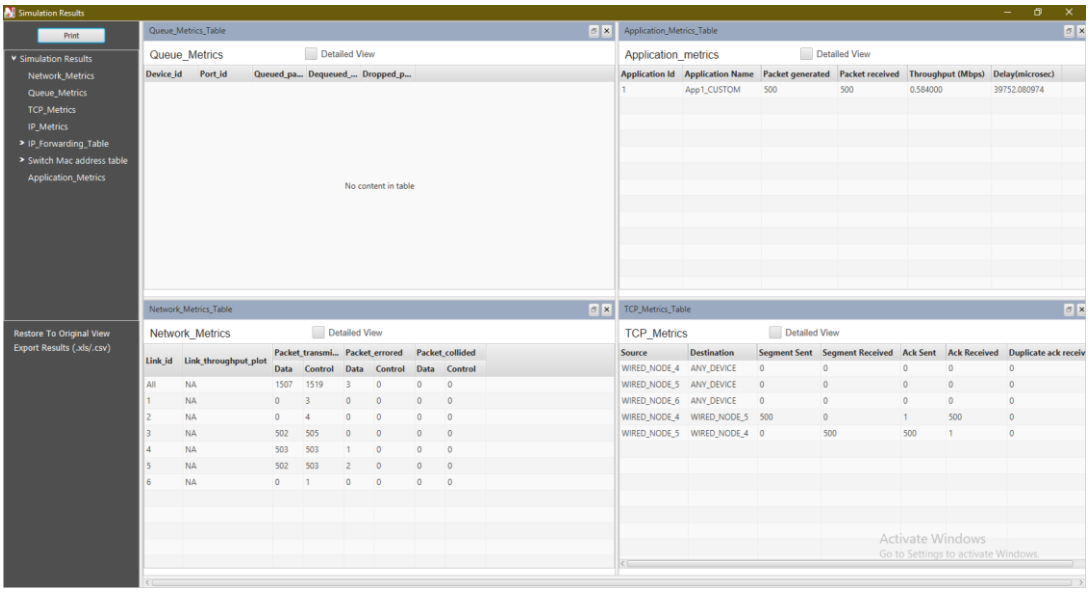
Sample 1:

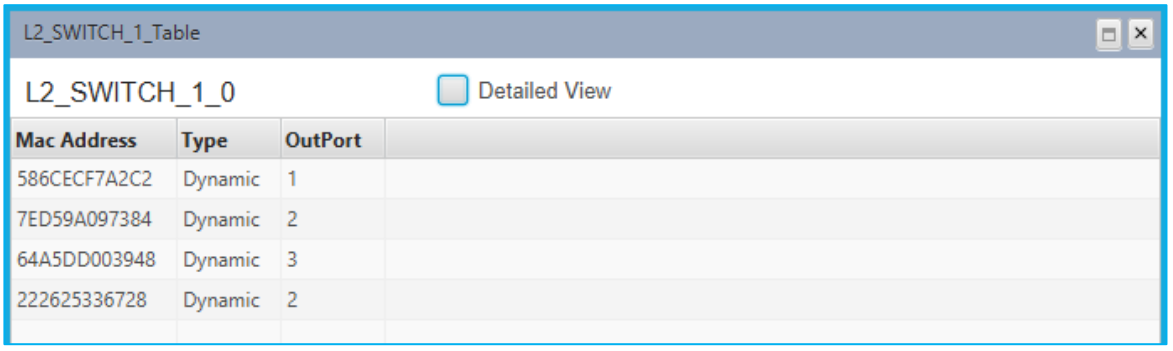


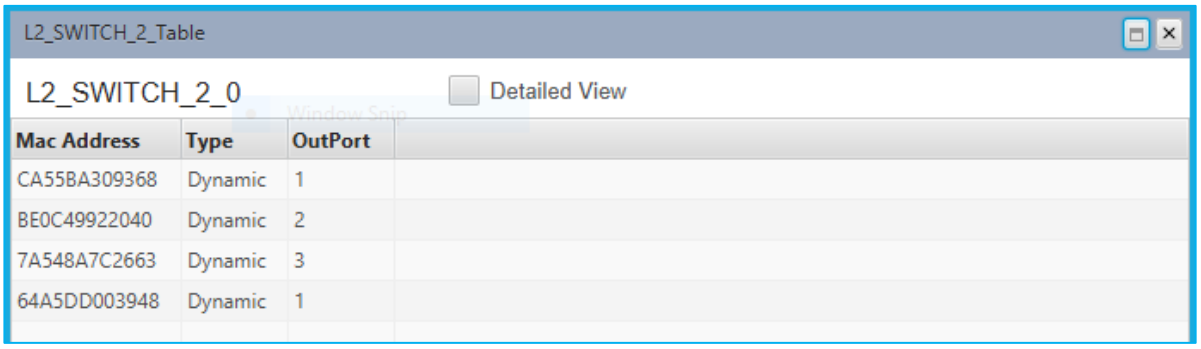
Sample 2:

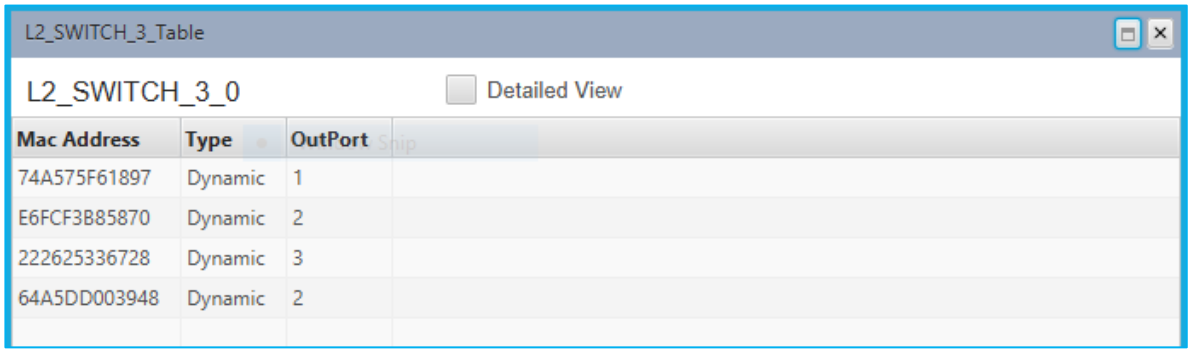


A switch table containing the MAC address entries, the port that is used for reaching it, along with the type of entry can also be obtained at the end of Simulation. The Switch table formed in each L2_Switch can be obtained from the Switch MAC address table list provided in the results window:









3.4 Inference:

Each L2_Switch has an ID which is a combination of its Lowest MAC address and priority. The Spanning tree algorithm selects the L2_Switch with the smallest ID as the root node of the Spanning Tree. The root node forwards frames out over all of its ports. In the other L2_switches the ports that have the least cost of reaching the root Switch are set as forward ports and the remaining are set as blocked ports. In the Sample 1, L2_Switch 2 was assigned least priority and was selected as a Root Switch. The green line indicates the forward path and the red line indicates the blocked path. The frame from Wired Node 4 should take the path through the L2_Switch 1, 2 and 3 to reach the Wired Node 5. In the Sample 2, L2_Switch 1 was assigned least priority and selected as a Root switch. In this case, the frame from Wired Node 4 takes the path through the L2_Switch 1 and 3 to reach the destination Wired Node 5.

4. Understand the working of "Connection Establishment" in TCP

4.1 Introduction

When two processes wish to communicate, their TCP's must first establish a connection i.e. initialize the status information on each side. Since connections must be established between unreliable hosts and over the unreliable internet communication system, a "three-way handshake" with clock based sequence numbers is the procedure used to establish a Connection. This procedure normally is initiated by one TCP and responded by another TCP. The procedure also works if two TCPs simultaneously initiate the procedure. When simultaneous attempt occurs, each TCP receives a "SYN" segment which carries no acknowledgement after it has sent a "SYN".

The simplest three-way handshake is shown in the following figure.

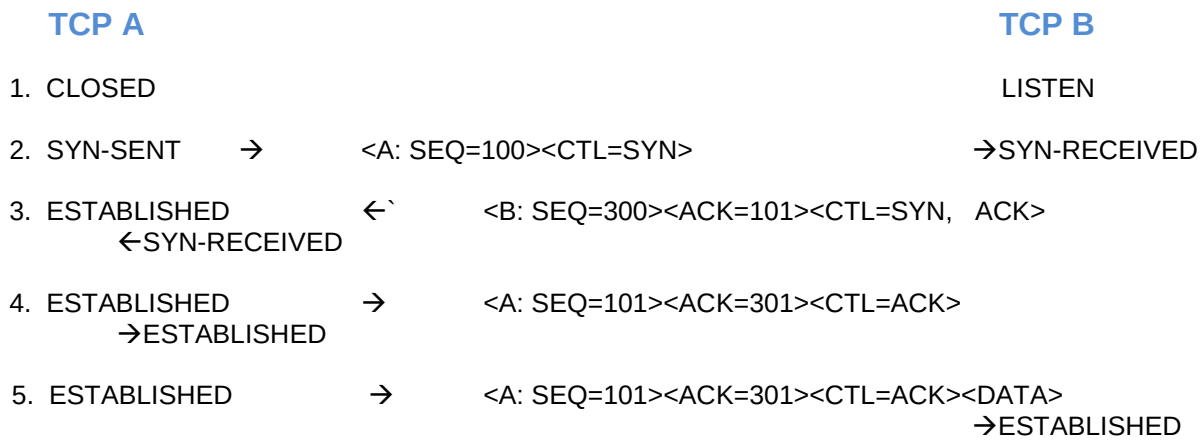


Fig: Basic 3-Way Handshake for Connection Synchronization

Explanation:

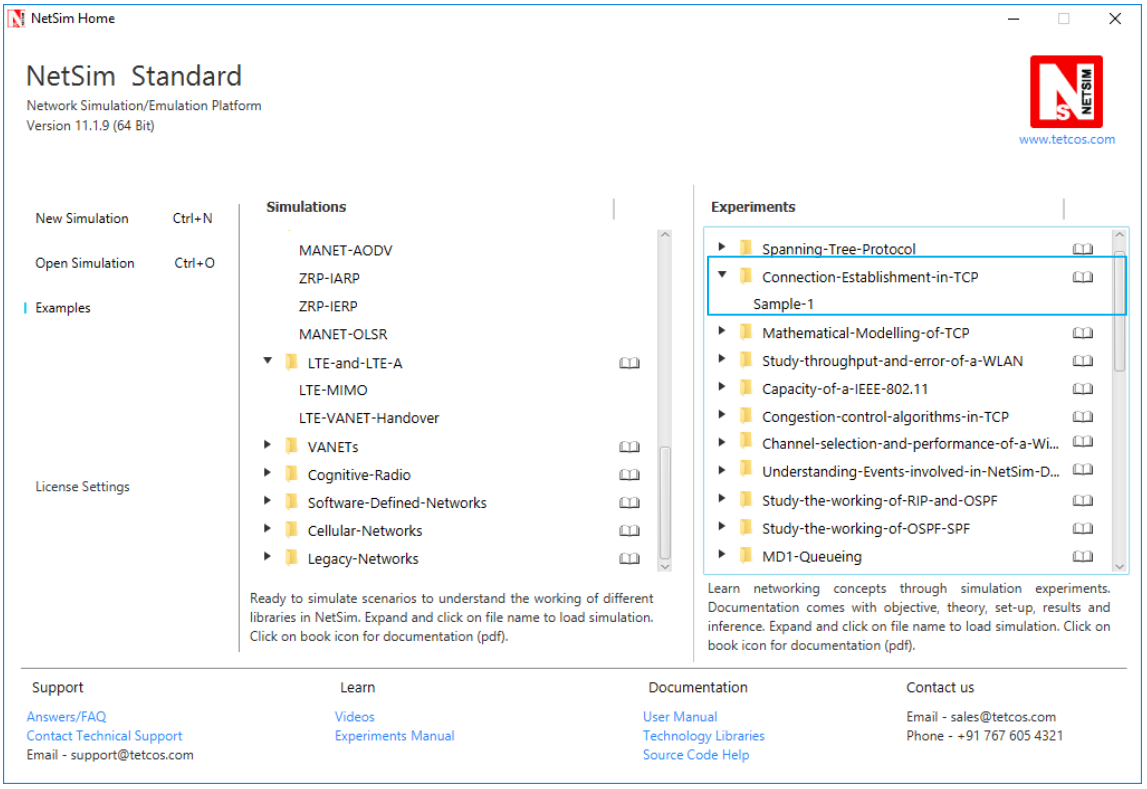
The above figure should be interpreted in the following way. Each line is numbered for reference purposes. Right arrows (→) indicates the departure of a TCP Segment from TCP A to TCP B, or arrival of a segment at B from A. Left arrows (←) indicates the reverse. TCP states represent the state after the departure or arrival of the segment (whose contents are shown in the center of each line). Segment contents are shown in abbreviated form, with sequence number, control flags, and ACK field. In line2 of the above figure, TCP A begins by sending a SYN segment indicating that it will use sequence numbers starting with sequence number 100. In line 3, TCP B sends a SYN and acknowledges the SYN it received from TCP A.

Note that the acknowledgment field indicates TCP B is now expecting to hear sequence 101, acknowledging the SYN which occupied sequence 100. At line 4, TCP A responds with an empty segment containing an ACK for TCP B's SYN; and in line 5, TCP A sends some data.

4.2 Network Setup

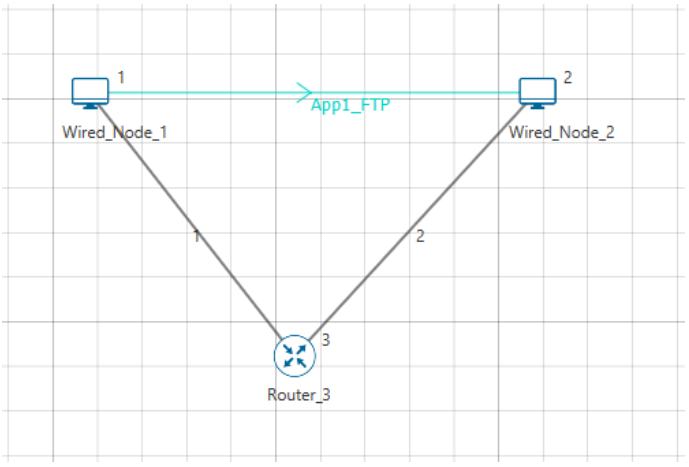
Step 1:

Open Examples → Connection-Establishment-in-TCP as shown below:



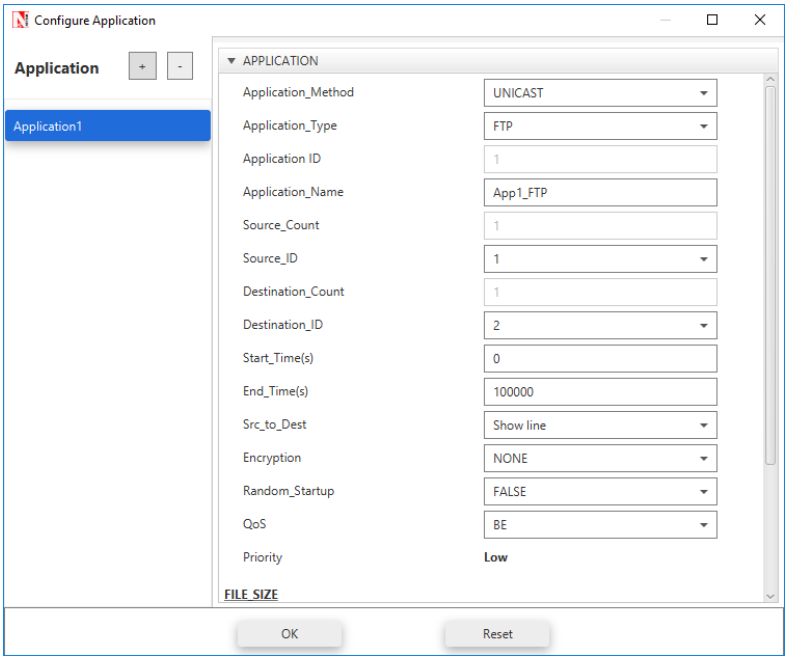
Step2:

Click & drop Wired Nodes and Router onto the Simulation Environment and link them as shown below.



Step3:

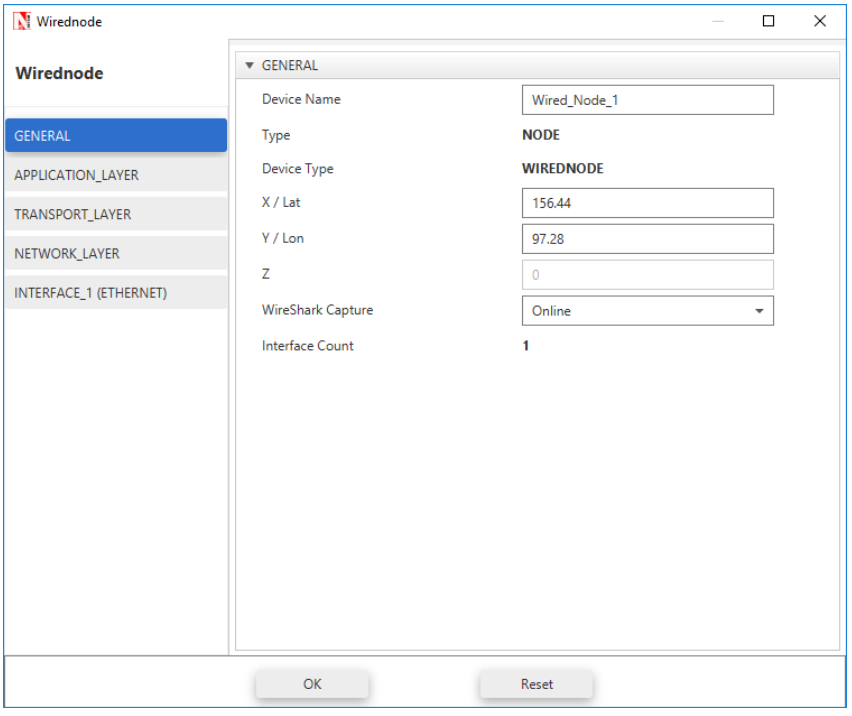
To run the simulation, click the Application icon present on ribbon and change the Application_Type to FTP. The Source_Id is 1 and Destination_Id is 2.



Router Properties: Accept default properties for Router.

Enabling Wireshark Capture on Source Node:

- Right Click on source node and go to properties.
- Under General properties enable Wireshark Capture option as “Online”.
- Click OK. Once the simulation starts, Wireshark capture all the packets.



Click on run simulation and set Simulation Time as 10 sec. In the Static ARP Configuration tab set Static ARP as Disable.

4.3 Output

The following results will be obtained. Open Wireshark window and you would see

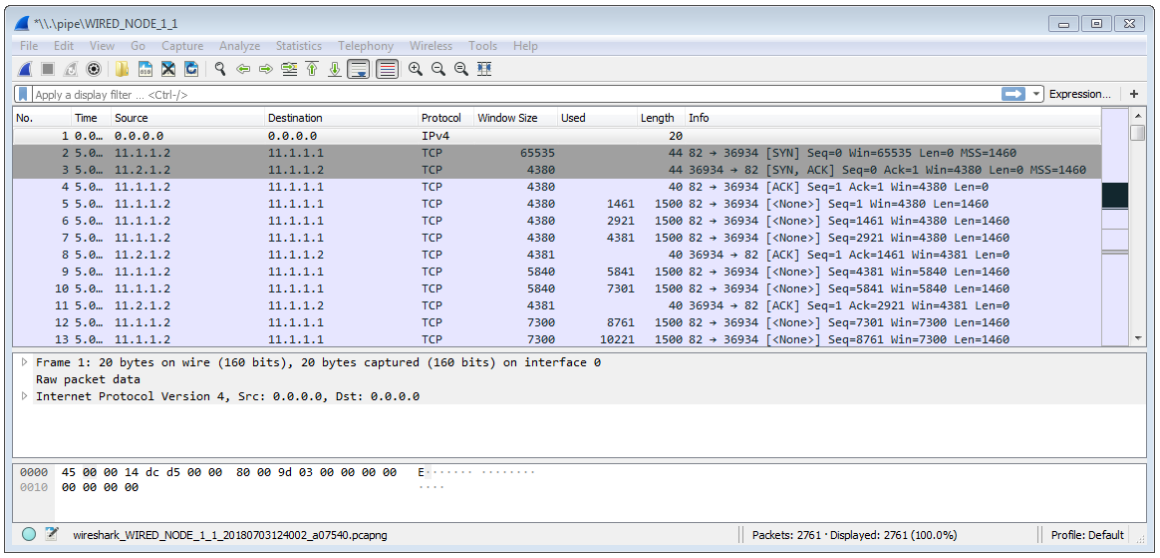


Fig: 3-way handshake captured in Wireshark

4.4 Inference

In the above figure we can see that NODE-1 (check the IP address of NODE-1 from the scenario) sends a control packet of type TCP_SYN requesting the connection with the NODE-2 (find the IP address from the scenario). NODE-2 responds with the control packet of type TCP_SYN_ACK to NODE-1. This TCP_SYN_ACK is the ACK packet for the TCP_SYN packet. NODE-1 then sends the TCP_ACK to NODE-2 via ROUTER-3 making the CONNECTION_STATE as TCP_ESTABLISHED. Once the connection is established, data transmission starts and we see that data packet (size 1500 bytes) sent from the NODE-1 to the NODE-2.

5. Appreciate the mathematical modelling of TCP and understand the fundamental relationship between packet loss probability and TCP performance.

5.1 Part 1: Estimate transmission rate in a simple error-less network

(***Reference:*** *Chapter 5 of the book, High Performance TCP/IP Networking. Concepts, Issues and Solutions by Mahbub Hassan, Raj Jain*)

The two key processes that a model of TCP needs to include are

1. The dynamics of the window that define the number of packets that a TCP source can transmit into the network
2. The packet loss process that indicates current traffic loads or congestion within the network

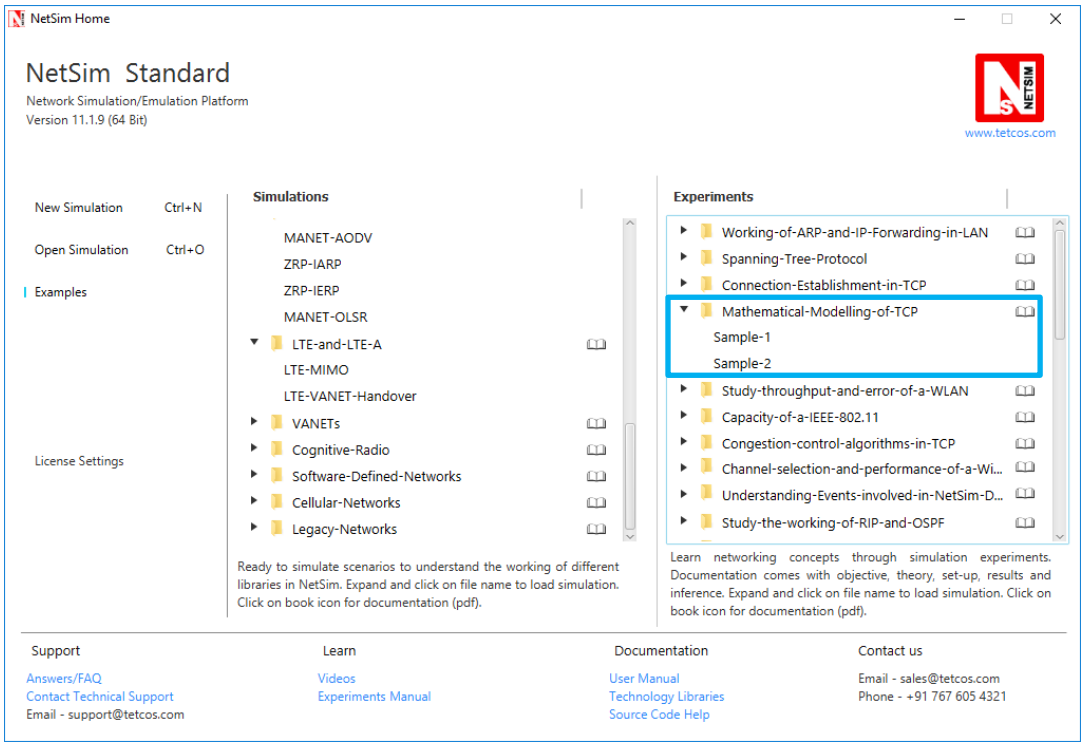
Now, during the interval in which TCP receives information that packets are not being lost in the network, TCP increases its window linearly. When the sources deduces that a packet has been lost it reduces its window by a factor of the current window size.

The standard assumption in this case is that the window size is related to the transmission rate by the roundtrip time (RTT)

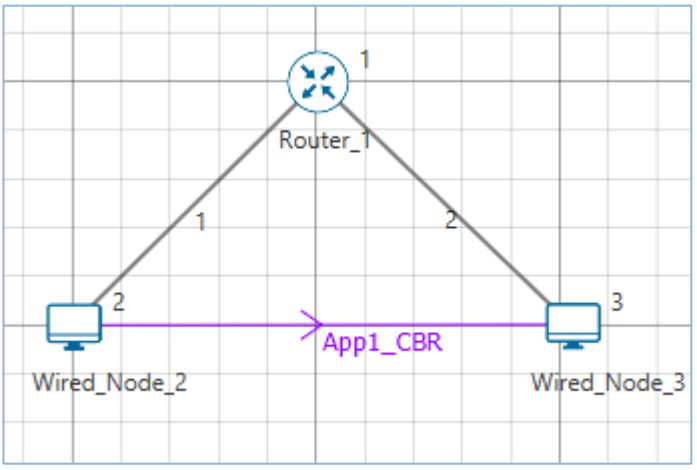
$$TranmissionRate = Window\ Size/RTT$$

5.1.1 Network setup:

Open Examples → Mathematical-Modeling-of-TCP as shown below:



Create a network scenario in Internetworks with 1 router and 2 wired nodes as shown below:



Set the Properties as mentioned below:

Application Properties	
Application Type	CBR
Source ID	2
Destination ID	3
Packet Size (Bytes)	1460
Inter Arrival Time (µs)	1168
Generation Rate (Mbps)	10

Wired Link Properties	
Propagation delay	100ms
Link speed	10Mbps
Bit error rate (both links)	0

Node Properties	Wired Node 2
Transport Layer Properties	
TCP	Enable
Congestion Control Algorithm	CUBIC

Accept default properties for all other parameters / devices.

Simulation Time – 100 Sec

Note: The Simulation Time can be selected only after performing the following two tasks,

- ***Setting the properties for the Wired Node, Wired Links and Application***
- ***Clicking on Run Simulation.***

5.1.2 Theoretical calculation:

In the output of simulation, the throughput measured equals the transmissions rates since there are packet errors or losses in the network. This implies that the expected theoretical value of throughput should be

$$Throughput(Mbps) = Window\ Size/RTT$$

Window size = 65535Bytes = 65535*8 = 524280bits

RTT = 100ms*4 = 400ms = 0.4s

Expected Throughput (Mbps) = 524280/0.4 = 1.3Mbps

Note that even though we have a 10 Mbps links the Throughput is limited to 1.3 Mbps, because TCP window size is limited to 65KB.

5.1.3 Simulation Output:

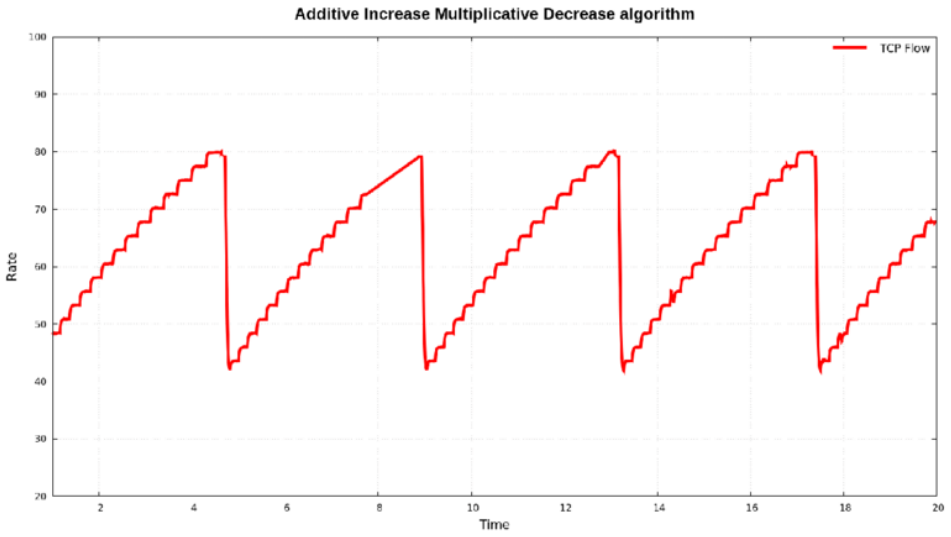
After running simulation, check throughput in Application metrics. The throughput obtained in NetSim is 1.24Mbps. The reason for the slightly lower throughput is due to the time taken initially for window to grow.

5.2 Part 2: TCP Model for a packet loss process

5.2.1 Theory

The simplest model in TCP that one can devise considers a periodic pattern in the dynamics of congestion window and the packet losses in steady state. TCP window evolves according to generic dynamics common to all TCP versions with each periodic loss event triggering a single, multiplicative window reduction. Because the steady state situation is being considered, a maximum window size W is always achieved by the time a packet is lost, which results in the window being reduced to W/2. The operation of the system in steady state means that packet losses occur with constant probability p, so that, on average, 1/p packets are transmitted into the network between each packet loss.

A trace of the window size results in a period saw tooth plot



The total number of packets transferred in a period would be the area underneath the trapezoidal shape of the period

$$No\ of\ Packets = \frac{1}{2}\frac{T}{RTT}\left(\frac{W}{2} + W\right)$$

Because of constant loss probability, p, the number of packets is given by 1/p. And the time taken to increase the window from W/2 to W is T = RTT. This gives us

$$\frac{1}{p} = \frac{W}{4}\left(\frac{W}{2} + W\right)$$

Solving this for W and applying that value of W gives us the average sending rate of TCP source, which is the number of packets transmitted during each period is

$$Sending\ Rate = \frac{1}{RTT}\left(\sqrt{\frac{3}{2p}}\right)$$

Where

RTT – Round Trip Time

p – Packet Error Rate = $1 - (1 - BER)^{Packet\ length}$

Node Properties	Wired Node 2
Transport Layer Properties	
TCP	Enable
Congestion Control Algorithm	CUBIC

5.2.2 Network Design and Set up:

Open the network created in PART1 and set the parameters shown below:

Set the Properties as mentioned below:

Application Properties	
Application Type	**CBR**
Source ID	**2**
Destination ID	**3**
Packet Size (Bytes)	**1460**
Inter Arrival Time (µs)	**1168**
Generation Rate (Mbps)	**10**

Wired Link Properties	
Propagation delay	**100ms**
Link speed	**10Mbps**

Bit error rate (both links)	**0.000001**

Enable Packet Trace and accept default properties for all other parameters / devices.

Simulation Time – 100 Sec

Note: The Simulation Time can be selected only after performing the following two tasks,

- ***Setting the properties for the Wired Node, Wired Links and Application***
- ***Clicking on Run Simulation.***

5.2.3 Observation:

Even though the packet size at the application layer is 1460 bytes, as the packet moves down the layers, some overhead is added which results in a greater packet size. This is the actual payload that is transmitted by the physical layer (1526 Bytes). The overheads added in different layers are shown in the table below and can be obtained from packet trace:

Layer	Overhead (Mbps)
Transport	20
Network	20
MAC	26
Physical	0
Total	**66**

Therefore, the payload size = Packet Size + Overhead

= 1460 + 66

= 1526 bytes

5.2.4 Theoretical calculation:

The number of packets transmitted per second is

$X = 1/RTT\sqrt{3/2p}$

RTT = 4*100µs = 0.4 s

Packet length = 1526*8 = 12208 bits

Packet Error Rate = $1 - (1 - 0.000001)^{12208}$ = 0.012

$= 1/0.4\sqrt{\frac{3}{2*0.012}} = 27.95$ packets per second

Throughput (Mbps) = 27.95*1526*8 = 0.32Mbps

5.2.5 Output:

After running simulation, check throughput in Application metrics. The throughput obtained in NetSim is 0.26 Mbps.

The reason for the slight difference in the results in that the theoretical model is a simplistic model that does not factor in additional TCP features such as a) limits on the window size enforced by the receiver b) Timeouts c) Duplicate ack's etc.

6. Study how throughput and error of a Wireless LAN network changes as the distance between the Access Point and the wireless nodes is varied

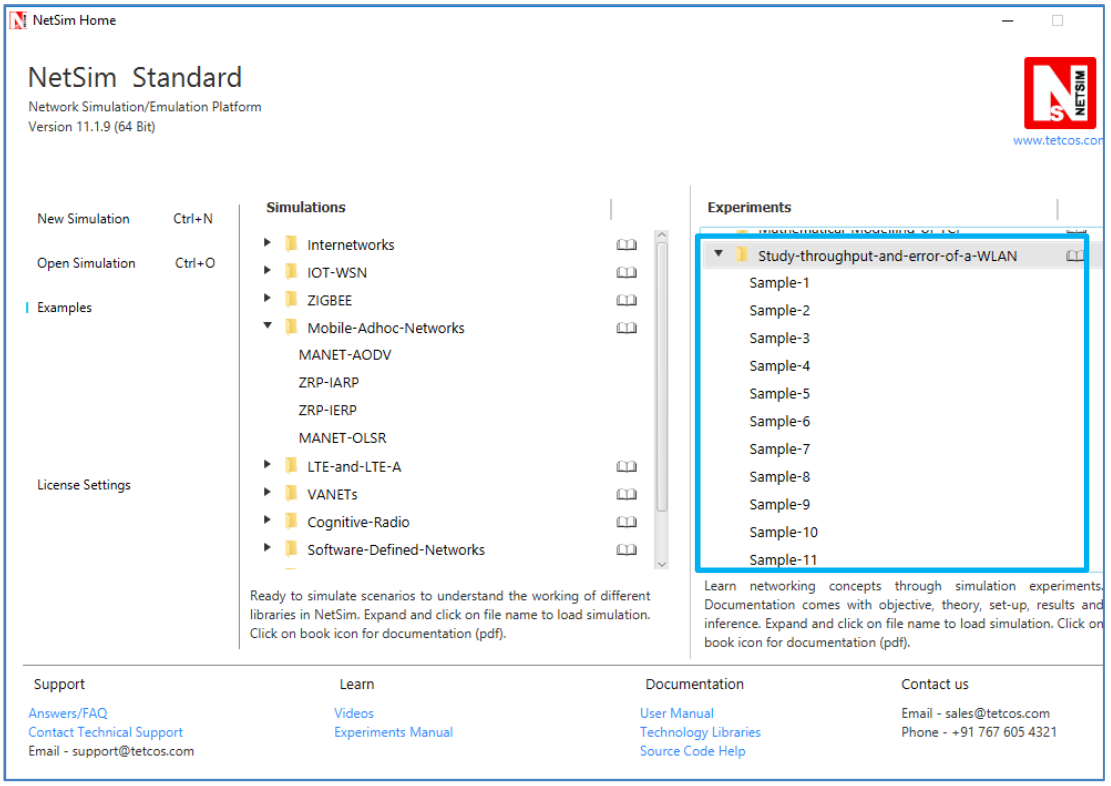
6.1 Introduction

In most of the WLAN products on the market based on the IEEE 802.11b technology the transmitter is designed as a Direct Sequence Spread Spectrum Phase Shift Keying (DSSS PSK) modulator, which is capable of handling data rates of up to 11 Mbps. The system implements various modulation modes for every transmission rate, which are Different Binary Phase Shift Keying (DPSK) for 1 Mbps, Different Quaternary Phase Shift Keying (DQPSK) for 2 Mbps and Complementary Code Keying (CCK) for 5.5 Mbps and 11 Mbps.

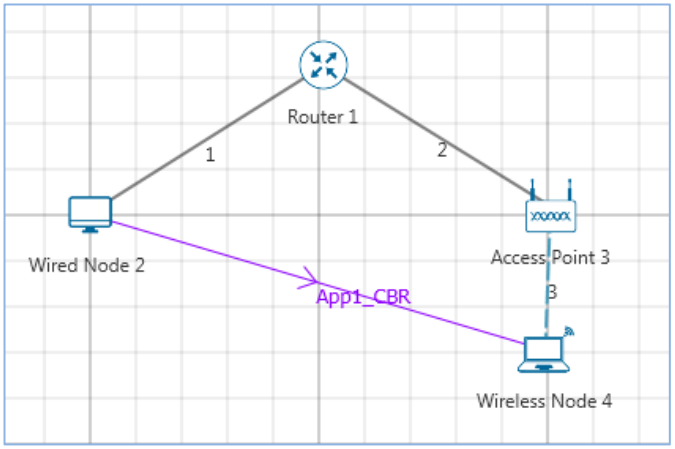
802.11b IEEE standard provides four data rates, 1 Mbps, 2 Mbps, 5.5 Mbps, and 11 Mbps. The rate is decided based on the received power and the errors in the channel. Note a higher data rate does not necessarily yield a higher throughput since packets may get errored. Only when the channel conditions are good, does a higher data rate give a higher throughput. In a realistic WLAN environment, the channel condition can vary due to path loss, fading, and shadowing. In NetSim to accommodate different channel conditions, rate adaptation is commonly employed.

6.2 Network Setup

Open Examples → Study-throughput-and-error-of-a-WLAN as shown below:



Create a scenario with 1 Wired Node, 1 Router, 1 Access Point and 1 Wireless Node as shown below:



Edit the following properties of Wireless Node 4:

Wireless Node 4 Properties	
X/Lat	200
Y/Lon	120
Interface_Wireless properties	
IEEE Standard	802.11b
Rate _Adaptation	False

Edit link properties as shown:

Wireless Link Properties	
Channel Characteristics	Path loss only
Path Loss Model	Log_Distance
Path Loss Exponent	3

Wired Link Properties	
Uplink Speed (Mbps)	100
Downlink Speed (Mbps)	100
Uplink BER	0.0000001
Downlink BER	0.0000001

Also edit the following properties of Wired Node 2:

Wired Node Properties 2	
TCP	Disabled

Set the properties of Access Point as follows:

Access Point Properties	
X/Lat	200
Y/Lon	70
Rate Adaptation	False

Click on the Application icon present on the ribbon, set properties and run the simulation.

Application Properties	
Application Type	**CBR**
Source_Id	**2**
Destination_Id	**4**
Packet Size	
Distribution	**Constant**
Value (Bytes)	**1450**
Packet Inter Arrival Time	
Distribution	**Constant**
Value (micro sec)	**770**

Enable Packet trace and run simulation for 10 seconds.

Note: The Simulation Time can be selected only after the following two tasks,

- ***Set the properties for all the devices and links.***
- ***Click on Run Simulation button.***

Similarly do the other samples by varying the distance between Access Point and Wireless Node.

- **Sample 2**: Distance from Wireless Node 4 to Access Point is 50m.
- **Sample 3**: Distance from Wireless Node 4 to Access Point is 60m.

……. And so on till 180 meters distance.

6.3 Output:

Note down the values of Data rate and Throughput for all the samples and compare with IEEE standards

Phy rate can be calculated using packet trace by using the formula shown below:

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

192 micro seconds is the approximate preamble time for 802.11b

Calculate PHY rate for all the data packets coming from Access Point to Wireless node. For doing this please refer section 6.5.1 How to set filters to NetSim Packet Trace file from NetSim's User Manual. Filter Packet Type to CBR, Transmitter to Access Point and Receiver to Wireless node.

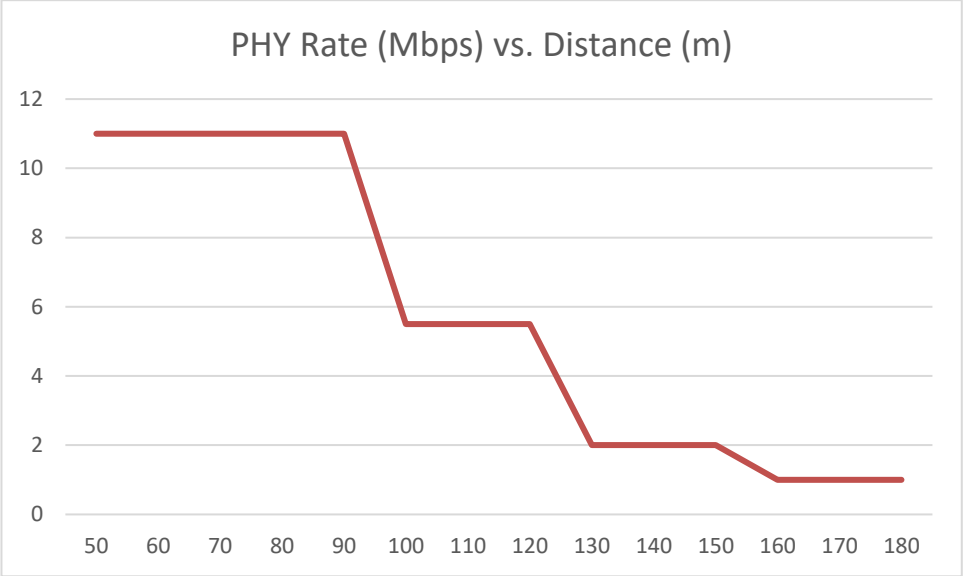
Since $\boldsymbol{PER = 1 - (1 - BER)^{PL}}$ where PER is packet error rate, PL is packet length in bits and BER is bit error rate, we get $\boldsymbol{BER = 1 - e^{\frac{log\ (1-PER)}{PL}}}$

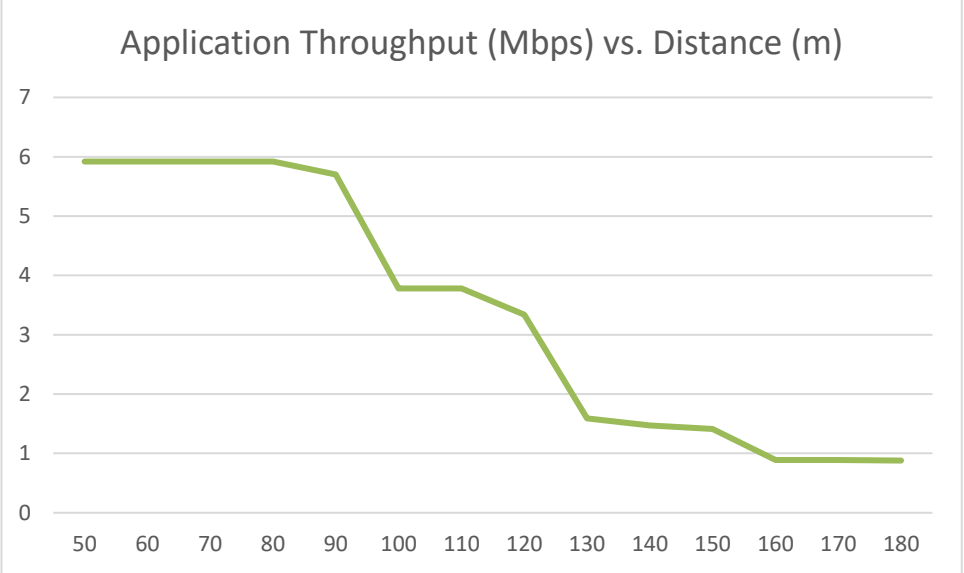
On tabulating the results you would see

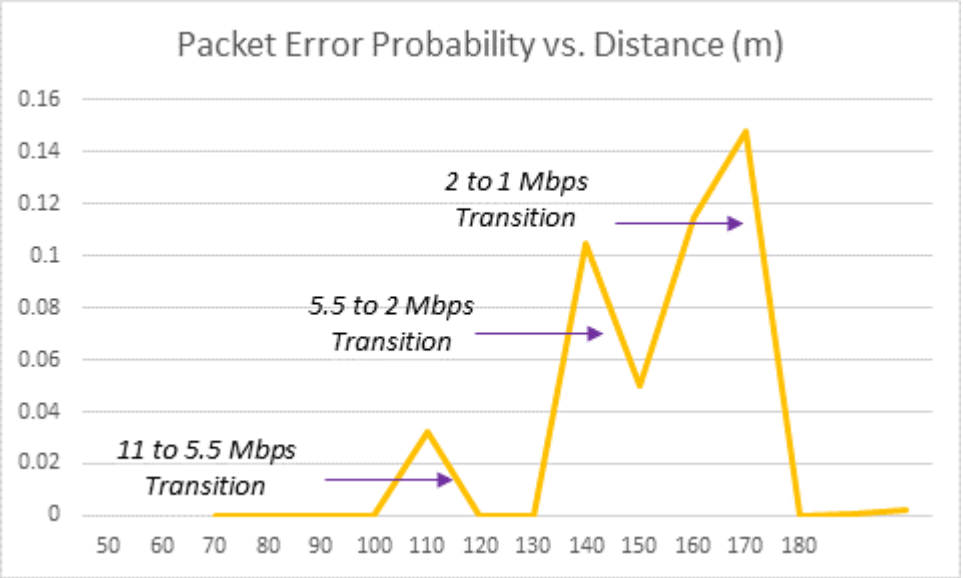
802.11b					
Distance (m)	**PHY rate in Mbps (Channel capacity)**	**Application Throughput (Mbps)**	**Packets Transmitted**	**Packets Errored**	**Packet error probability**
50	11	5.92	5109	0	0
60	11	5.92	5109	0	0
70	11	5.92	5109	0	0
80	11	5.92	5109	0	0
90	5.5	5.57	5066	264	0.052
100	5.5	3.78	3265	0	0
110	5.5	3.78	3265	0	0

120	2	1.61	1442	49	0.033
130	2	1.58	1441	85	0.058
140	2	1.50	1437	143	0.099
150	1	0.89	769	0	0
160	1	0.89	769	0	0
170	1	0.88	769	2	0.002
180	1	0.89	769	1	0.001

6.4 Inference:







Note: All the above plots highly depend upon the placement of nodes in the simulation environment. So, note that even if the placement is slightly different, the same set of values will not be got but one would notice a similar trend.

We notice that as the distance increases, the 802.11b PHY rate (channel capacity decreases) decreases. This is because the underlying data rate depends on the received power at the receiver.

$$Received\ Power = Transmitted\ Power - RF\ losses$$

RF losses are directly proportional to distance to the power of path loss exponent. As RF propagation losses increase, the received power decreases.

We can see that the rate drops from 11 Mbps to 5.5 Mbps at around 75m, and then to 2 Mbps at 90m and to 1 Mbps at 125m (in this case the path loss exponent is set to 3.5).

We also notice how the packet error rate increases with distance, then when the data rate changes (a lower modulation scheme is chosen), the error rate drops. This happens for all the transitions i.e. 11 to 5.5, 5.5 to 2 and from 2 to 1 Mbps.

One must note that WLAN involves ACK packets after data transmission. These additional packet transmission lead to reduced Application throughput of 5.8 Mbps (at lower distances) even though the PHY layer data rate is 11 Mbps and the error rates is almost NIL.

The application throughput is dependent on the PHY rate and the channel error rate, and one can notice it drops / rise accordingly.

7. How many downloads can a WiFi access point simultaneously handle?

7.1 Objective

In this experiment we will learn how to obtain n_θ, and in this process we will understand some interesting facts about how WiFi networks perform when doing file transfers.

7.2 Theory

WiFi has become the system of choice for access to Internet services inside buildings, offices, malls, airports, etc. In order to a obtain access to the Internet over WiFi a user connects his/her mobile device (a laptop or a cellphone, for example) to a nearby WiFi access point (AP). A popular use of such a connection is to download a document, or a music file; in such an application, the user's desire is to download the file as quickly as possible, i.e., to get a high throughput during the download. It is a common experience that as the number of users connected to an AP increases, the throughput obtained by all the users decreases, thereby increasing the time taken to download their files. The following question can be asked in this context.

If during the download, a user expects to get a throughput of at least θ bytes per second, what is the maximum number of users (say, n_θ) up to which the throughput obtained by every user is at least θ. We can say that n_θ is the *capacity* of this simple WiFi network for the *Quality of Service (QoS)* objectiveθ.

7.3 Procedure

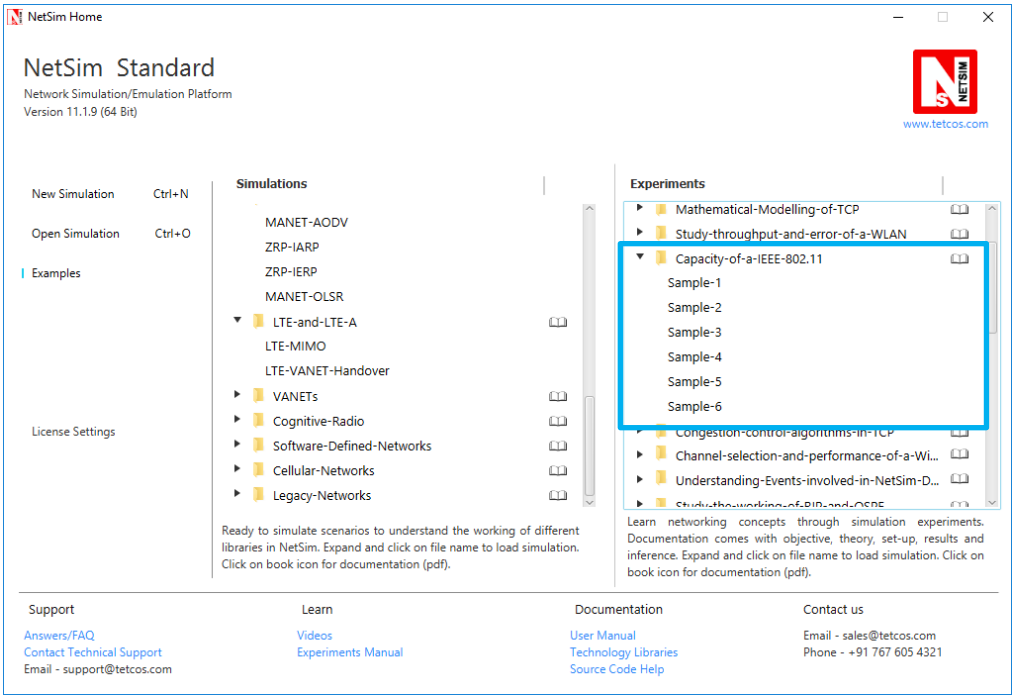
In NetSim, we set up a network comprising a server that carries a large number of large files that the users would like to download into their mobile devices. The server is connected to a WiFi AP, with the IEEE 802.11b version of the protocol, via an Ethernet switch. Several mobile devices (say N) are associated with the AP, each downloading one of the files in the server. The Ethernet speed is 100Mbps, whereas the mobile devices are connected to the AP at 11Mbps, which is one of the IEEE 802.11b speeds.

We observe, from the above description, that the file transfer throughputs will be limited by the wireless links between the AP and the mobile devices. There are two interacting mechanisms that will govern the throughputs that the individual users will get:

1. The WiFi medium access control (MAC) determines how the mobile devices obtain access to the wireless medium. There is one instance of the WiFi MAC at each of the mobile devices.

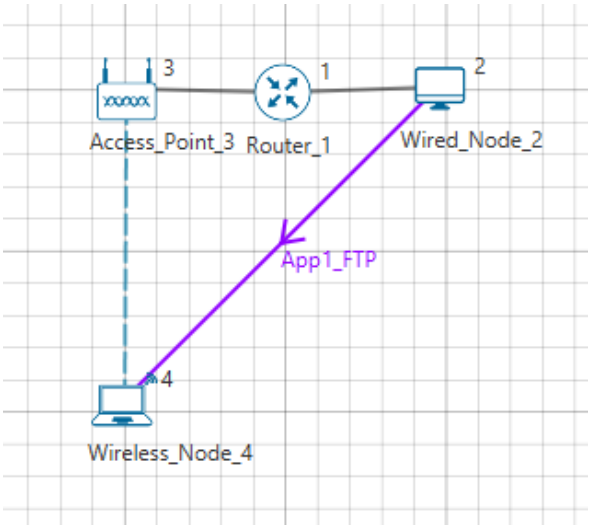
2. The end-to-end protocol TCP that controls the sharing of the wireless bandwidth between the ongoing file transfers. In our experiment, there will be one instance of TCP between the server and each of the mobile devices.

Step 1: Open Examples → Capacity-of-a-IEEE-802.11 as shown below:



Step 2:

Sample 1: Click and drop 1 Router, 1 Wired Node, 1 Access Point and 1 Wireless node and connect as per the following figure:

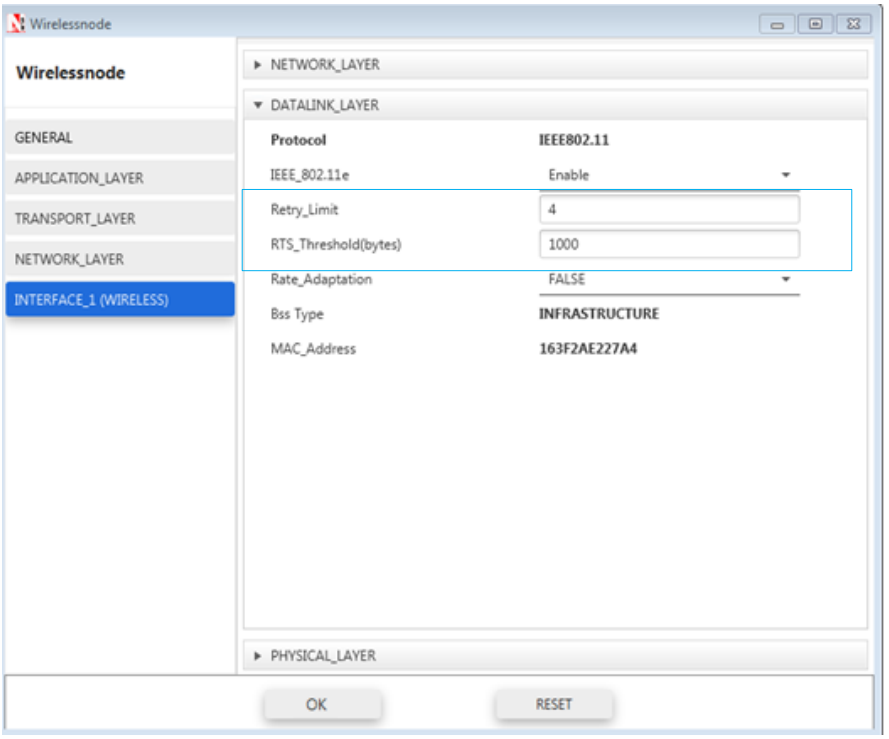


Step 3: Application Properties:

Application Properties	
Application Type	FTP

Source Id	Wired Node 2
Destination Id	Wireless Node 4
File size (Bytes)	100000000
File Inter arrival time (s)	1

Wireless Node Properties: Right click on Wireless Node → Select Properties → Click on Interface1_Wireless. Change the properties as shown below:



Similarly change RTS_Threshold value=1000 in Access point

Link Properties:

Wired Link	
Uplink BER rate	0
Downlink BER rate	0
Uplink Delay	0µs
Downlink Delay	0µs

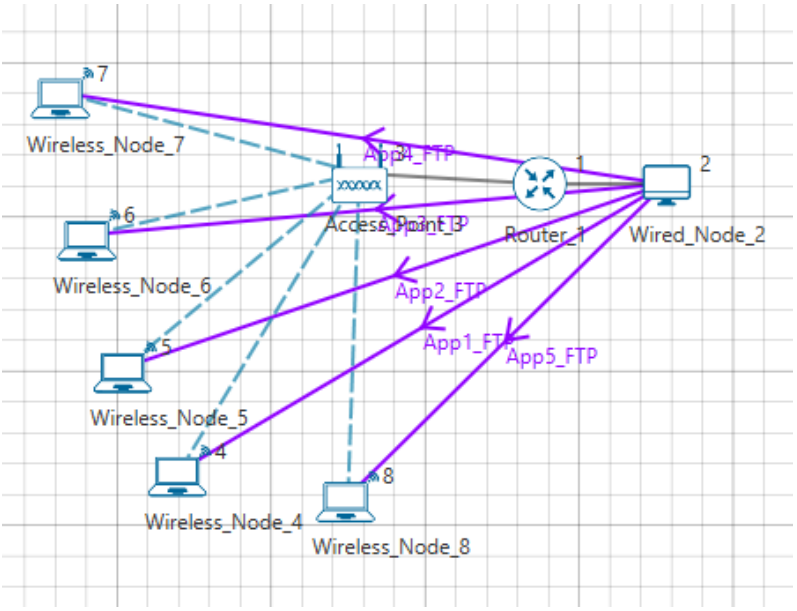
Wireless Link	
Channel Characteristics	No path loss

Step 4: Simulation time = 100sec. Click on run simulation and note down the throughput.

Sample 2:

No. of wireless nodes = 5

Drop another 5 Wireless Nodes and connect to Access Point as per the following figure:



Application Properties

Properties	App1	App2	App3	App4	App5
Application Type	FTP	FTP	FTP	FTP	FTP
Source Id	2	2	2	2	2
Destination Id	4	5	6	7	8
File size (Bytes)	100000000	100000000	100000000	100000000	100000000
File Inter arrival time	1s	1s	1s	1s	1s

Run simulation and note down the sum of throughputs obtained for all applications.

Note: Follow the same procedure for next samples with wireless nodes 10, 15, 20, 25 and note down the sum of throughputs for all applications.

7.4 Measurements and Output

Aggregated download throughput with different values of N (wireless nodes) is shown below:

Sample Number	Number of Devices	Sum of throughputs (Mbps)	Throughput Per Device (Mbps)
1	1	3.38	3.38
2	5	3.41	0.682
3	10	3.37	0.337
4	15	3.38	0.225
5	20	3.31	0.165
6	25	3.28	0.131

(Note: In the referred paper we see that, the throughput value for 11 Mbps WLAN is 3.8 Mbps. Please note that this is the aggregate PHY throughput of the AP. However in NetSim, we are calculating the total Application throughput.)

To derive the PHY layer throughput from the APP layer throughput, we need to add overheads of all layers

Layer	Overhead (Bytes)
Transport Layer	20
Network Layer	20
MAC Layer	40
PHY layer	48µs = (11*48)/8 = 66
Total Overhead	**146**

*PHY_Throughput = APP_Throughput * 1606/1460 = 3.41*1606/1460 = 3.79 Mbps)*

7.5 Inference:

We see that as the number of devices increase the aggregate (combined) throughput remains constant whereas the throughput per user decreases.

As discussed earlier, our goal was to identify that if during the download, a user expects to get a throughput of at least θ bytes per second, what is the maximum number of users (say, n_θ)?

If we set θ to be 300 Kbps then we see that from the output table that the maximum number of users who can simultaneously download files is 10 (n_θ)

7.6 Reference Documents

1. *Analytical models for capacity estimation of IEEE 802.11 WLANs using DCF for internet applications. George Kuriakose, Sri Harsha, Anurag Kumar, Vinod Sharma*

8. Understand the working of Slow start and Congestion Avoidance (Old Tahoe), Fast Retransmit (Tahoe) and Fast Recovery (Reno) Congestion Control Algorithms in TCP.

8.1 Theory:

One of the important functions of a TCP Protocol is congestion control in the network. Given below is a description of the working of Old Tahoe and Tahoe variants (of TCP) control congestion.

Old Tahoe: Old Tahoe is one of the earliest variants of TCP. It implements two algorithms called slow start and congestion avoidance to update the congestion window.

Slow Start: At the start of data transmission the size of congestion window is one. This means TCP can send only one packet until it receives an acknowledgement. When the ACK is received by the sender the congestion window increases to two. Now the sender can send two data packets. Upon the arrival of every new ACK the sender increases its congestion window by one. This phase is known as the slow start phase where the congestion window increases exponentially. So on the arrival of a new ACK, $cwnd$ + = MSS;

Congestion Avoidance: TCP will continue the slow start phase until it reaches a certain threshold, or if packet loss occurs. Now it enters in to a phase called congestion avoidance. Here the congestion window grows linearly. This means that the congestion window increases from 'n' to 'n+1' only when it has received 'n' new ACKs. The rate of growth of congestion window slows down because this is the stage where TCP is susceptible to packet loss. The formula used here is $cwnd$ += ($SMSS * SMSS$)/cwnd

Tahoe (Fast Retransmit): TCP Tahoe implements all the above mentioned algorithms used by Old Tahoe. The Fast Retransmit algorithm was included in Tahoe to improve the response time of TCP.

Fast Retransmit: One of the major drawbacks of Old Tahoe is that it depends on the timer to expire before it can retransmit a packet. TCP Tahoe tries to improve upon Old Tahoe by implementing the Fast Retransmit algorithm. Fast Retransmit takes advantage of the fact that duplicate ACKs can be an indication that a packet loss has occurred. So, whenever it receives 3 duplicate ACKs it assumes that a packet loss has occurred and retransmits the packet.

Reno (Fast Recovery): TCP Reno retains the basic principles of Tahoe such as Slow Start, Congestion Avoidance and Fast Retransmit. However it is not as aggressive as Tahoe in the

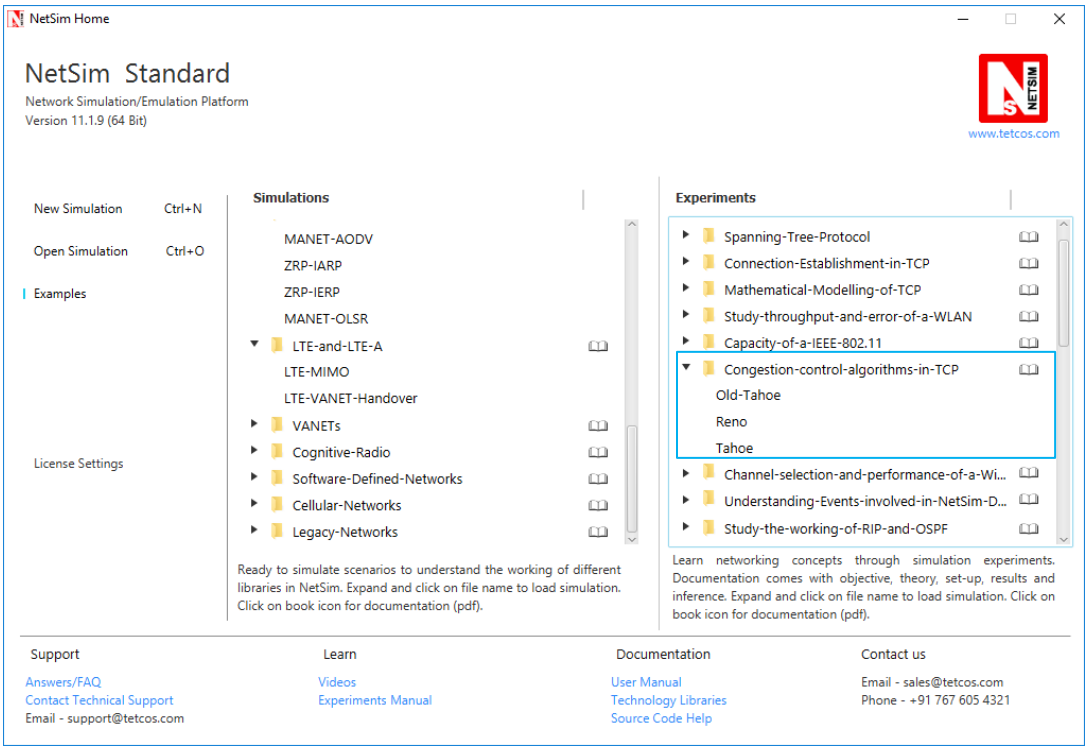
reduction of the congestion window. Reno implements the Fast Recovery algorithm which is described below.

Fast Recovery: The congestion window drop to one on the arrival of a 3 duplicate ACK can be considered as an extreme precaution. Arrival of 3 duplicate ACKs corresponds to light congestion in the network and there is no need for the congestion window to drop down drastically. The Fast Recovery algorithm does the following on the arrival of a third duplicate ACK:

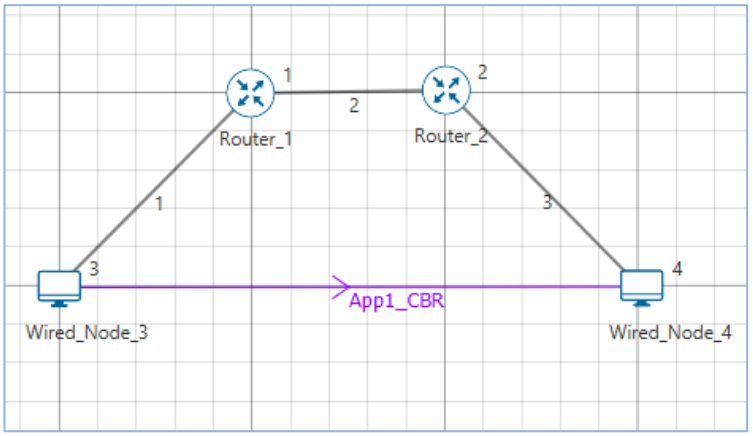
- The threshold value is set to half of the congestion window. $ssthresh$ = $cwnd$/2.
- The congestion window is now set to be threshold plus three times the MSS. $cwnd$ = $ssthresh$ + 3 $*$ $SMSS$.
- On the arrival of another duplicate ACK the congestion window increases by one MSS. This is done because an ACK signifies that a segment if out of the network and the sender can pump in another packet into the network. This is somewhat similar to slow start. $cwnd$ += $SMSS$.
- TCP remains in fast recovery phase until it receives a higher ACK from the receiver.
- On receiving a higher ACK the congestion window is set to the threshold value. From now onwards congestion avoidance is followed. $cwnd$ = $ssthresh$.

8.2 Network Set Up:

Open Examples → Congestion-control-algorithms-in-TCP as shown below:



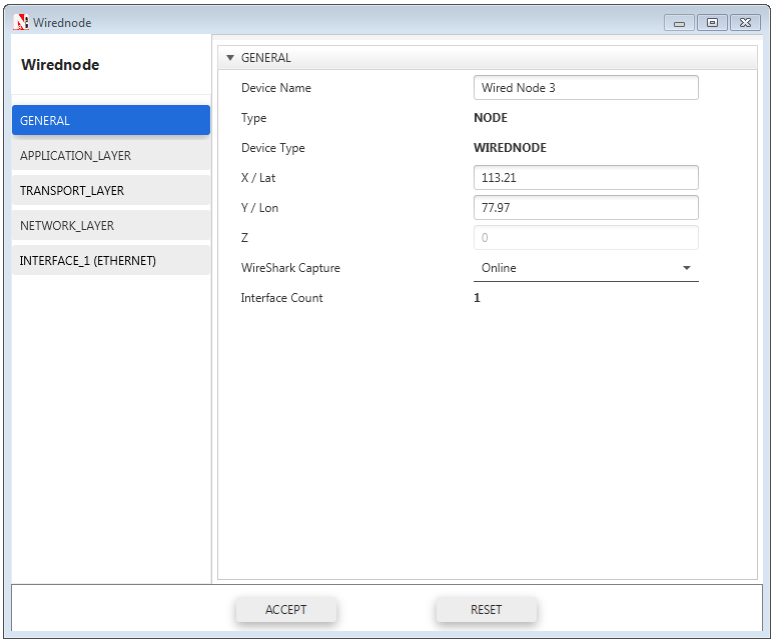
Create a network scenario in Internetworks with 2 routers and 2 wired nodes as shown below:



SAMPLE 2: Set the Properties as mentioned below:

Enabling Wireshark Capture on Source Node:

- Right Click on source node and go to properties.
- Under General properties enable Wireshark Capture option as “Online”.
- Click OK. Once the simulation starts, Wireshark will capture packets.



Wired Link Properties	
Node to Router propagation delay	5µs
Router to Router propagation delay	100µs
Node to Router Link speed	20Mbps
Router to Router Link speed	100Mbps
Bit error rate (all links)	10 e(-7)

Wired links between the Node and router are configured with a rate lower than that of the packet generation rate to create bottle necks.

Node Properties	Wired Node 3
Transport Layer Properties	
TCP	Enable
Congestion Control Algorithm	Old Tahoe

Application Properties	
Application Type	CBR
Source ID	3
Destination ID	4
Packet Size (Bytes)	1460
Inter Arrival Time (µs)	400

Packet size and inter arrival time are set in such a way that a traffic of 30 Mbps (Approx.) is generated.

Accept default properties for all other parameters / devices.

Simulation Time – 30 Sec

(Note: The Simulation Time can be selected only after the following two tasks,

- ***Set the properties for the Wired Node, Router, Wired Links and Application.***
- ***Click on Run Simulation)***

SAMPLE 2: Change the following properties in Wired Node 3 and run the simulation for 30 seconds. All other properties are default.

Node Properties	Wired Node 3
Transport Layer Properties	
TCP	Enable
Congestion Control Algorithm	Tahoe

SAMPLE 3: Change the following properties in Wired Node 3 and run the simulation for 30 seconds. All other properties are default.

Node Properties	Wired Node 3
Transport Layer Properties	
TCP	Enable
Congestion Control Algorithm	Reno

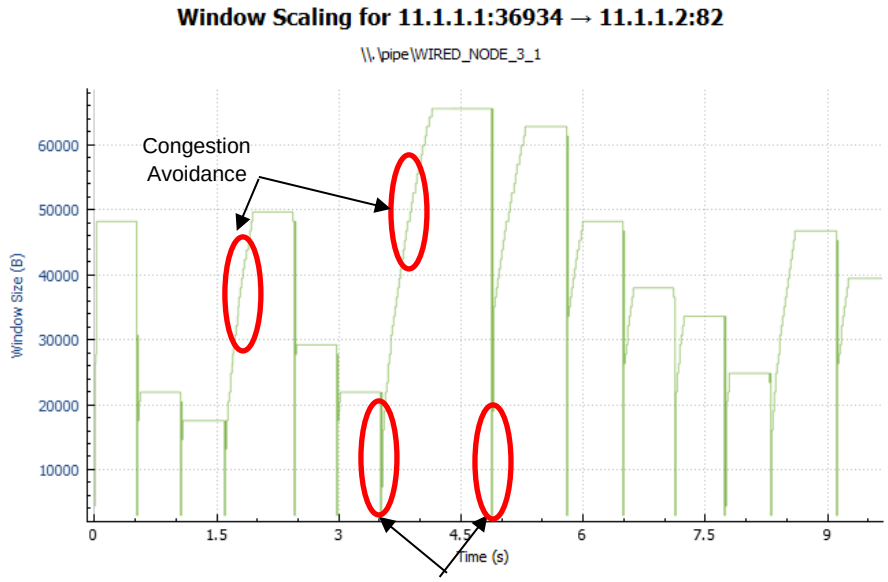
8.3 Output

Comparison Table:

Congestion control algorithm	Throughput (Mbps)
Old Tahoe	7.26
Tahoe	16.10
Reno	17.76

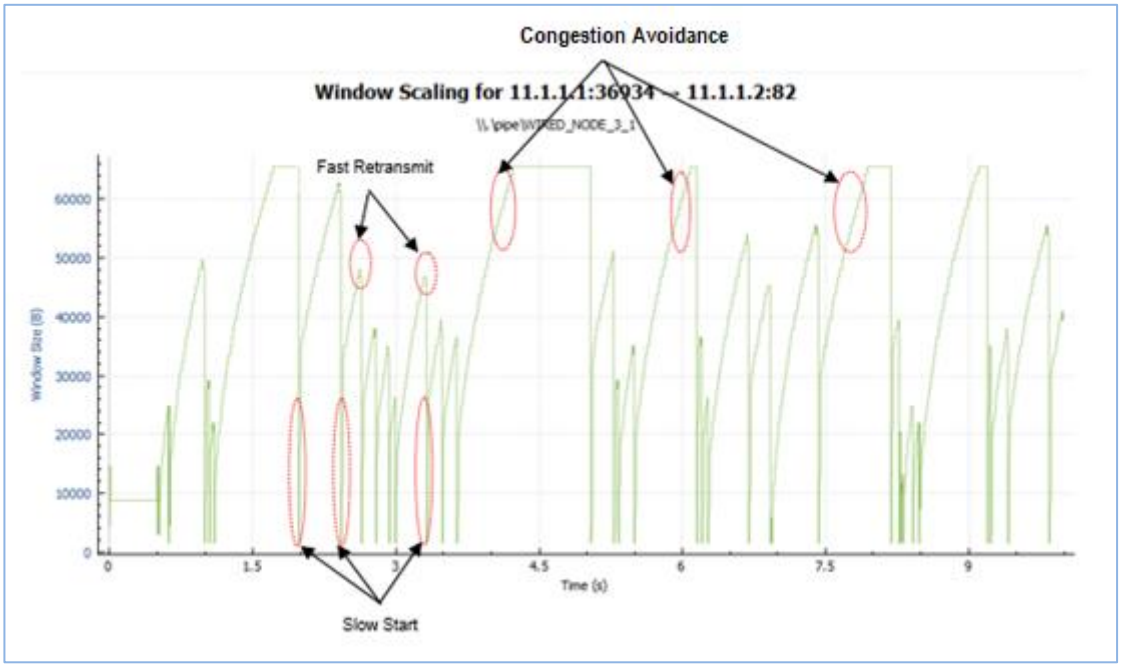
From the above table, throughput for Tahoe is high since Tahoe retransmits the packets faster than Old Tahoe. Throughput for Reno is higher compared to Tahoe since *cwnd* is set to *ssthresh* instead of setting to 1 SMSS. To create wireshark graph, click on data packet i.e. <None>. Go to Statistics-> TCP Stream Graphs->Window Scaling. User will get a graph similar to the one shown below:

Sample1 (Old Tahoe):



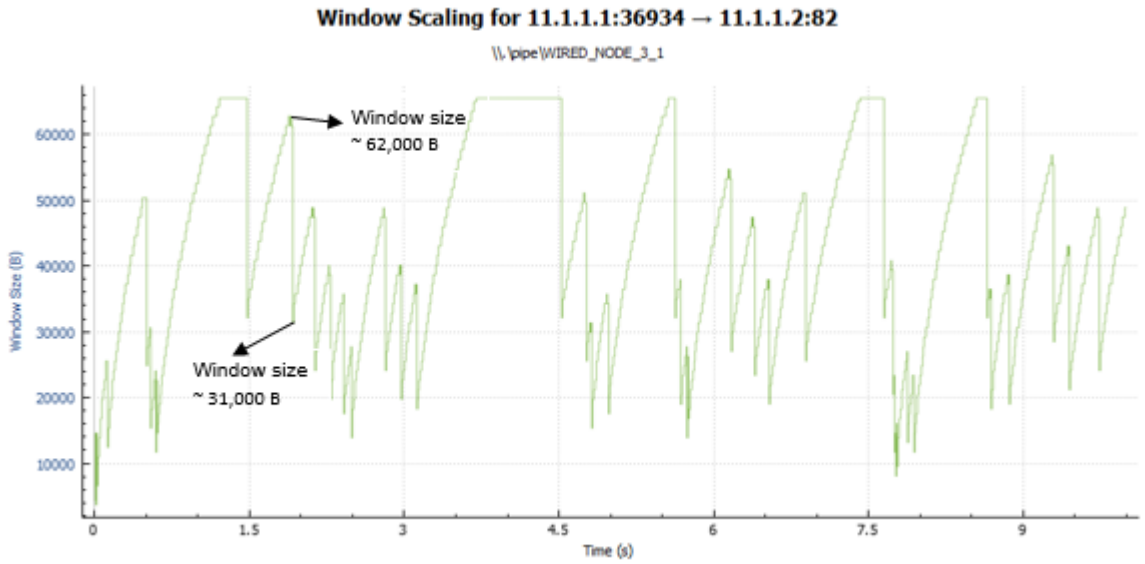
The graph shown above is a plot of Congestion window vs. Time. Each point on the graph represents the congestion window at the time when the packet is sent. You can observe that after every congestion window drop (caused by packet loss or timer expiry) Old Tahoe enters slow start, thereby increasing its window exponentially and then enters congestion avoidance where it increases the congestion window linearly.

Sample2 (Tahoe):



Tahoe uses the fast retransmit algorithm with which it responds to packet errors faster than Old Tahoe. Comparing the graphs of Old Tahoe and Tahoe one can observe that the latter drops the congestion window and retransmits faster than Old Tahoe (when three duplicate ack's are received).

Sample3 (Reno):



TCP Reno upon receiving the third duplicate ACK sets its congestion window according to the formula cwnd = cwnd/2 + 3*MSS. This can be observed in the above graph. On further arrival of duplicate ACKs it increases its congestion window by one MSS. If it receives a higher ACK then it drops the congestion window to the new threshold value. This mechanism is known as fast recovery.

8.4 Inference:

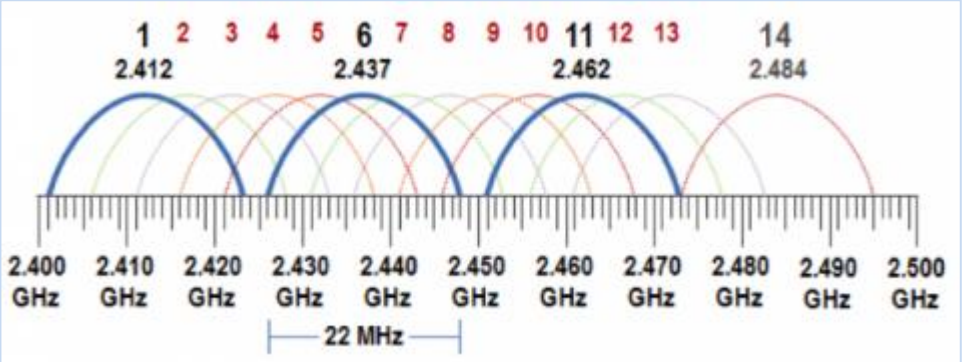
From this experiment we were able to understand how selection of TCP congestion control algorithm can have an impact on the throughput experienced by the applications. The major difference between Old Tahoe, Tahoe and Reno algorithms is the time taken to retransmit packets and the way congestion window size is reduced. There is a considerable improvement in the performance as we go from Old Tahoe to Tahoe and then to Reno. This is evident from the throughput readings obtained from the simulations performed.

9. Understand how channel selection can improve performance of a Wi-Fi network

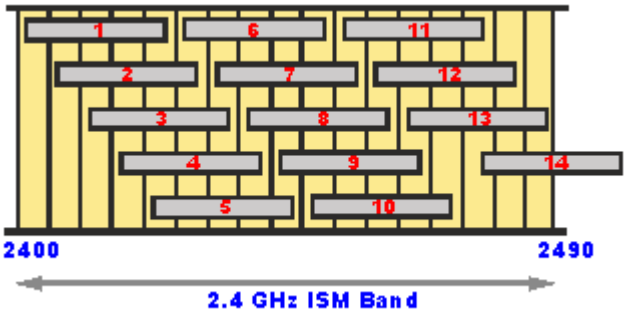
9.1 Theory

The 2.4 GHz band is 100 MHz wide and spans from 2.4 GHz to 2.5 GHz. The IEEE standard divides the 2.4 GHz band into 14 separate channels. Channels are designated by their center frequency and how wide the channel is depends on the technology used by the 802.11 transmitter. Unfortunately, the distance between channel center frequencies in the 2.4 GHz spectrum is only 5 MHz (and each channel is 22 MHz wide), which means that the channels have overlapping frequency space.

Only channels 1, 6, and 11 are separated from each other by enough frequencies in such a way that they do not overlap. Enterprise deployments of three or more access points in the 2.4 GHz band should normally only use channels 1, 6, and 11 which are shown below.



802.11b channels in 2.4GHz band



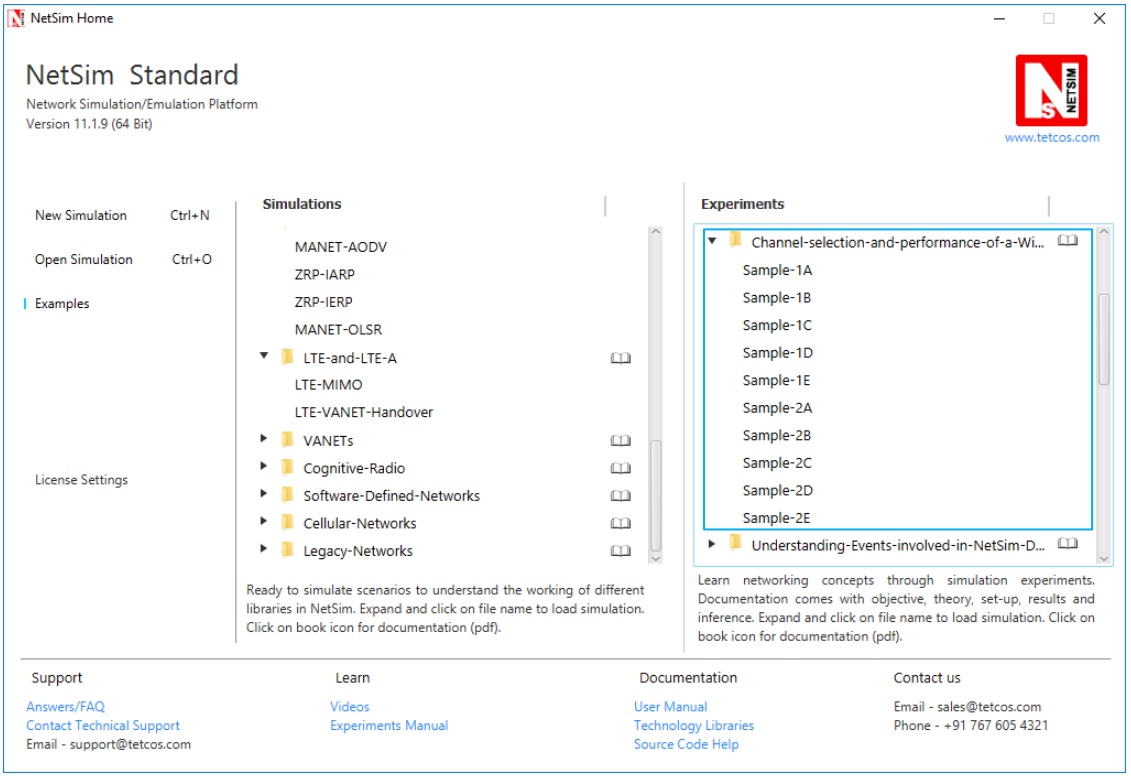
From the diagram above, it can be seen that Wi-Fi channels 1, 6, 11, or 2, 7, 12, or 3, 8, 13 or 4, 9, 14 (if allowed) or 5, 10 (and possibly 14 if allowed) can be used together as sets. Often WiFi routers are set to channel 6 as the default, and therefore the set of channels 1, 6 and 11 is possibly the most widely used.

9.2 Procedure:

Input Samples:

Sample 1.a: Step 1:

Open Examples → Channel-selection-and-performance-of-a-WiFi-network as shown below:



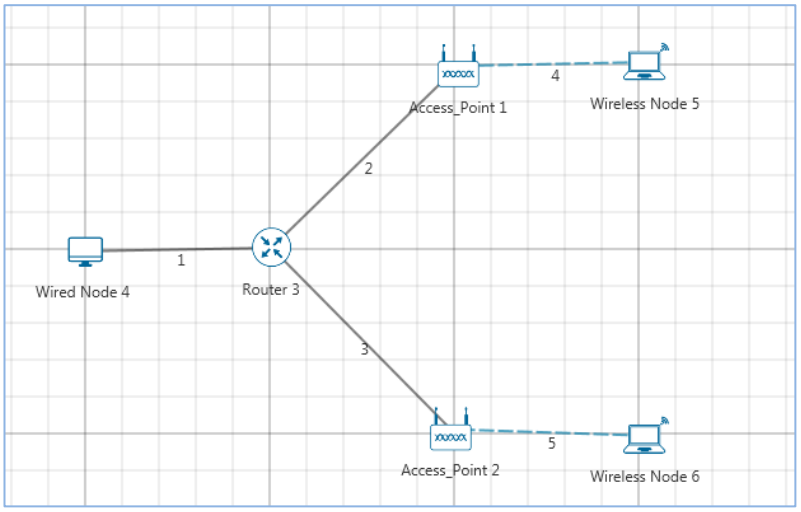
Step 2:

Devices Required:

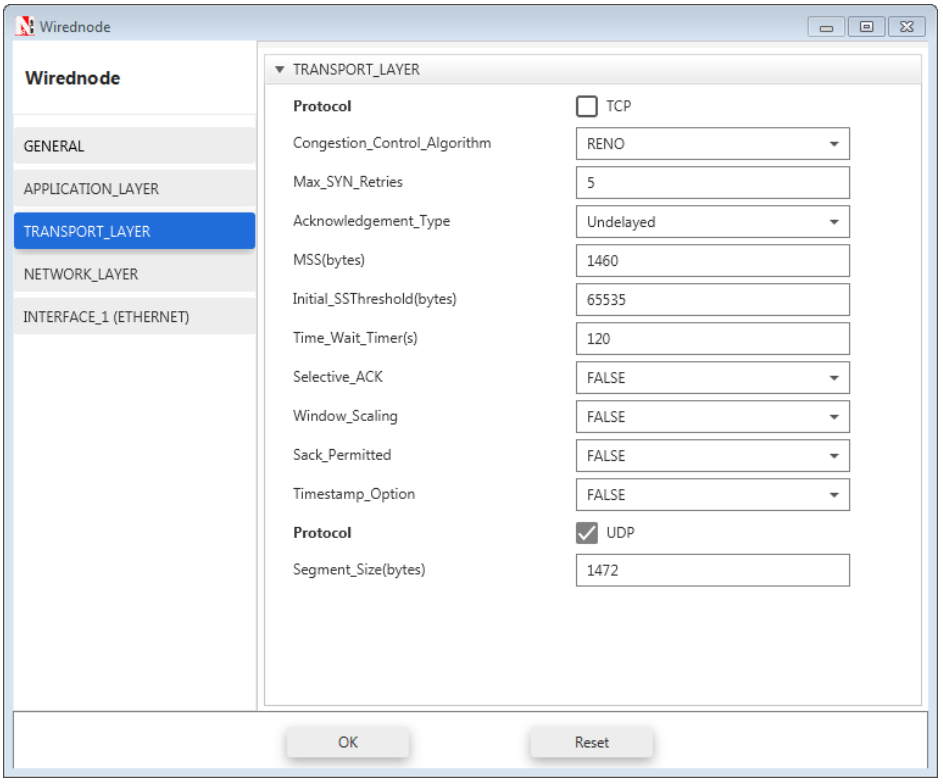
2 Wireless Node, 2 Access point,

1 Router, 1 Wired Node.

Note: Distance between Access point & Wireless node should be 50 m.



Step 3:



Node properties: Disable TCP in all nodes in Transport layer

Access Point Properties	AP 1	AP 2
General Properties		
X_Coordinate	**200**	**200**
Y_Coordinate	**50**	**100**

Step 4:

Wireless Node Properties:

Right Click on Wireless Node and Select Properties, select Interface_Wireless and go to Physical Layer Change the Standard as IEEE 802.11b for wireless Node 5 and IEEE 802.11b for Wireless Node 6.

Wireless Node Properties	Wireless Node 5	Wireless Node 6
General Properties		
X_Coordinate	**250**	**250**
Y_Coordinate	**50**	**100**

Access Point Properties:

In Access Point properties, select Interface_Wireless and go to Physical Layer and change the Standard Channel as follows:-

1_2412 for **IEEE 802.11b** Standard in Access point 1

5_2432 for **IEEE 802.11b** Standard in Access point 2.

Change the Standard Channel as **1_2412** and Standard as **IEEE 802.11b** for Access Point 1 and **5_2432** for Access Point 2

Wireless Link Properties:

Right click on Wireless Link and Change the Channel characteristics to "**No Path Loss**"

Step 5:

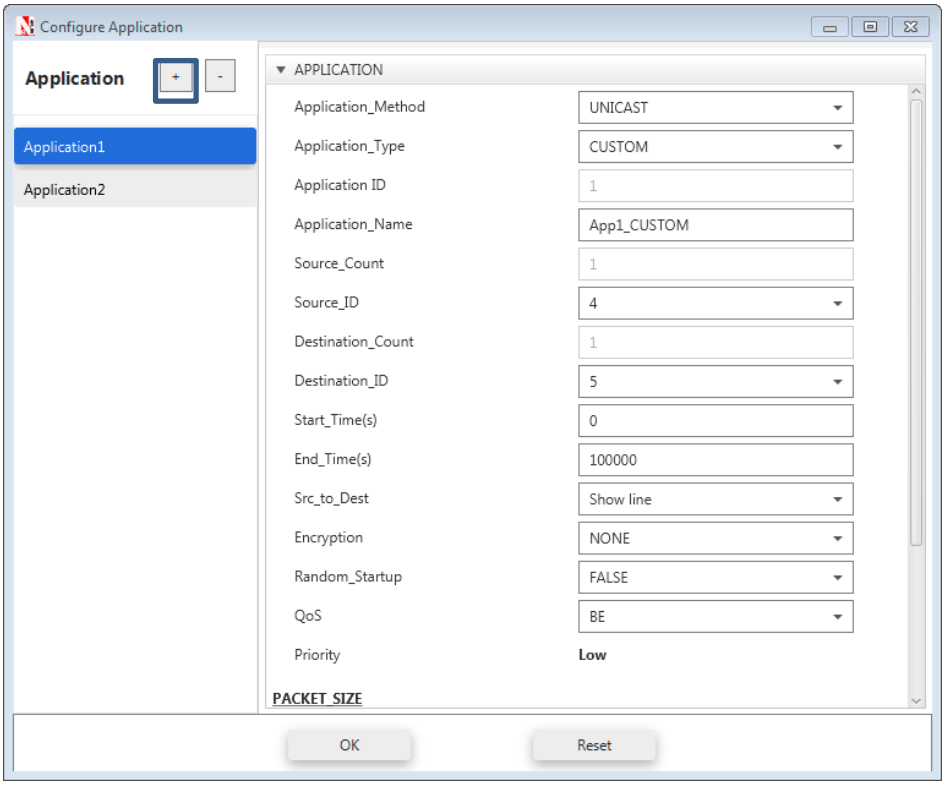
Click on Application present on the ribbon. Select Properties as shown below:

Application Type	Custom	Custom
Source ID	Wired Node 4	Wired Node 4
Destination ID	Wireless Node 5	Wireless Node 6
Packet Size		
Distribution	Constant	Constant
Value(Bytes)	1460	1460
Inter Arrival Time		
Distribution	Constant	Constant

Value(µs)	2336	2336

NOTE: The procedure to create multiple applications are as follows:

Click on the ADD button present in the top left corner to add a new application.



Simulation Time – 10 sec

Sample 1.b - Go to Application Properties (Click on Application icon), Change the INTER ARRIVAL TIME: 1168(Micro Sec)

Sample 1.c - Go to Application Properties, Change the INTER ARRIVAL TIME: 778(Micro Sec)

Sample 1.d - Go to Application Properties, Change the INTER Arrival TIME: 584(Micro Sec)

Sample 1.e - Go to Application Properties, Change the INTER ARRIVAL TIME: 467(Micro Sec)

Sample 2.a: Open Sample 1.a: Right Click on Access Point Node and Select Properties, under Interface_Wireless, go to Physical Layer and change the Standard Channel as follows:-

1_2412 for **IEEE 802.11b** Standard in Access point 1

1_2412 for **IEEE 802.11b** Standard in Access point 2.

Sample 2.b - Go to Application Properties (Click on Application icon), Change the INTER ARRIVAL TIME: 1168(Micro Sec)

Sample 2.c - Go to Application Properties, Change the INTER ARRIVAL TIME: 778(Micro Sec)

Sample 2.d - Go to Application Properties, Change the INTER ARRIVAL TIME: 584(Micro Sec)

Sample 2.e - Go to Application Properties, Change the INTER ARRIVAL TIME: 467(Micro Sec)

9.3 Output

Comparison Table:

Different Channel:

Generation Rate(Mbps	Throughput(Mbps) App1	App2	Delay(Micro Sec) App1	App2	Packets Collided
5	4.98	4.98	1905.9	2025.5	0
10	5.93	5.94	1133184.1	1132943.8	0
15	5.93	5.94	1208669.6	1208199.5	0
20	5.93	5.94	1230876.6	1229952.0	0
25	5.93	5.94	1241121.7	1240439.4	0

Same Channel

Generation Rate(Mbps	Throughput(Mbps) App1	App2	Delay(Micro Sec) App1	App2	Packets Collided
5	3.08	3.07	1723757.0	1700489.6	352
10	3.07	3.06	2127449.7	2129169.2	374
15	3.07	3.06	2193574.1	2194286.2	374
20	3.07	3.06	2220030.9	2221263.1	374
25	3.07	3.06	2234271.6	2236143.3	374

9.4 Inference

As can be seen, when 802.11 b operate in the same channel there is co-channel interference and this leads to collisions between 801.11b packets. When 802.11b run in different channels there is no interference and hence no collisions. The throughput is much higher when the two APs operate in two different and non-overlapping channels.

10. Plot the characteristic curve of throughput versus offered traffic for a Pure and Slotted ALOHA system

Note: NetSim Academic supports a maximum of 100 nodes and hence this experiment can only be done partially with NetSim Academic. NetSim Standard/Pro would be required to simulate all the configurations.

10.1 Theory:

ALOHA provides a wireless data network. It is a multiple access protocol (this protocol is for allocating a multiple access channel). There are two main versions of ALOHA: pure and slotted. They differ with respect to whether or not time is divided up into discrete slots into which all frames must fit.

Pure ALOHA:

In pure Aloha, time is continuous. In Pure ALOHA, users transmit whenever they have data to be sent. There will be collisions and the colliding frames will be damaged. Senders need some way to find out if this is the case. If the frame was destroyed, the sender just waits a random amount of time and sends it again. The waiting time must be random or the same frames will collide over and over, in lockstep. Systems in which multiple users share a common channel in a way that can lead to conflicts are known as contention systems.

The probability of no other traffic being initiated during the entire vulnerable period is given by e^{-2G} which leads to S= $G * e^{-2G}$ where, S (frames per frame time) is the mean of the Poisson distribution with which frames are being generated. For reasonable throughput S should lie between 0 and 0.5.

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

The maximum throughput occurs at G = 0.5, with S = 1/2e, which is about 0.184. In other words, the best we can hope for is a channel utilization of18%. This result is not very encouraging, but with everyone transmitting at will, we could hardly have expected a 100% success rate.

Slotted ALOHA:

In slotted Aloha, time is divided up into discrete intervals, each interval corresponding to one frame. In Slotted ALOHA, a computer is required to wait for the beginning of the next slot in order to send the next packet. The probability of no other traffic being initiated during the entire vulnerable period is given by e^{-G} which leads to S = $G * e^{-G}$ where, S (frames per frame time) is the mean of the

Poisson distribution with which frames are being generated. For reasonable throughput S should lie between 0 and 1.

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

It is easy to note that Slotted ALOHA peaks at G=1, with a throughput of $S = {}^{1}/_{e}$ or about 0.368. It means that if the system is operating at G=1, the probability of an empty slot is 0.368

Calculations used in NetSim to obtain the plot between S and G:

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

$$G = \frac{Number\ of\ packet\ transmitted * Slot\ length(s)}{ST}$$

Where, G = Attempts per packet time

ST = Simulation time (in second)

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

$$S = \frac{Number\ of\ packet\ sucess * Slot\ length(s)}{ST}$$

Where, S = Throughput per packet time

ST = Simulation time (in second)

In the following experiment, we have taken packet size=1460 (Data Size) + 28 (Overheads) = 1488 bytes.

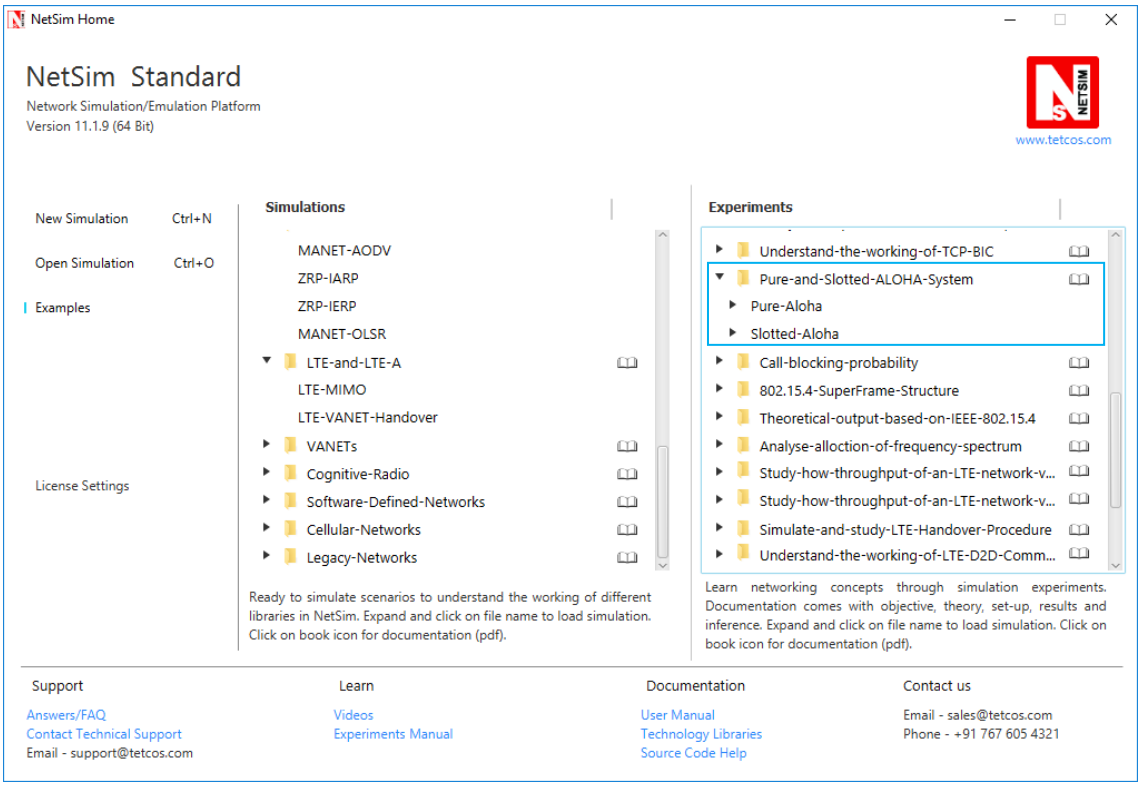
Bandwidth is 10 Mbps and hence, packet time comes as 1.2 milliseconds.

(Reference:** A good reference for this topic is Section 4.2.1: ALOHA, of the book, Computer Networking, 5th Edition by Tanenbaum and Wetherall**)

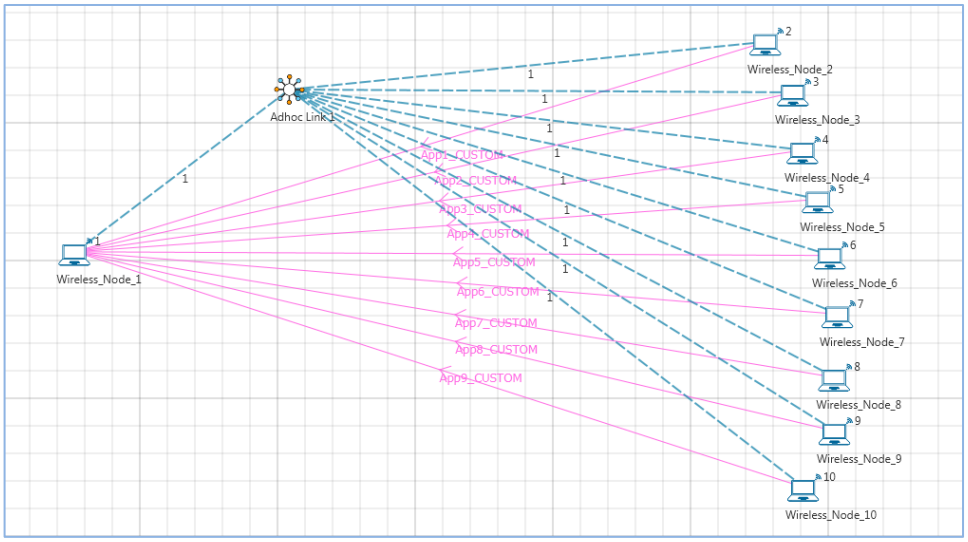
10.2 Network Set Up:

Part-1

In NetSim, Open Examples → Pure-and-Slotted-Aloha-Systems as shown below:



Click and drop 10 nodes and connect with Adhoc link as shown in the screenshot.



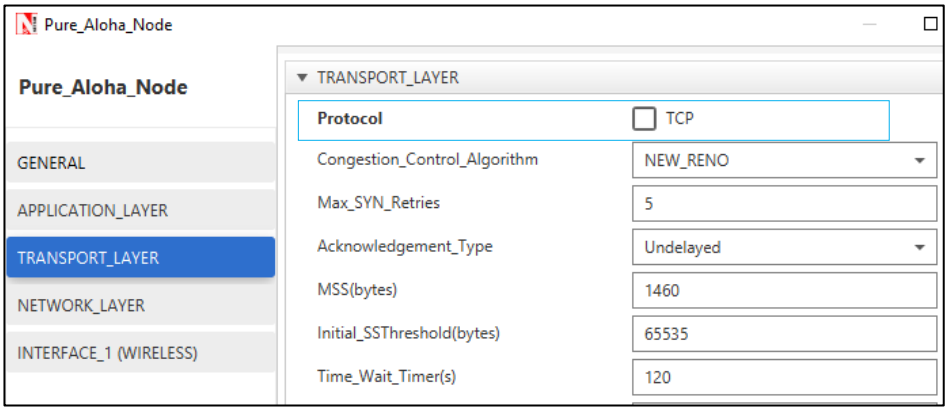
Sample Inputs:

Input for Sample 1: Drop 10 nodes (I.e. 9 Nodes are generating traffic.)

Node 2, 3, 4, 5, 6, 7, 8, 9, and 10 generates traffic. The properties of Node 2, 3, 4, 5, 6, 7, 8, 9, and 10 which transmits data to Node 1 are selected as follows

Wireless Node Properties:

Wireless Node Properties	
Transport Layer	
TCP	**disable**
Interface1_Wireless (PHYSICAL_LAYER)	
Slot Length(µs)	**1200**
Data Rate(mbps)	**10**
Interface1_Wireless (DATALINK_LAYER)	
ARP_Retry_Limit	**0**
Ismac_Buffer	**FALSE**



Right click on the Adhoc link and select the channel characteristics as **no path loss**.

Application Properties: Right click on the Application icon and set following properties as shown in below figure:

Application_1 Properties		
Application Method	**Unicast**	
Application Type	**Custom**	
Source_Id	**2**	
Destination_Id	**1**	
Packet Size	**Distribution**	**Constant**
	Value (bytes)	**1460**
Inter Arrival Time	**Distribution**	**Exponential**
	Packet Inter Arrival Time (µs)	**200000**

Similarly create 8 more application, i.e. Source_Id as 3, 4, 5, 6, 7, 8, 9 and Destination_Id as 1, set Packet Size and Inter Arrival Time as shown in above table.

Simulation Time- 10 Seconds

Note: The Simulation Time can be selected only after doing the following two tasks,

- ***Set the properties of Nodes***
- ***Then click on Run Simulation button***
- ***Obtain the values of Total Number of Packets Transmitted and Collided from the results window of NetSim***

Input for Sample2: Drop 20 nodes (i.e. 19 Nodes are generating traffic.)

Nodes 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20 transmit data to Node 1.

Continue the experiment by increasing the number of nodes generating traffic as 29, 39, 49, 59, 69, 79, 89, 99, 109, 119, 129, 139, 149, 159, 169, 179, 189 and 199 nodes.

Part-2

Slotted ALOHA:

Input for Sample1: Drop 20 nodes (i.e. 19 Nodes are generating traffic.)

Nodes 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, and 20 transmit data to Node 1 and set properties for nodes and application as mentioned above.

Continue the experiment by increasing the number of nodes generating traffic as 39, 59, 79, 99, 119, 139, 159, 179, 199, 219, 239, 259, 279, 299, 319, 339, 359, 379, and 399 nodes.

10.3 Output:

Comparison Table: The values of Total Number of Packets Transmitted and Collided obtained from the network statistics after running NetSim simulation are provided in the table below along with Throughput per packet time& Number of Packets Transmitted per packet time

Pure Aloha:

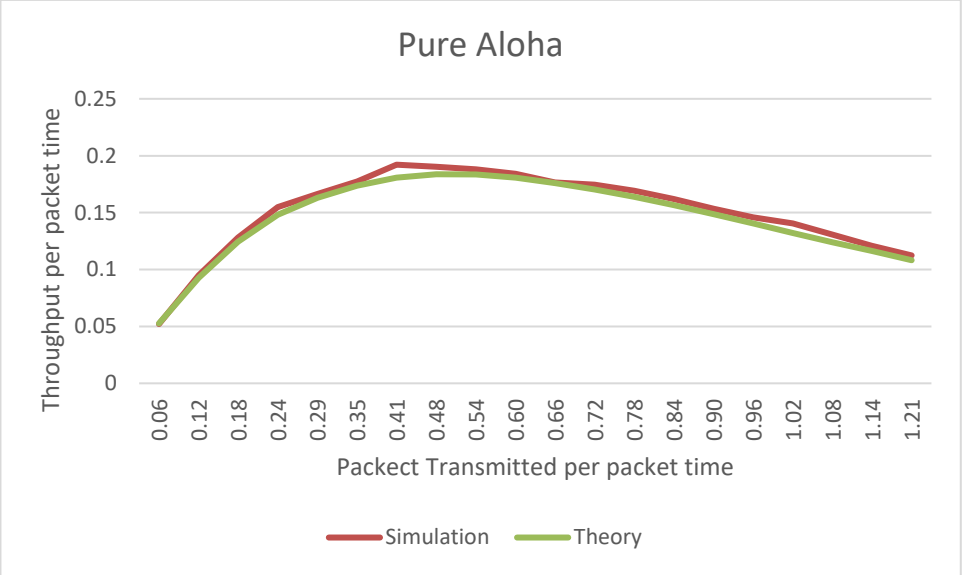
Number of nodes generating traffic	Total number of Packets Transmitted	Total number of Packets Collided	Total number of Packets Success (Packets Transmitted -Packets Collided)	Throughput per packet time(G)	Number of Packets Transmitted per packet time(S)	Packets per packet time theoretical ($S = G * e^{-2G}$)
9	494	60	434	0.05928	0.05208	0.05265
19	978	187	791	0.11736	0.09492	0.09281
29	1482	415	1067	0.17784	0.12804	0.12461
39	1991	700	1291	0.23892	0.15492	0.14816
49	2443	1056	1387	0.29316	0.16644	0.16311
59	2907	1429	1478	0.34884	0.17736	0.17363

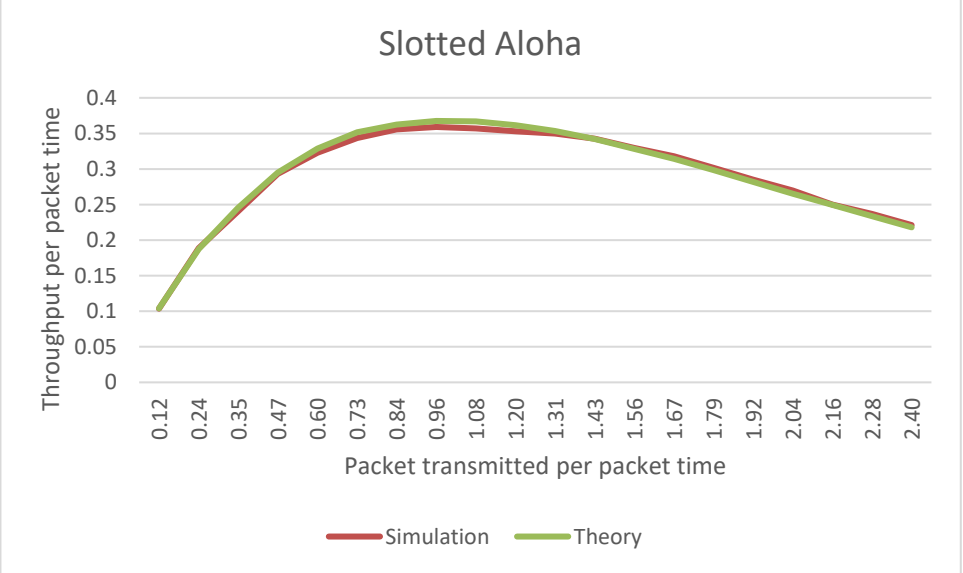
69	3435	1834	1601	0.4122	0.19212	0.18075
79	3964	2377	1587	0.47568	0.19044	0.18371
89	4468	2900	1568	0.53616	0.18816	0.18348
99	4998	3464	1534	0.59976	0.18408	0.18073
109	5541	4071	1470	0.66492	0.1764	0.17588
119	6029	4575	1454	0.72348	0.17448	0.17022
129	6512	5103	1409	0.78144	0.16908	0.16374
139	7005	5657	1348	0.8406	0.16176	0.15648
149	7499	6221	1278	0.89988	0.15336	0.14878
159	8020	6804	1216	0.9624	0.14592	0.14042
169	8531	7361	1170	1.02372	0.1404	0.13213
179	9033	7945	1088	1.08396	0.13056	0.12402
189	9512	8505	1007	1.14144	0.12084	0.11642
199	10053	9117	936	1.20636	0.11232	0.10806

Slotted Aloha:

Number of nodes generating traffic	Total number of Packets Transmitted	Total number of Packets Collided	Total number of Packets Success (Packets Transmitted -Packets Collided)	Throughput per packet time(G)	Number of Packets Transmitted per packet time(S)	Packets per packet time theoretical ($S = G * e^{-G}$)
19	974	111	863	0.11688	0.10356	0.103987
39	1981	407	1574	0.23772	0.18888	0.187424
59	2893	891	2002	0.34716	0.24024	0.245335
79	3946	1504	2442	0.47352	0.29304	0.294911
99	4976	2286	2690	0.59712	0.3228	0.328652
119	6059	3197	2862	0.72708	0.34344	0.351411
139	6961	3999	2962	0.83532	0.35544	0.362308
159	7971	4979	2992	0.95652	0.35904	0.367521
179	8969	5994	2975	1.07628	0.357	0.366862
199	9983	7042	2941	1.19796	0.35292	0.361555
219	10926	8011	2915	1.31112	0.3498	0.35337
239	11928	9073	2855	1.43136	0.3426	0.342072
259	12969	10224	2745	1.55628	0.3294	0.328249

279	13916	11266	2650	1.66992	0.318	0.314383
299	14945	12430	2515	1.7934	0.3018	0.29841
319	15967	13592	2375	1.91604	0.285	0.282019
339	17011	14765	2246	2.04132	0.26952	0.26508
359	17977	15895	2082	2.15724	0.24984	0.249472
379	18983	17010	1973	2.27796	0.23676	0.233475
399	19987	18146	1841	2.39844	0.22092	0.217921



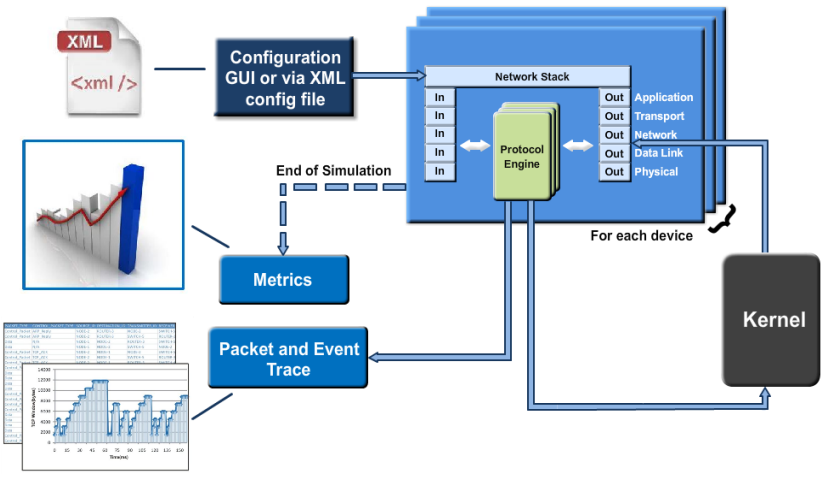


Thus the following characteristic plot for the Pure ALOHA and Slotted ALOHA is obtained, which matches the theoretical result.

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

11.1 Theory

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



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

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

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

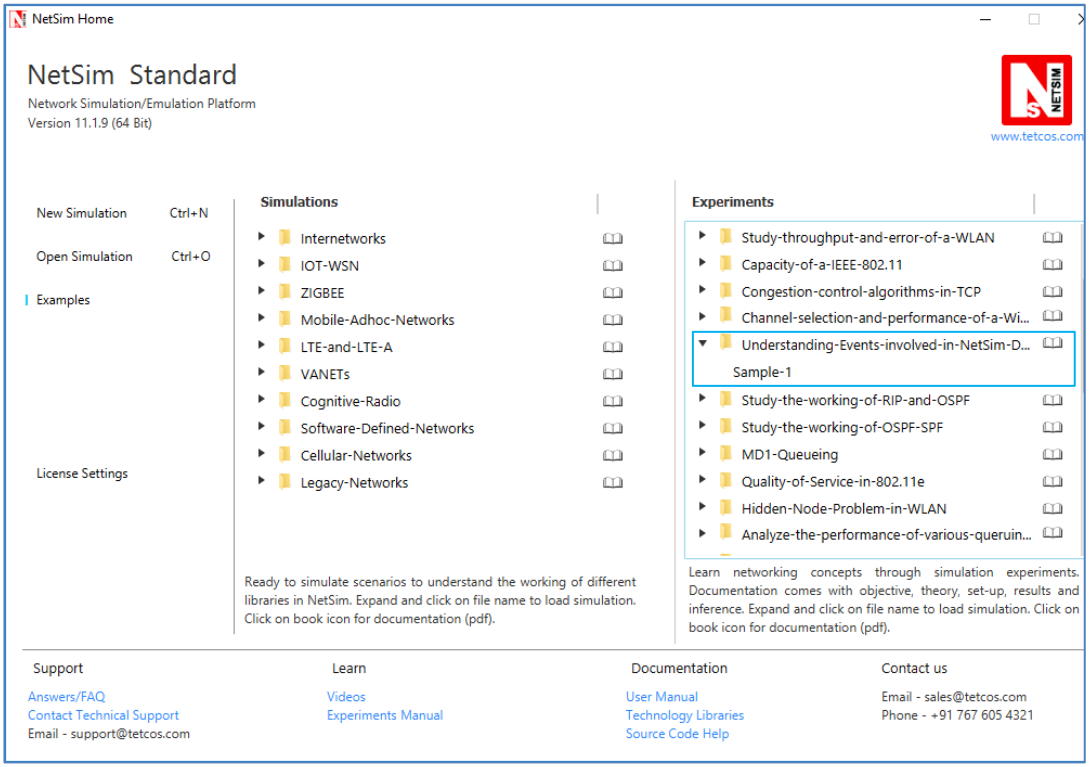
Each event gets added in the Simulation kernel by the protocol operating at the particular layer of the Network Stack. The required sub events are passed into the Simulation kernel. These sub events are then fetched by the Network Stack in order to execute the functionality of each protocol. At the end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

Event Trace:

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

11.2 Procedure:

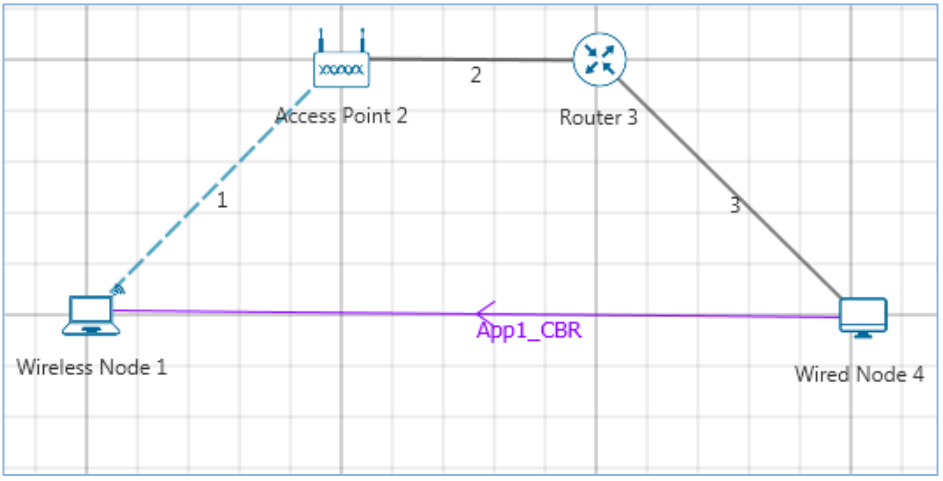
Open Examples → Understanding-Events-involved-in-NetSim-DES as shown below:



Follow the steps given in order to arrive at the objective.

- Total no of APs (Access Points) used: 1
- Total no of Wireless Nodes used: 1
- Total no of Routers used: 1
- Total no of Wired Nodes used: 1

The Wireless Node, AP, Router and Wired Node are interconnected as shown:



Also edit the following properties of Wireless Node 4, AP 2, Router 3 and Wireless Node 1:

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

Edit link properties as shown:

Wireless Link Properties	
Channel Characteristics	No path loss

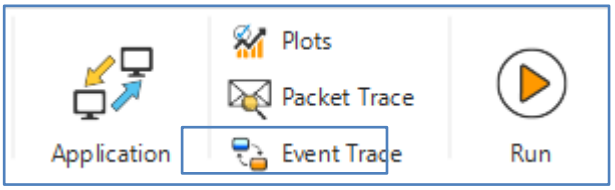
Disable TCP in Wireless Node 1, Router 3 and Wired Node 4.

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

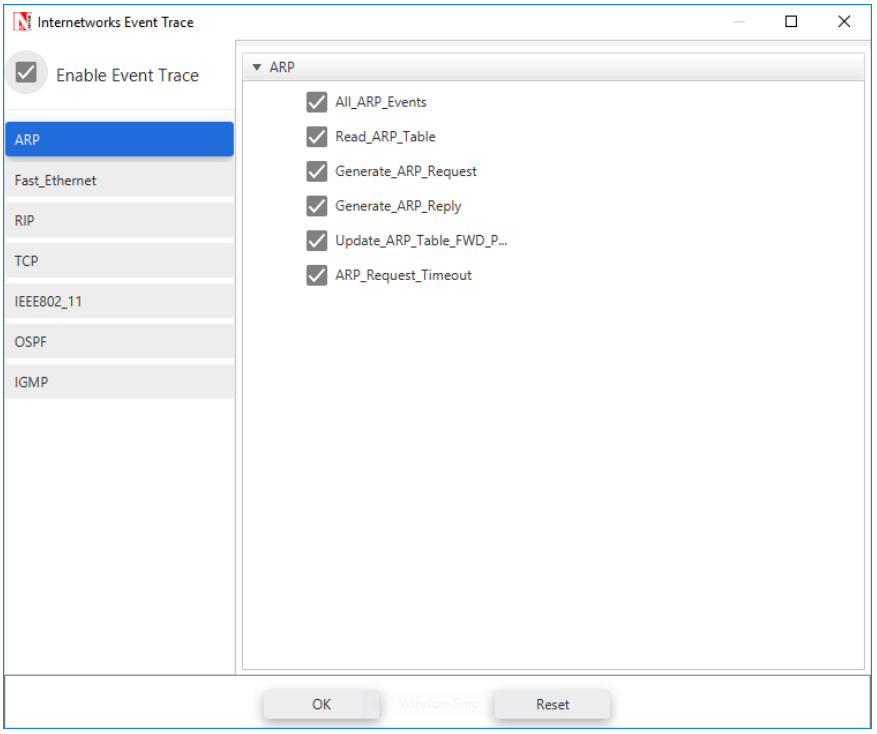
Application Properties	
Application Type	**CBR**
Source_Id	**4**
Destination_Id	**1**
Packet Size	
Distribution	**Constant**
Value (bytes)	**1460**
Packet Inter Arrival Time	
Distribution	**Constant**
Value (micro sec)	**20000**

Enable Event Trace:

Event trace can be turned on by clicking the Event Trace icon in the tool bar and select the Event Trace check box and click on OK.



Further users can select the required fields in the event trace. In this case let us leave it to default settings.



Simulation Time - 10 Sec

(Note: The Simulation Time can be selected only after the following two tasks,

- ***Set the properties for all the devices and links.***
- ***Click on Run Simulation button.***

Upon completion of the experiment, Save it using the save button (or ctrl + s) in the current workspace.

11.3 Output

Open the event trace by clicking on the Open Event Trace link from the Simulation Results window,

An Event trace file similar to the following opens in Excel as shown below:

Event_Id	Event_Type	Event_Time(US)	Device_Type	Device_Id	Interface_Id	Application_Id	Packet_Id	Segment_Id	Protocol_Name	Subevent_Type	Packet_Size(	Prev_Event_Id	Line_No	File_Name
1	TIMER_EVENT	0	NODE	1	0	0	0	0	IPV4	IP_INIT_TABLE	0	0	1270	IP.c
2	TIMER_EVENT	0	ROUTER	3	0	0	0	0	IPV4	IP_INIT_TABLE	0	0	1270	IP.c
3	TIMER_EVENT	0	NODE	4	0	0	0	0	IPV4	IP_INIT_TABLE	0	0	1270	IP.c
4	TIMER_EVENT	0	ACCESSPOINT	2	2	0	0	0	ETHERNET	ETH_IF_UP	0	0	623	libEthernet.c
5	TIMER_EVENT	0	ROUTER	3	1	0	0	0	ETHERNET	ETH_IF_UP	0	0	623	libEthernet.c
6	TIMER_EVENT	0	ROUTER	3	2	0	0	0	ETHERNET	ETH_IF_UP	0	0	623	libEthernet.c
7	TIMER_EVENT	0	NODE	4	1	0	0	0	ETHERNET	ETH_IF_UP	0	0	623	libEthernet.c
8	TIMER_EVENT	0	NODE	4	0	1	1	0	APPLICATION		1460	0	141	Database_FTP_Custom
9	APPLICATION_OUT	0	NODE	4	0	1	1	0	APPLICATION	0	1460	8	241	Application.c
10	TRANSPORT_OUT	0	NODE	4	0	1	1	0	UDP	0	1460	9	101	Application.c
12	NETWORK_OUT	0	NODE	4	0	1	1	0	IPV4	0	1468	10	87	Send_User_Datagram.c
13	MAC_OUT	0	NODE	4	1	1	1	0	ETHERNET	0	1488	12	104	ReadArpTable.c
14	PHYSICAL_OUT	0	NODE	4	1	1	1	0	ETHERNET	0	1514	13	49	Ethernet_Mac.c
15	PHYSICAL_IN	127.08	ROUTER	3	2	1	1	0	ETHERNET	0	1514	14	98	Ethernet_Phy.c
16	MAC_IN	127.08	ROUTER	3	2	1	1	0	ETHERNET	0	1514	15	192	Ethernet_Phy.c
17	NETWORK_IN	127.08	ROUTER	3	2	1	1	0	IPV4	0	1488	16	203	Ethernet_Mac.c
18	NETWORK_OUT	127.08	ROUTER	3	2	1	1	0	IPV4	0	1468	17	446	IP.c
19	MAC_OUT	127.08	ROUTER	3	1	1	1	0	ETHERNET	0	1488	18	104	ReadArpTable.c
20	PHYSICAL_OUT	127.08	ROUTER	3	1	1	1	0	ETHERNET	0	1514	19	49	Ethernet_Mac.c
21	PHYSICAL_IN	253.2	ACCESSPOINT	2	2	1	1	0	ETHERNET	0	1514	20	98	Ethernet_Phy.c
22	MAC_IN	253.2	ACCESSPOINT	2	2	1	1	0	ETHERNET	0	1514	21	192	Ethernet_Phy.c
23	MAC_OUT	253.2	ACCESSPOINT	2	1	1	1	0	WLAN	0	1488	22	237	Ethernet_Mac.c
24	MAC_OUT	253.2	ACCESSPOINT	2	1	1	1	0	WLAN	CS	1488	23	101	CSMACA.c
25	MAC_OUT	303.2	ACCESSPOINT	2	1	1	1	0	WLAN	IEEE802_11_EVENT	1488	24	132	CSMACA.c

We start from the **APPLICATION_OUT** event of the first packet, which happens in the Wired Node and end with the **MAC_IN** event of the **WLAN_ACK** packet which reaches the Wired Node.

Events in the event trace are logged with respect to the time of occurrence due to which, event id may not be in order.

11.3.1 Events Involved:

Events are listed in the following format:

[EVENT_TYPE, EVENT_TIME, PROTOCOL, EVENT_NO, SUBEVENT_TYPE]

[APP_OUT, 20000, APP, 6, -]

[TRNS_OUT, 20000, UDP, 7, -]

[NW_OUT, 20000, IPV4, 9, -]

[MAC_OUT, 20000, ETH, 10, -]

[MAC_OUT, 20000, ETH, 11, CS]

[MAC_OUT, 20000.96, ETH, 12, IFG]

[PHY_OUT, 20000.96, ETH, 13, -]

[PHY_OUT, 20122.08, ETH, 14, PHY_SENSE]

[PHY_IN, 20127.08, ETH, 15, -]

[MAC_IN, 20127.08, ETH, 16, -]

[NW_IN, 20127.08, IPV4, 17, -]

[NW_OUT, 20127.08, IPV4, 18, -]

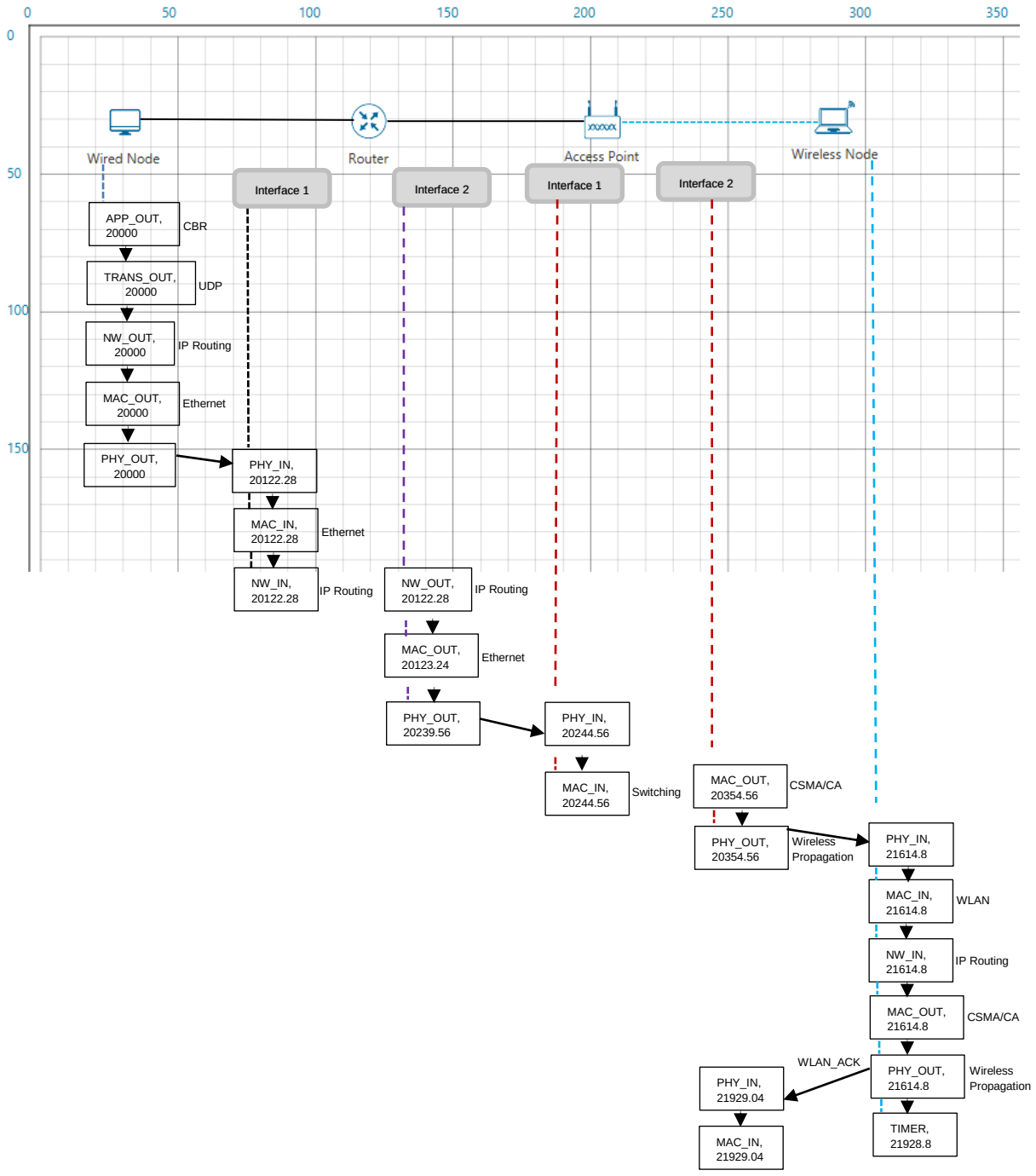
[MAC_OUT,	20127.08,	ETH,	19,	-]
[MAC_OUT,	20127.08,	ETH,	20,	CS]
[MAC_OUT,	20128.04,	ETH,	21,	IFG]
[PHY_OUT,	20128.04,	ETH,	22,	-]
[PHY_OUT,	20249.16,	ETH,	23,	PHY_SENSE]
[PHY_IN,	20254.16,	ETH,	24,	-]
[MAC_IN,	20254.16,	ETH,	25,	-]
[MAC_OUT,	20254.16,	WLAN,	26,	-]
[MAC_OUT,	20254.16,	WLAN,	27,	DIFS_END]
[MAC_OUT,	20304.16,	WLAN,	28,	BACKOFF]
[MAC_OUT,	20324.16,	WLAN,	29,	BACKOFF]
[MAC_OUT,	20344.16,	WLAN,	30,	BACKOFF]
[MAC_OUT,	20364.16,	WLAN,	31,	BACKOFF]
[PHY_OUT,	20364.16,	WLAN,	32,	-]
[TIMER,	21668.16,	WLAN,	35,	UPDATE_DEVICE_STATUS]
[PHY_IN,	21668.4,	WLAN,	33,	-]
[MAC_IN,	21668.4,	WLAN,	36,	RECEIVE_MPDU]
[NW_IN,	21668.4,	IPV4,	37,	-]
[MAC_OUT,	21668.4,	WLAN,	38,	SEND_ACK]
[TRNS_IN,	21668.4,	UDP,	39,	-]
[APP_IN,	21668.4,	APP,	41,	-]
[PHY_OUT,	21678.4,	WLAN,	40,	-]
[TIMER,	21982.4,	WLAN,	43,	UPDATE_DEVICE]

[PHY_IN, 21982.63, WLAN, 42, -]

[MAC_IN, 21982.63, WLAN, 44, RECEIVE_ACK]

[TIMER, 21985, WLAN, 34, ACK_TIMEOUT]

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

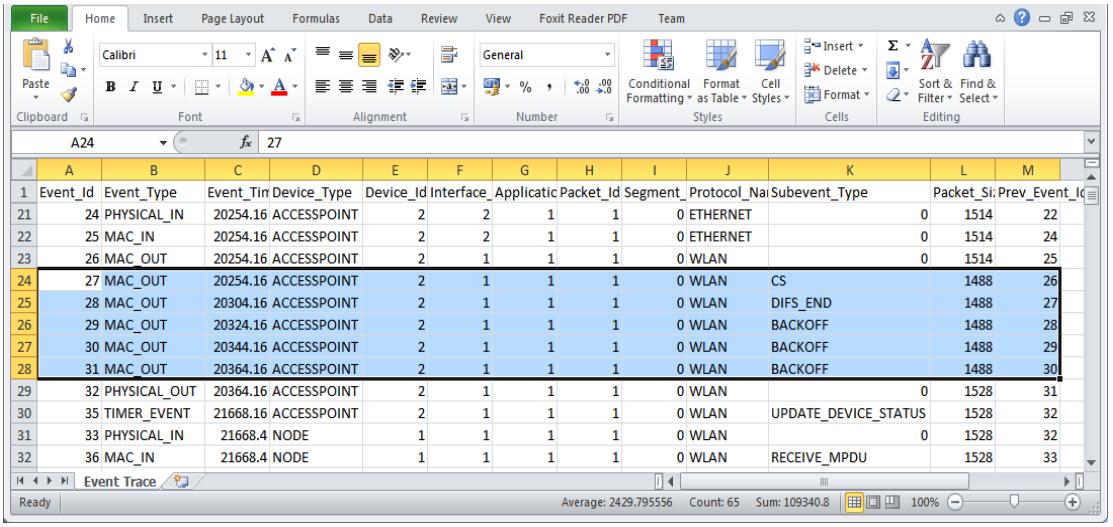


For Example:

MAC_OUT in the Access Point involves sub events like CS, DIFS_END and BACKOFF.

As you can see in the trace file shown below, CS happens at event time 20254.16,

Adding DIFS time of 50µs to this will give DIFS_END sub event at 20304.16. Further it is followed by three Backoff's each of 20 µs, at event time 20314.16, 20324.16, 20344.16 respectively.



	Event_Id	Event_Type	Event_Tim	Device_Type	Device_Id	Interface_	Applicatic	Packet_Id	Segment_	Protocol_Na	Subevent_Type	Packet_Si	Prev_Event_Id
21	24	PHYSICAL_IN	20254.16	ACCESSPOINT	2	2	1	1	0	ETHERNET	0	1514	22
22	25	MAC_IN	20254.16	ACCESSPOINT	2	2	1	1	0	ETHERNET	0	1514	24
23	26	MAC_OUT	20254.16	ACCESSPOINT	2	1	1	1	0	WLAN	0	1514	25
24	27	MAC_OUT	20254.16	ACCESSPOINT	2	1	1	1	0	WLAN	CS	1488	26
25	28	MAC_OUT	20304.16	ACCESSPOINT	2	1	1	1	0	WLAN	DIFS_END	1488	27
26	29	MAC_OUT	20324.16	ACCESSPOINT	2	1	1	1	0	WLAN	BACKOFF	1488	28
27	30	MAC_OUT	20344.16	ACCESSPOINT	2	1	1	1	0	WLAN	BACKOFF	1488	29
28	31	MAC_OUT	20364.16	ACCESSPOINT	2	1	1	1	0	WLAN	BACKOFF	1488	30
29	32	PHYSICAL_OUT	20364.16	ACCESSPOINT	2	1	1	1	0	WLAN	0	1528	31
30	35	TIMER_EVENT	21668.16	ACCESSPOINT	2	1	1	1	0	WLAN	UPDATE_DEVICE_STATUS	1528	32
31	33	PHYSICAL_IN	21668.4	NODE	1	1	1	1	0	WLAN	0	1528	32
32	36	MAC_IN	21668.4	NODE	1	1	1	1	0	WLAN	RECEIVE_MPDU	1528	33

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

11.4 Inference

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

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

Understanding NetSim's Event trace and its flow is very much helpful especially when customizing existing code and debugging to verify the correctness the modified code. The event IDs provided in the event trace can be used to go to a specific event while debugging.

12. Study the working and routing table formation of Interior routing protocols, i.e. Routing Information Protocol (RIP) and Open Shortest Path First (OSPF)

12.1 Introduction

RIP

RIP is intended to allow hosts and gateways to exchange information for computing routes through an IP-based network. RIP is a distance vector protocol which is based on Bellman-Ford algorithm. This algorithm has been used for routing computation in the network.

Distance vector algorithms are based on the exchange of only a small amount of information using RIP messages.

Each entity (router or host) that participates in the routing protocol is assumed to keep information about all of the destinations within the system. Generally, information about all entities connected to one network is summarized by a single entry, which describes the route to all destinations on that network. This summarization is possible because as far as IP is concerned, routing within a network is invisible. Each entry in this routing database includes the next router to which datagram's destined for the entity should be sent. In addition, it includes a "metric" measuring the total distance to the entity.

Distance is a somewhat generalized concept, which may cover the time delay in getting messages to the entity, the dollar cost of sending messages to it, etc. Distance vector algorithms get their name from the fact that it is possible to compute optimal routes when the only information exchanged is the list of these distances. Furthermore, information is only exchanged among entities that are adjacent, that is, entities that share a common network.

OSPF

In OSPF, the Packets are transmitted through the shortest path between the source and destination.

Shortest path: OSPF allows administrator to assign a cost for passing through a link. The total cost of a particular route is equal to the sum of the costs of all links that comprise the route. A router chooses the route with the shortest (smallest) cost.

In OSPF, each router has a link state database which is tabular representation of the topology of the network (including cost). Using Dijkstra algorithm each router finds the shortest path between source and destination.

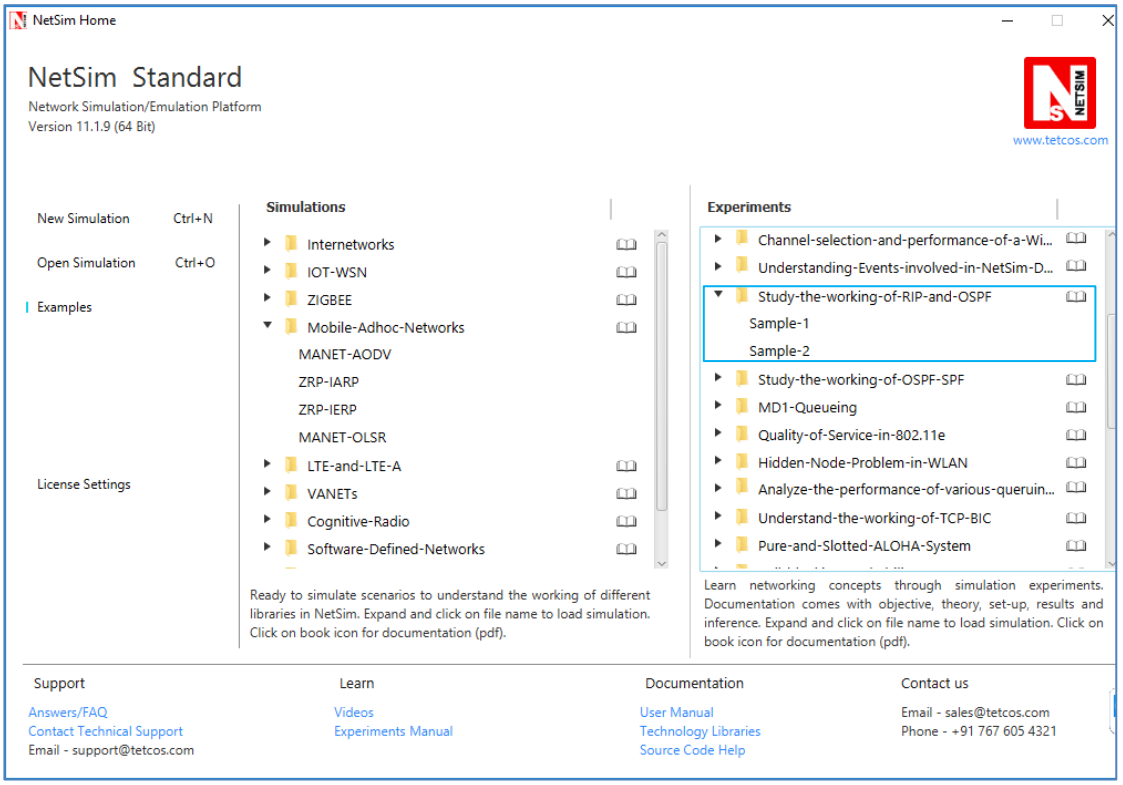
Formation of OSPF Routing Table

1. OSPF-speaking routers send Hello packets out all OSPF-enabled interfaces. If two routers sharing a common data link agree on certain parameters specified in their respective Hello packets, they will become neighbors.
2. Adjacencies, which can be thought of as virtual point-to-point links, are formed between some neighbors. OSPF defines several network types and several router types. The establishment of an adjacency is determined by the types of routers exchanging Hellos and the type of network over which the Hellos are exchanged.
3. Each router sends link-state advertisements (LSAs) over all adjacencies. The LSAs describe all of the router's links, or interfaces, the router's neighbors, and the state of the links. These links might be to stub networks (networks with no other router attached), to other OSPF routers, or to external networks (networks learned from another routing process). Because of the varying types of link-state information, OSPF defines multiple LSA types.
4. Each router receiving an LSA from a neighbor records the LSA in its link-state database and sends a copy of the LSA to all of its other neighbors.
5. By flooding LSAs throughout an area, all routers will build identical link-state databases.
6. When the databases are complete, each router uses the SPF algorithm to calculate a loop-free graph describing the shortest (lowest cost) path to every known destination, with itself as the root. This graph is the SPF tree.
7. Each router builds its route table from its SPF tree

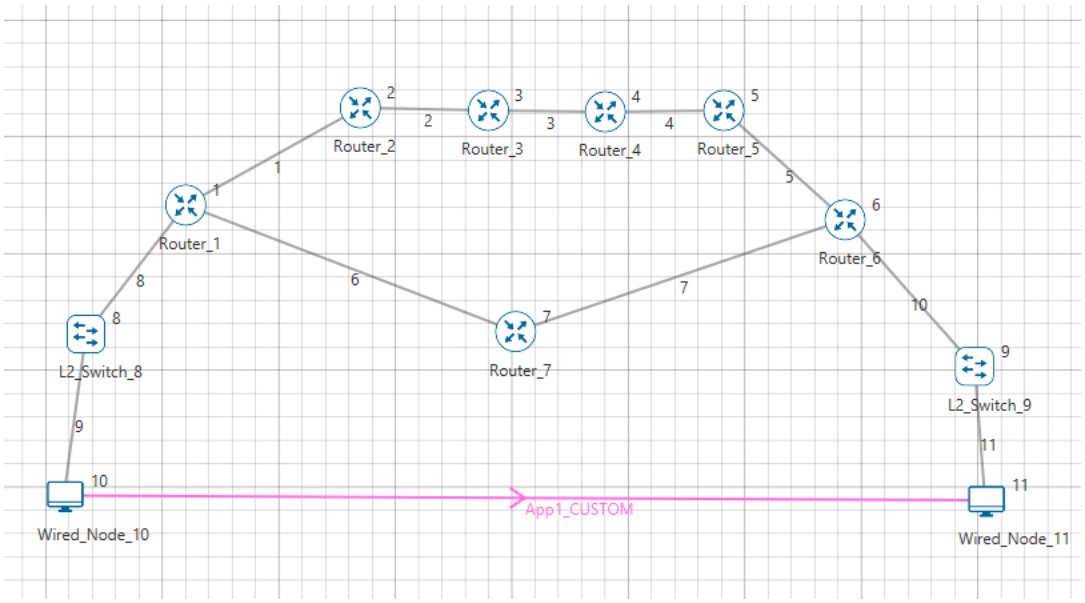
12.2 Procedure

Sample 1:

Step 1: Open Examples → Study-the-working-of-RIP-and-OSPF as shown below:

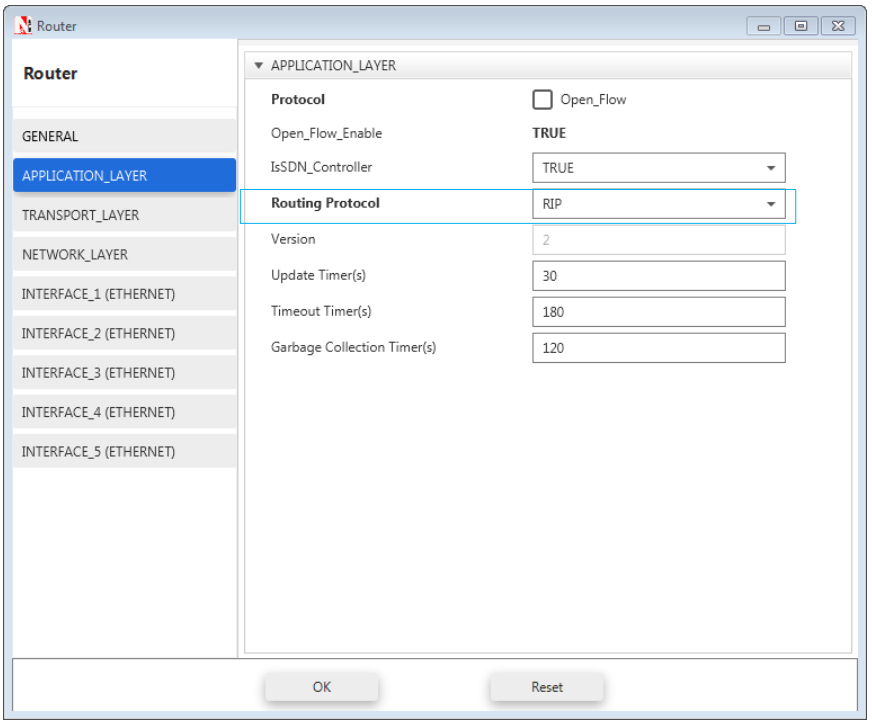


Step 2: Click & drop Routers, Switches and Nodes onto the Simulation Environment and link them as shown:



Step 3: These properties can be set only after devices are linked to each other as shown above.

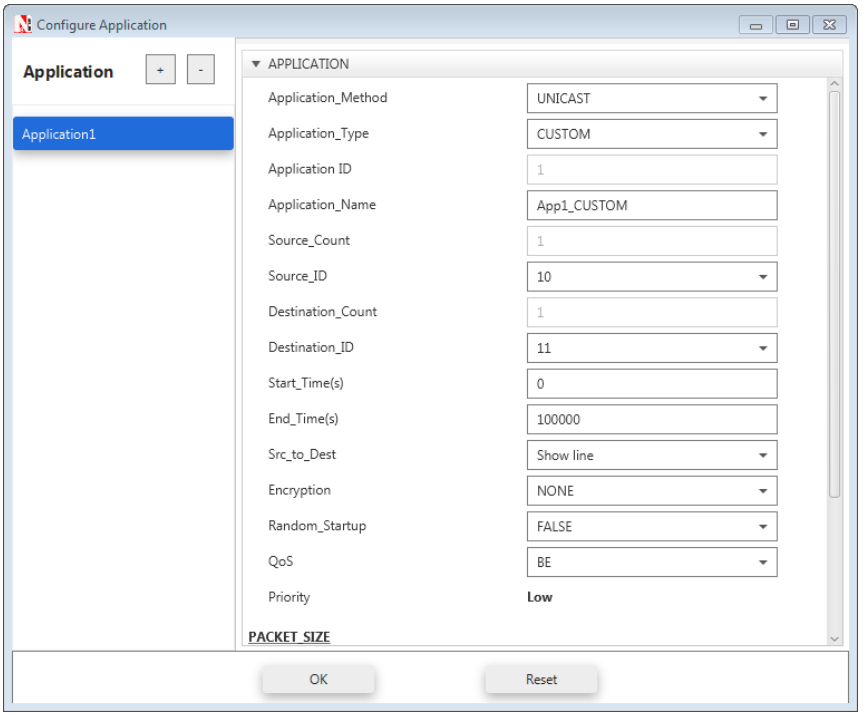
Set the properties of the Router 1 as follows:



Node Properties: In Wired Node 10, go to Transport Layer and set TCP as Disable

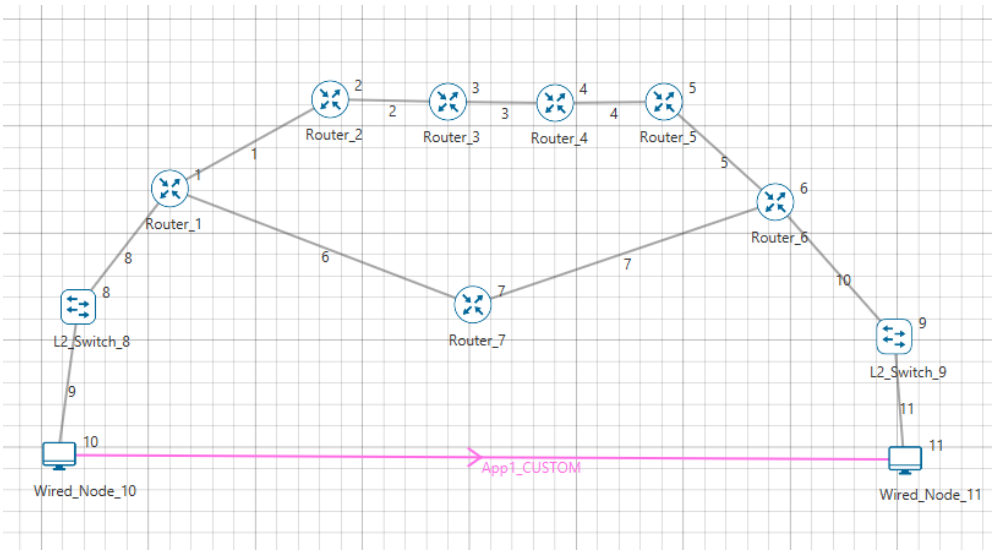
Switch Properties: Accept default properties for Switch.

Application Properties: Click on the Application icon and set properties as follows:

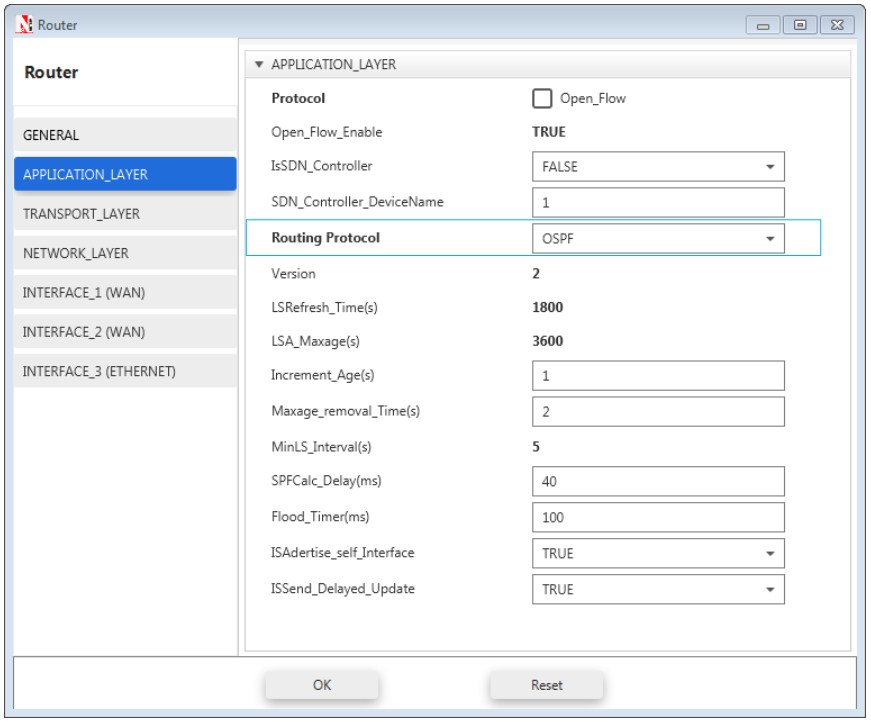


Enable packet trace and run simulation for 100s. After Simulation is performed, save the experiment.

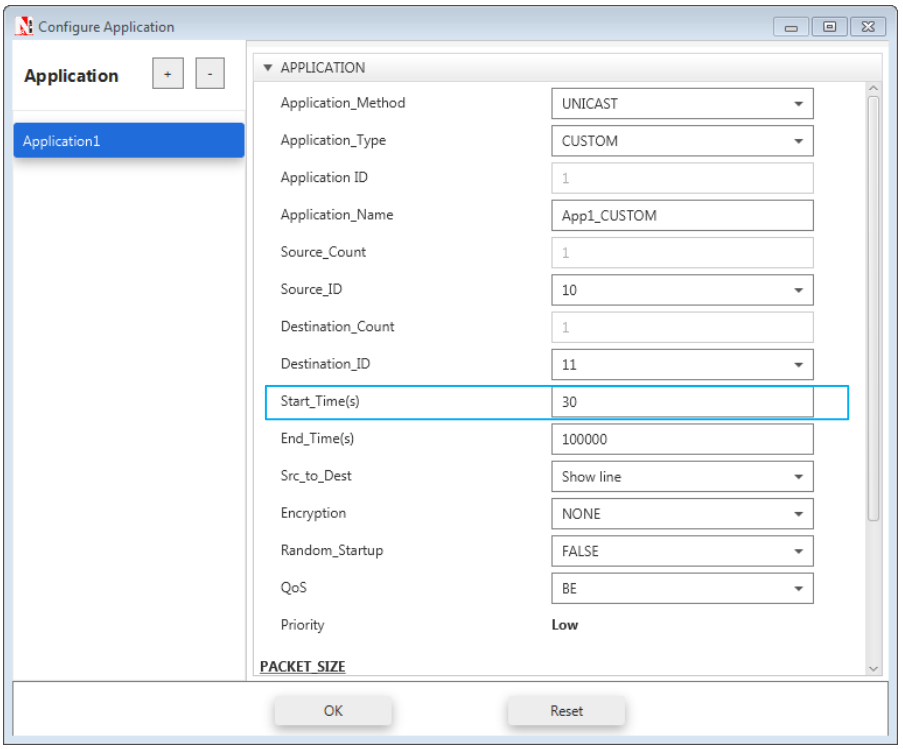
Sample 2: To model a scenario, follow the same steps given below:



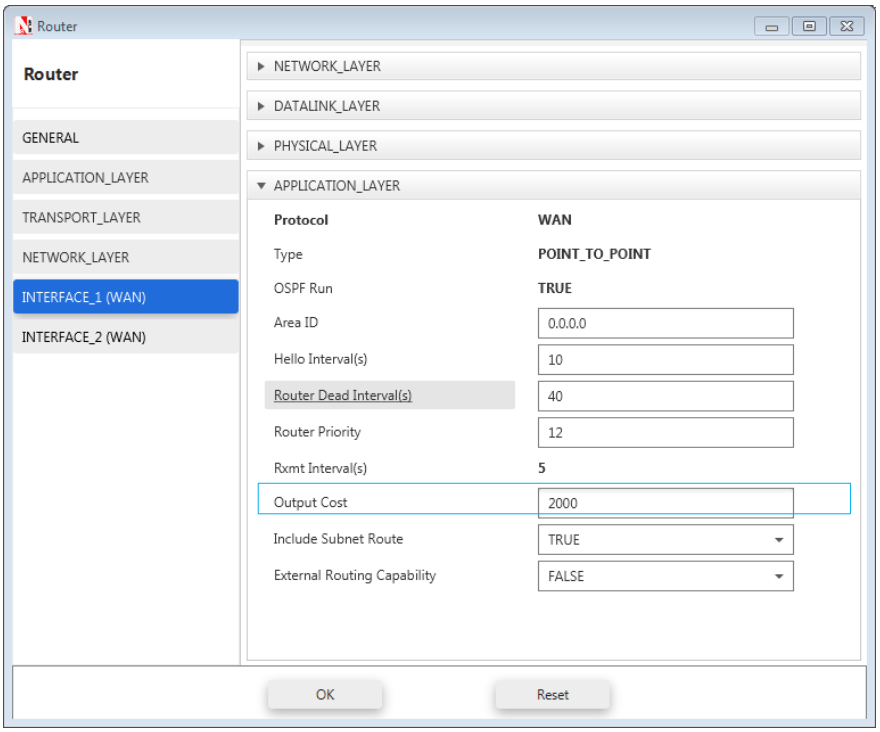
Right click on router and set Routing protocol as OSPF



Application Properties: Click on the Application icon and set properties as follows



Right click on Router 7 and go to properties. In the WAN Interfaces – INTERFACE_1 (WAN), INTERFACE_2 (WAN) set the Output Cost as 2000 as shown below:



Note that in a wired network with routers and switches OSPF, Spanning tree etc. takes times to converge and hence it is a good practice to set the application start time greater than OSPF convergence time. Convergence time increases as the size of the network grows.

Simulation Time- 100 Sec

12.3 Output and Inference:

RIP

In Distance vector routing, each router periodically shares its knowledge about the entire network with its neighbors. The three keys for understanding the algorithm,

1. Knowledge about the whole network

Router sends all of its collected knowledge about the network to its neighbors

2. Routing only to neighbors

Each router periodically sends its knowledge about the network only to those routers to which it has direct links. It sends whatever knowledge it has about the whole network through all of its ports. This information is received and kept by each neighboring router and used to update that router's own information about the network.

3. Information sharing at regular intervals

For example, every 30 seconds, each router sends its information about the whole network to its neighbors. This sharing occurs whether or not the network has changed since the last time information was exchanged

In NetSim the Routing table Formation has 3 stages

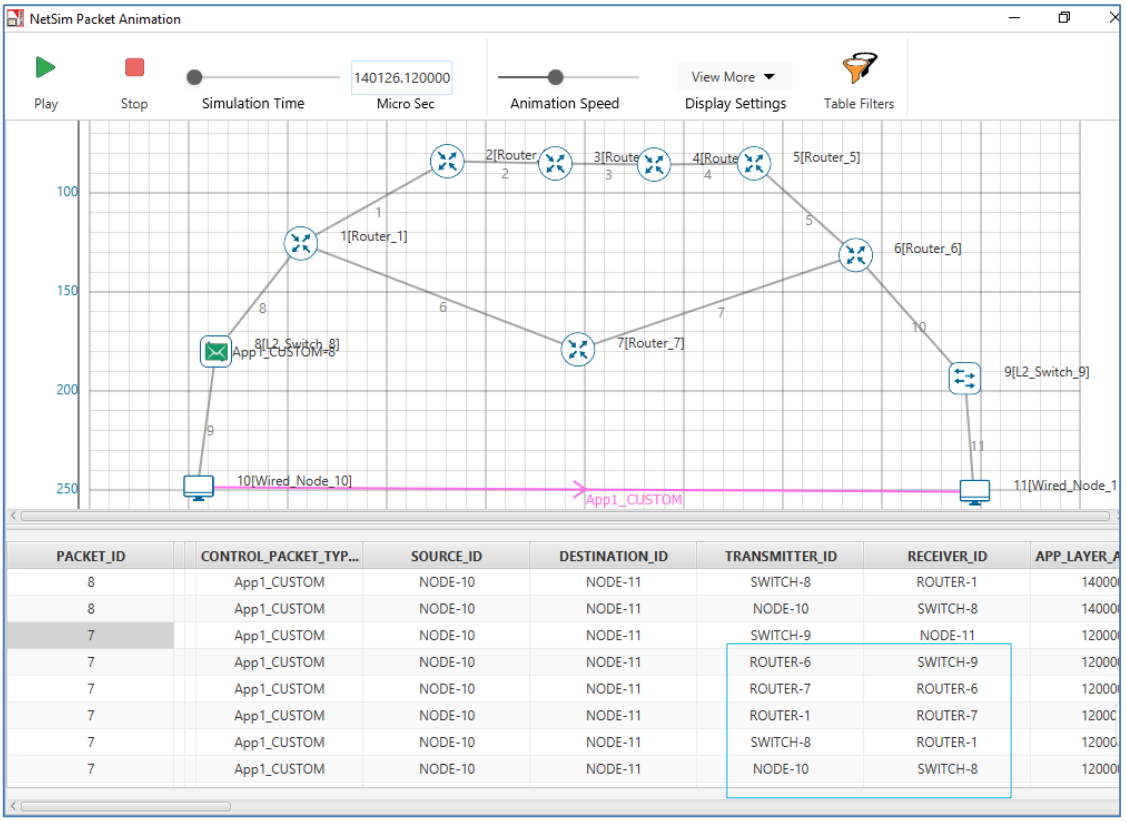
Initial Table: This table will show the direct connections made by each Router.

Intermediate Table: The Intermediate table will have the updates of the Network in every 30 seconds

Final Table: This table is formed when there is no update in the Network.

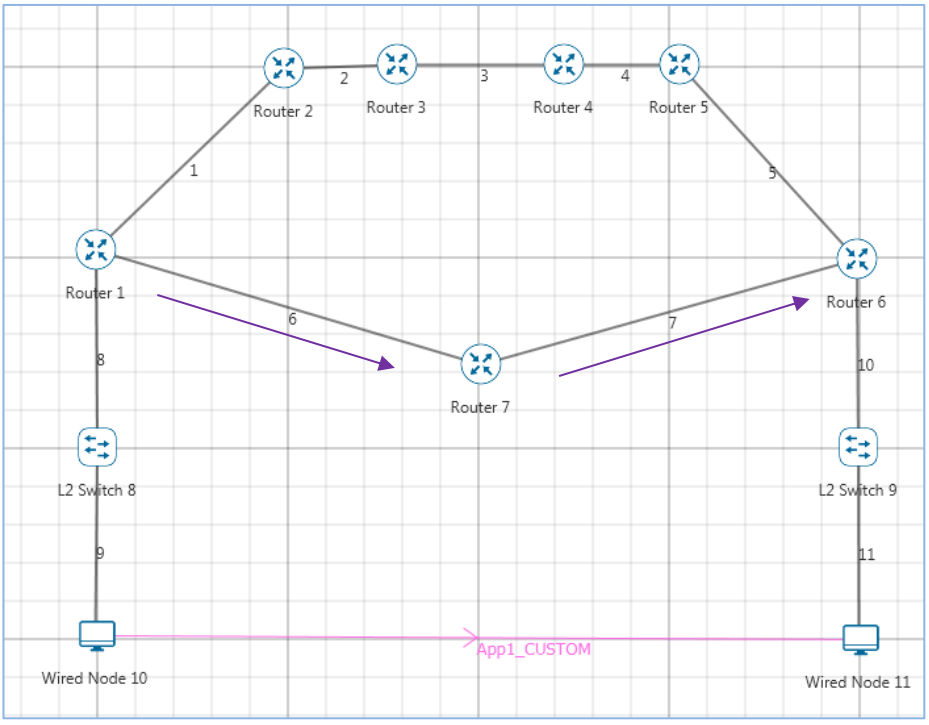
The data should be forwarded using Routing Table with the shortest distance.

Go to NetSim Packet Animation window and play animation. You will be able to see the path packets take both in the animation and in the packet trace table as shown below:



PACKET_ID	CONTROL_PACKET_TYP...	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_A
8	App1_CUSTOM	NODE-10	NODE-11	SWITCH-8	ROUTER-1	14000
8	App1_CUSTOM	NODE-10	NODE-11	NODE-10	SWITCH-8	14000
7	App1_CUSTOM	NODE-10	NODE-11	SWITCH-9	NODE-11	12000
7	App1_CUSTOM	NODE-10	NODE-11	ROUTER-6	SWITCH-9	12000
7	App1_CUSTOM	NODE-10	NODE-11	ROUTER-7	ROUTER-6	12000
7	App1_CUSTOM	NODE-10	NODE-11	ROUTER-1	ROUTER-7	12000
7	App1_CUSTOM	NODE-10	NODE-11	SWITCH-8	ROUTER-1	12000
7	App1_CUSTOM	NODE-10	NODE-11	NODE-10	SWITCH-8	12000

- Shortest Path from Wired Node 10 to Wired Node 11 in RIP (Use Packet Animation to view) is **Wired Node 10->L2 Switch 8->Router 1->Router 7->Router 6->L2 Switch 9->Wired Node 11**. RIP chooses the lower path (number of hops is less) to forward packets from source to destination since it is based on hop count as shown below



OSPF

The main operation of the OSPF protocol occurs in the following consecutive stages and leads to the convergence of the internetworks:

1. Compiling the LSDB.
2. Calculating the Shortest Path First (SPF) Tree.
3. Creating the routing table entries.

Compiling the LSDB

The LSDB is a database of all OSPF router LSAs. The LSDB is compiled by an ongoing exchange of LSAs between neighboring routers so that each router is synchronized with its neighbor. When the Network converged, all routers have the appropriate entries in their LSDB.

Calculating the SPF Tree Using Dijkstra's Algorithm

Once the LSDB is compiled, each OSPF router performs a least cost path calculation called the Dijkstra algorithm on the information in the LSDB and creates a tree of shortest paths to each other router and network with themselves as the root. This tree is known as the SPF Tree and contains a single, least cost path to each router and in the Network. The least cost path calculation is performed by each router with itself as the root of the tree

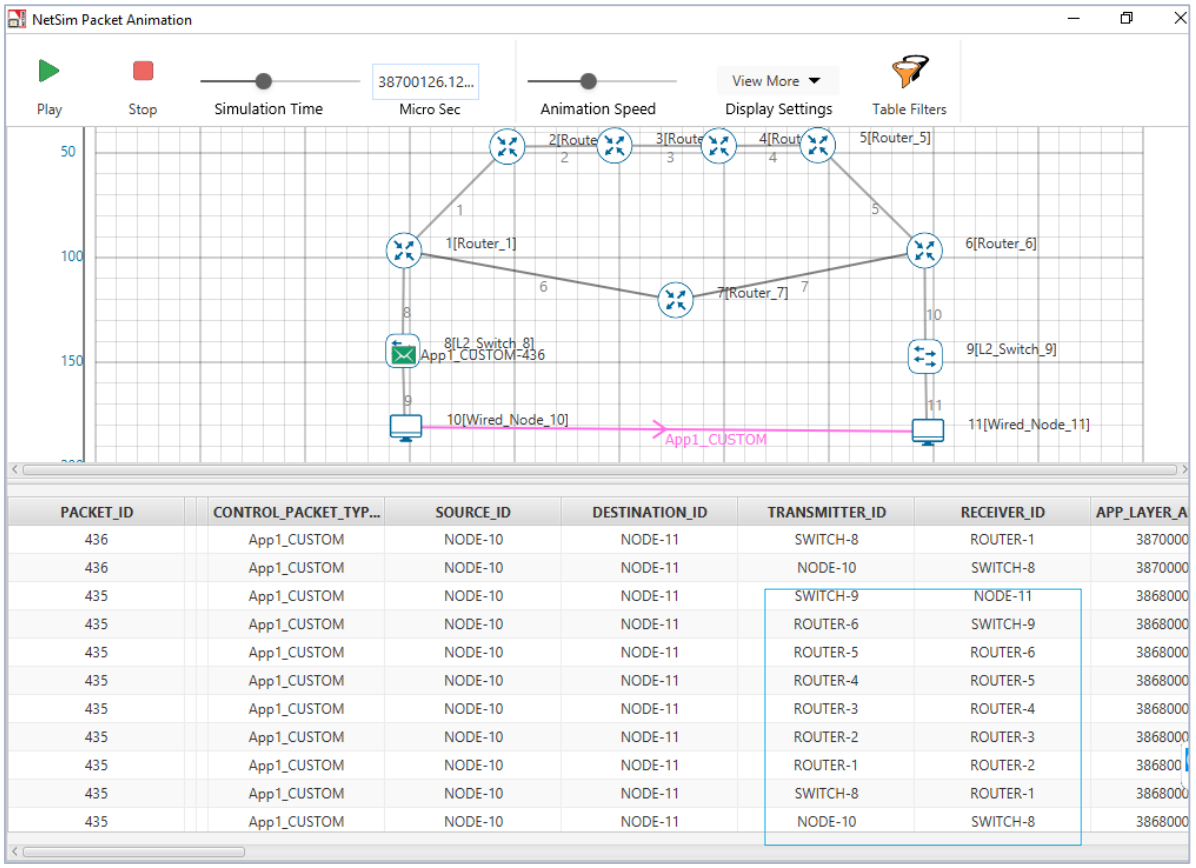
Calculating the Routing Table Entries from the SPF Tree

The OSPF routing table entries are created from the SPF tree and a single entry for each network in the AS is produced. The metric for the routing table entry is the OSPF-calculated cost, not a hop count.

Go to NetSim Packet Animation window and play animation. If app start time isn't changed then:

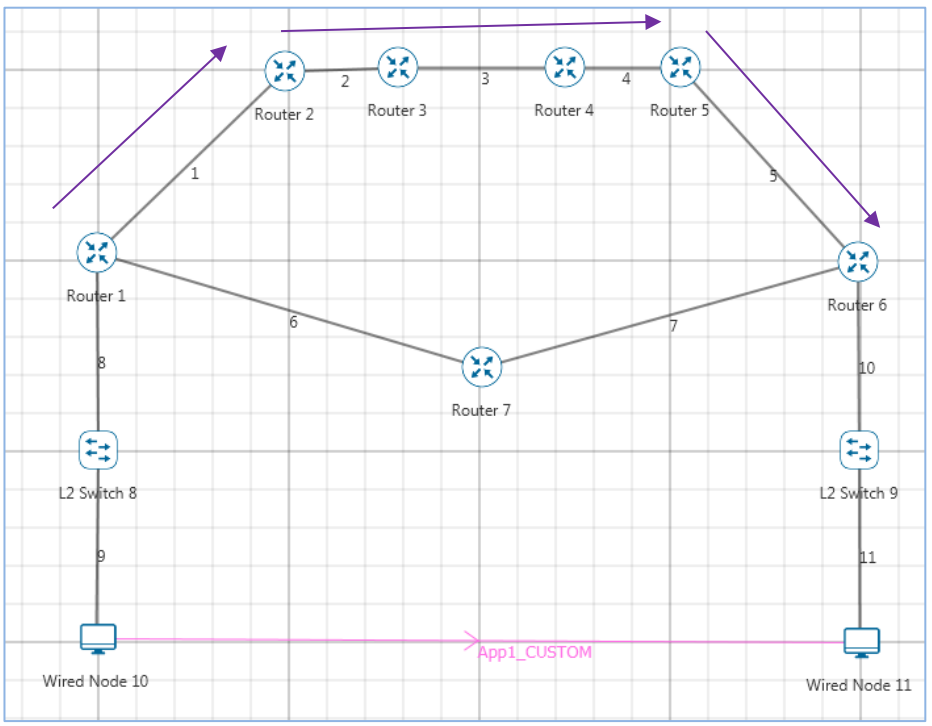
1. Packets generated before OSPF table convergence may be dropped at the gateway router.
2. The application may also stop if ICMP is enabled in the router
3. If TCP is enabled TCP may stop after the re-try limit is reached (since the SYN packets would not reach the destination)

You will be able to see the path packets take both in the animation and in the packet trace table as shown below:



PACKET_ID	CONTROL_PACKET_TYP...	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_A
436	App1_CUSTOM	NODE-10	NODE-11	SWITCH-8	ROUTER-1	3870000
436	App1_CUSTOM	NODE-10	NODE-11	NODE-10	SWITCH-8	3870000
435	App1_CUSTOM	NODE-10	NODE-11	SWITCH-9	NODE-11	3868000
435	App1_CUSTOM	NODE-10	NODE-11	ROUTER-6	SWITCH-9	3868000
435	App1_CUSTOM	NODE-10	NODE-11	ROUTER-5	ROUTER-6	3868000
435	App1_CUSTOM	NODE-10	NODE-11	ROUTER-4	ROUTER-5	3868000
435	App1_CUSTOM	NODE-10	NODE-11	ROUTER-3	ROUTER-4	3868000
435	App1_CUSTOM	NODE-10	NODE-11	ROUTER-2	ROUTER-3	3868000
435	App1_CUSTOM	NODE-10	NODE-11	ROUTER-1	ROUTER-2	386800
435	App1_CUSTOM	NODE-10	NODE-11	SWITCH-8	ROUTER-1	3868000
435	App1_CUSTOM	NODE-10	NODE-11	NODE-10	SWITCH-8	3868000

Shortest Path from Wired Node 10 to Wired Node 11 in OSPF (Use Packet Animation to view) **Wired Node 10->L2 Switch 8->Router 1->Router 2->Router 3->Router 4->Router 5->Router 6->L2 Switch 9->Wired Node 11.** OSPF chooses the upper path (cost is less-5) since OSPF is based on cost.



Note: The device / link numbering and IP Address setting in NetSim is based on order in which in the devices are dragged & dropped, and the order in which links are connected. Hence if the order in which a user executes these tasks is different from what is shown in the screen shots, users would notice different tables from what is shown in the screen shots.

13. M/D/1 Queuing

13.1 Objective:

To create an M/D/1 queue: a source to generate packets, a queue to act as the buffer and server, a sink to dispose of serviced packets and to study how the queuing delay of such a system varies.

13.2 Theory:

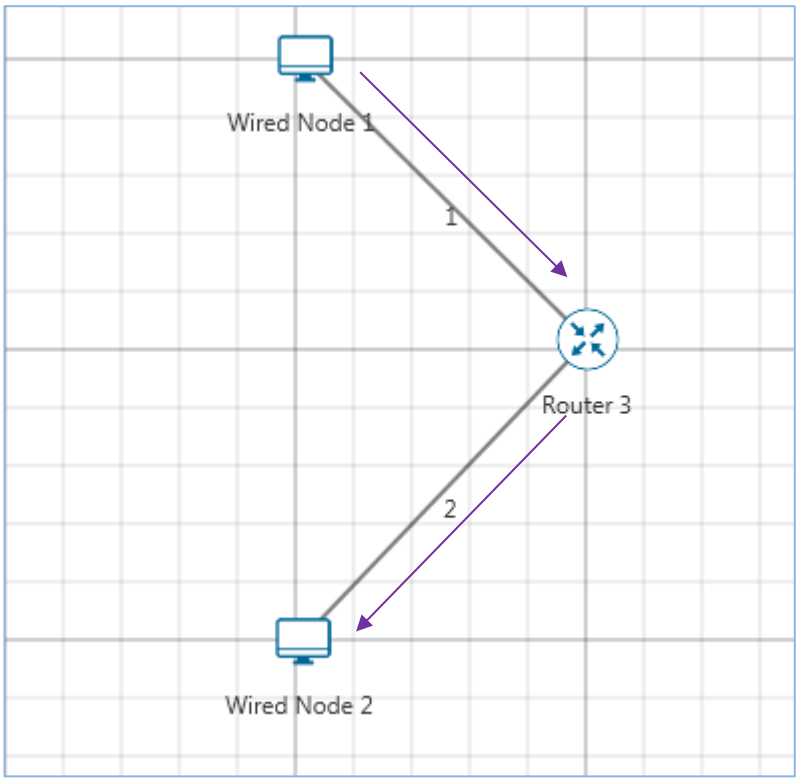
In systems where the service time is a constant, the M/D/1, single-server queue model, can be used. Following Kendall's notation, M/D/1 indicates a system where:

- **Arrivals** are a Poisson process with parameter **λ**
- **Service time(s)** is deterministic or constant
- There is **one server**

For an M/D/1 model, the total expected queuing time is $\boldsymbol{T = \frac{1}{2\mu} \times \frac{\rho}{1-\rho}}$

Where μ **=** Service Rate = 1/Service time and ρ is the utilization given as follows, $\boldsymbol{\rho = \frac{\lambda}{\mu}}$

To model an M/D/1 system in NetSim, we use the following model



Traffic flow from Node 1 to Node 2 (Node 1: Source, Node 2: Sink) **Inter-arrival time**: Exponential Distribution with mean 2000 µs **Packet size**: Constant Distribution with mean of 1250 bytes

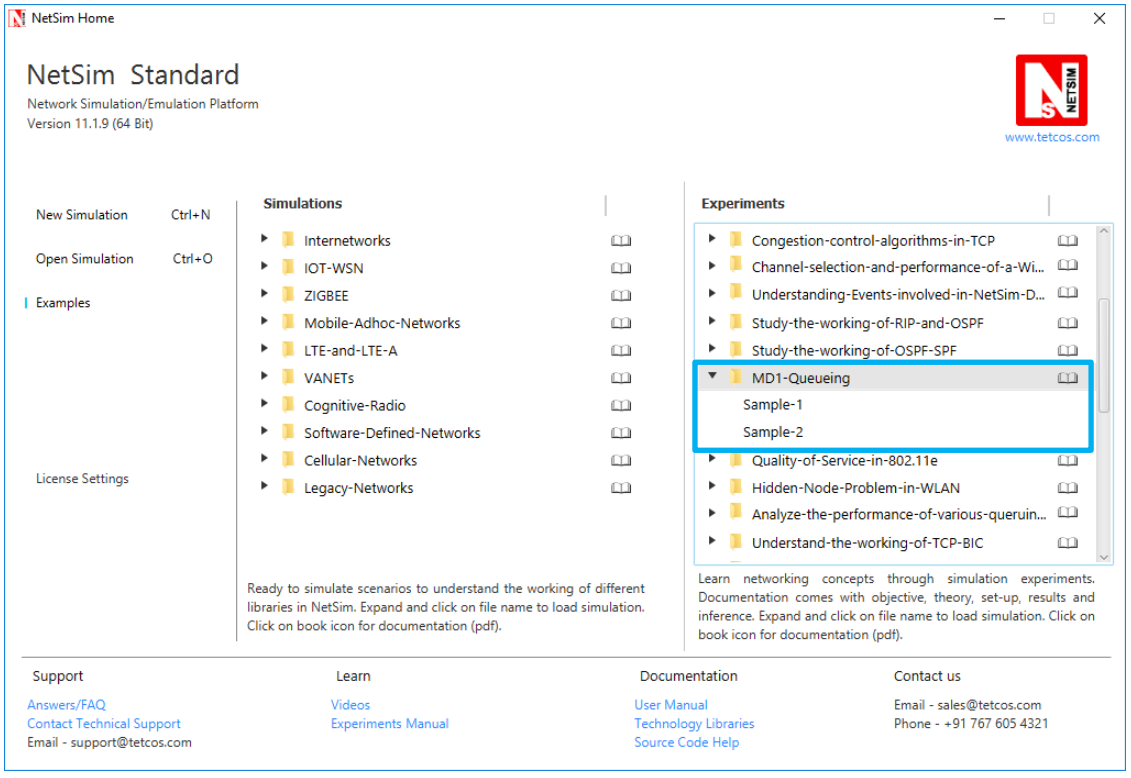
Note:

1. Exponentially distributed inter-arrivals times give us a Poisson arrival process. Different mean values are chosen as explained in the section Sample Inputs. (Dropping the devices in different order may change the result because the random number generator will get initialized differently)

2. To get constant service times, we use constant distribution for packet sizes. Since, the service (which in our case is link transmission) times are directly proportional to packet size (greater the packet size, greater the time for transmission through a link), a constant packet size leads to a constant service time.

Procedure:

Open Examples → MD1-Queuing as shown below:



Nodes 1 and Node 2 are connected with Router 1 by Link 1 and Link 2 respectively. Set the properties for each device as given below,

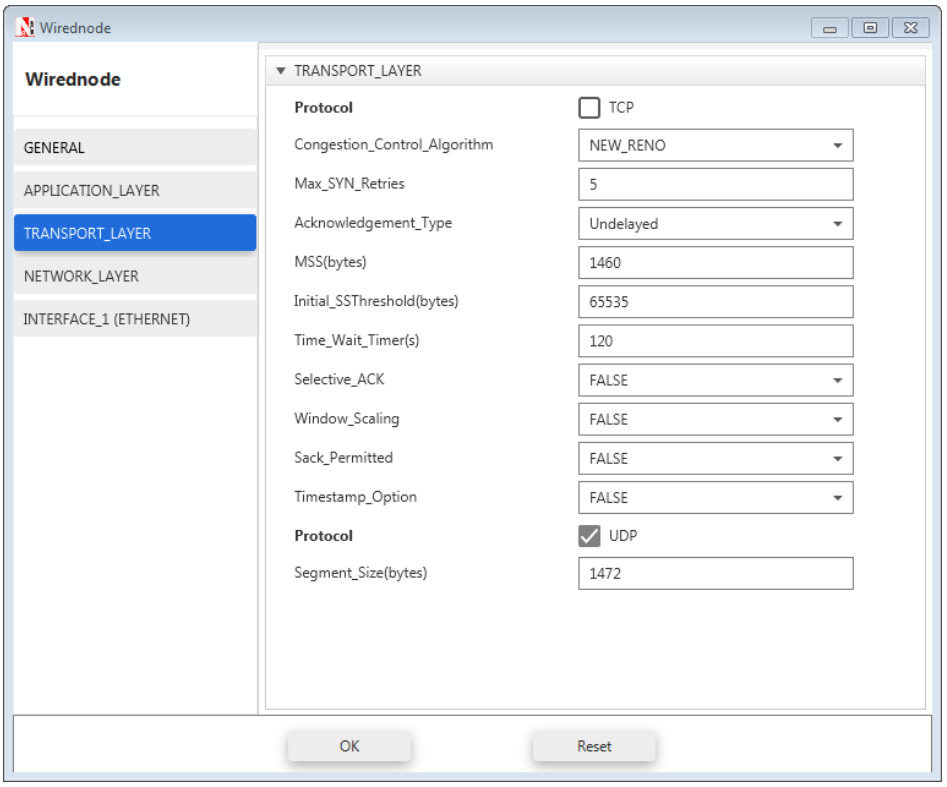
Sample 1:

Application Properties: Click on the Application icon present on the ribbon and set following properties:

Application Properties		
Application Method	Unicast	
Application Type	Custom	
Source_Id	1	
Destination_Id	2	
Packet Size	Distribution	Constant
	Value (bytes)	1250
Inter Arrival Time	Distribution	Exponential
	Packet Inter Arrival Time (µs)	2000

Disable TCP in the Transport Layer in **Node Properties** as follows (in both the wired nodes):

Link Properties	Link 1	Link 2
Uplink Speed (Mbps)	10	**10**
Downlink Speed (Mbps)	10	**10**
Uplink BER	0	**0**
Downlink BER	0	**0**
Uplink Propagation Delay (ms)	0	**0**
Downlink Propagation Delay(ms)	**0**	**0**



Router Properties: Accept the default properties for Router.

Simulation Time: 100 Sec

Observation:

Even though the packet size at the application layer is 1250 bytes, as the packet moves down the layers, some overhead is added which results in a greater packet size. This is the actual payload that is transmitted by the physical layer. The overheads added in different layers are shown in the below table and can be obtained from the packet trace:

Layer	Overhead (Bytes)
Transport Layer	**8**
Network Layer	**20**
MAC layer	**26**
Physical Layer	**0**
Total	**54**

Therefore, the payload size = Packet Size + Overhead

= 1250 + 54

= **1304 bytes**

Theoretical Calculation:

By formula,

$$\boldsymbol{Queuing\ Time} = \boldsymbol{T} = \frac{\mathbf{1}}{\mathbf{2}\boldsymbol{\mu}} \times \frac{\boldsymbol{\rho}}{\mathbf{1} - \boldsymbol{\rho}}$$

μ = Service Rate, i.e., the time taken to service each packet

= Link capacity (bps) / (Payload Size (Bytes) * 8)

= (10×10^6) / (1304*8)

= **958.59 packets / sec**

λ = Arrival rate, i.e., the rate at which packets arrive (Packets per second)

Inter-arrival time = 2,000 micro sec

Arrival rate **λ** = 1/ Inter Arrival time

= 1/2000 micro sec

= **500 packets / sec**

ρ = Utilization

= λ/μ

= 500/958.59

= 0.522

By formula, **Queuing Time** = $\frac{1}{2 \times 958.59} \times \frac{0.522}{1 - 0.522}$ = **569.61 micro sec**

13.3 Output:

After running the simulation, check the **"Delay"** in the Application Metrics.

Delay = 2654.9 micro sec

This Delay (also known as Mean Delay) is the sum of Queuing Delay, Total Transmission time and Routing Delay.

$$\begin{pmatrix}\textbf{Mean}\\\textbf{Delay}\end{pmatrix} = \begin{pmatrix}\textbf{Queuing}\\\textbf{Delay}\end{pmatrix} + \begin{pmatrix}\textbf{Total Transmission}\\\textbf{Time}\end{pmatrix} + \begin{pmatrix}\textbf{Routing}\\\textbf{Delay}\end{pmatrix}$$

Total Transmission Time is the sum of transmission time through Link 1 and Link 2.

Transmission time through each link is the same and is given by:

Transmission time through each link = $\frac{Payload\ Size\ (Bytes) \times 8}{Uplink\ Speed\ (bps)}$

$$= \frac{1304 \times 8}{10 \times 10^6}$$

= 1000.0 micro sec

Routing Delay is approximately 1 micro sec and can be found from the Event Trace. It is the difference between "Physical In" and "Physical Out" time for the Router.

Therefore, for simulation

Queuing Delay = 2654.9– (2 × 1000.0) – 1 = 653.9 micro sec

Sample 2

Keeping all the other parameters same as in previous example, if Packet Inter Arrival Time is taken as 1500 micro sec, then

λ = 666.67 packets per sec

Utilization **ρ = λ/μ = 666.67/958.59 = 0.695**

And **Queuing Time T = 1188.56 micro sec**

From NetSim,

Delay =3277.31 micro sec

Therefore, **Queuing Time** = 3277.31 - (2×1000.0) – 1 = **1276.3 micro sec**

Note: Obtained value is slightly higher than the theoretical value because of initial delays in forming ARP table, Switch table and Routing table etc.

A Note on M/M/1 queuing in NetSim

M/M/1 queue can be generated similarly by setting the "**Packet Size Distribution**" as "**Exponential**" instead of "**Constant**". However, the results obtained from simulation deviate from the theoretical

value because of the effect of packet fragmentation. Whenever a packet with size greater than Transport Layer MSS and / or MAC Layer MTU (which is 1500 bytes in NetSim) is generated, it gets fragmented in the application layer. Then the packet is sent as multiple frames, and makes it impossible to calculate the exact queuing time.

14. Quality of Service (QoS) in 802.11e based WLANs

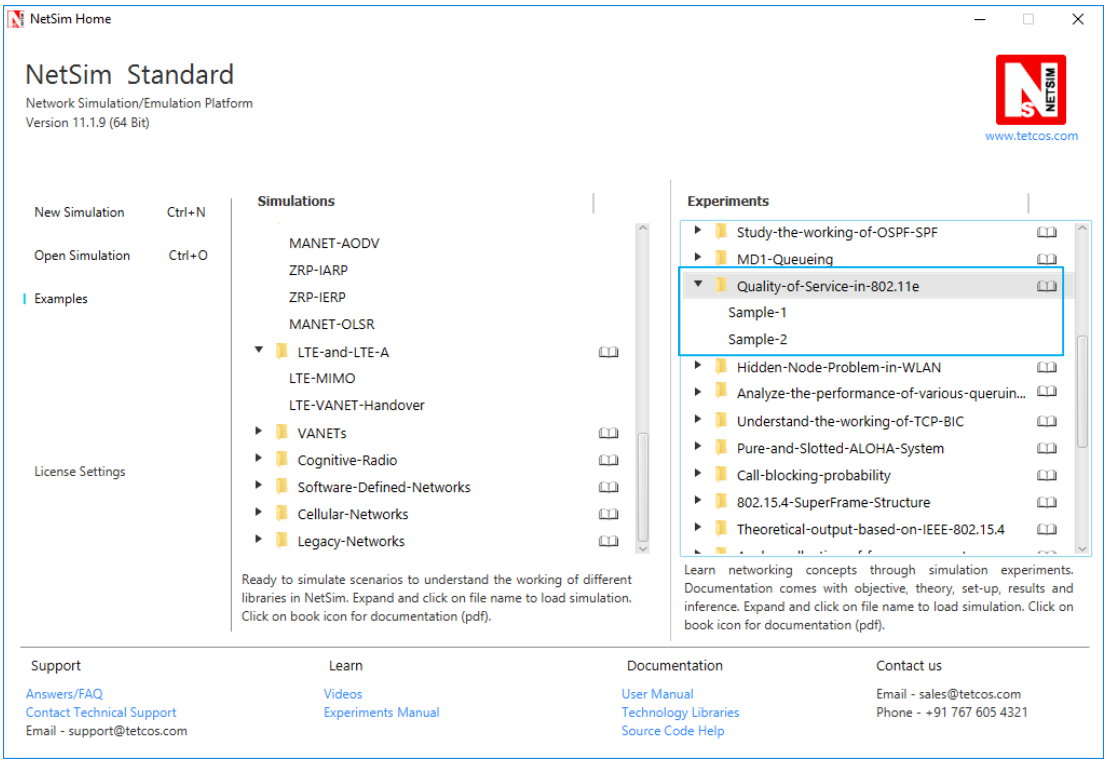
14.1 Theory

IEEE 802.11e Medium Access Control (MAC) is a supplement to the IEEE 802.11 Wireless Local Area Network (WLAN) standard to support Quality-of-Service (QoS). When 802.11e is enabled high-priority traffic has a higher chance of being sent than low-priority traffic: an application with high priority traffic waits a little less before its packet is processed and compared to an application with low priority traffic. The various application traffic generated in NetSim have the following priority and QoS values:

Application Type	Priority Value	Priority	QoS Class
Voice – One way **Voice – Two way**	8 8	High Very High	RTPS UGS
Video	6	Medium	nRTPS
FTP	2	Low	BE
Database	2	Low	BE
Custom	2	Low	BE

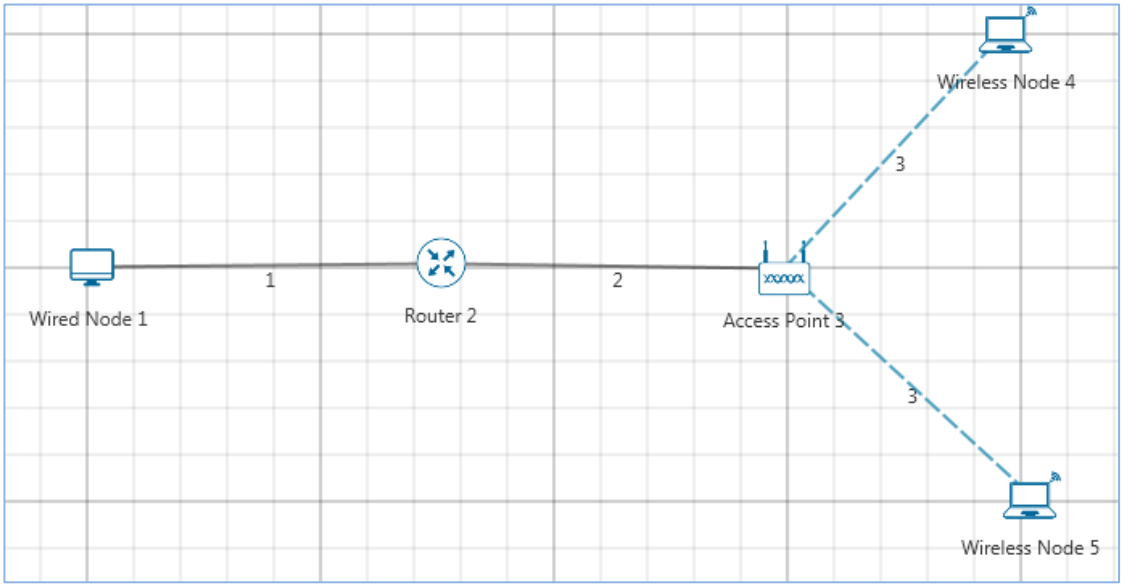
14.2 Network Set Up

Open Examples → Quality-of-Service-in-802.11e as shown below:



Sample 1:

Create a network as per the screen shot below. Devices Required: 2 Wireless Node, 1 Access point, 1 Router and 1 Wired Node.



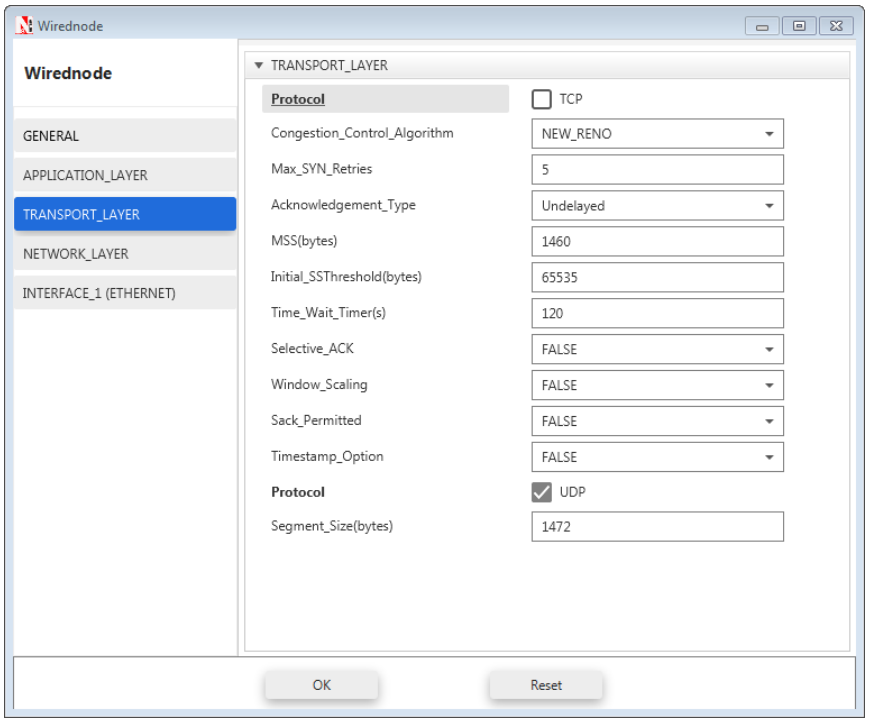
Set

Access Point → Right Click Properties → X-Coordinate: 250, Y-Coordinate: 100

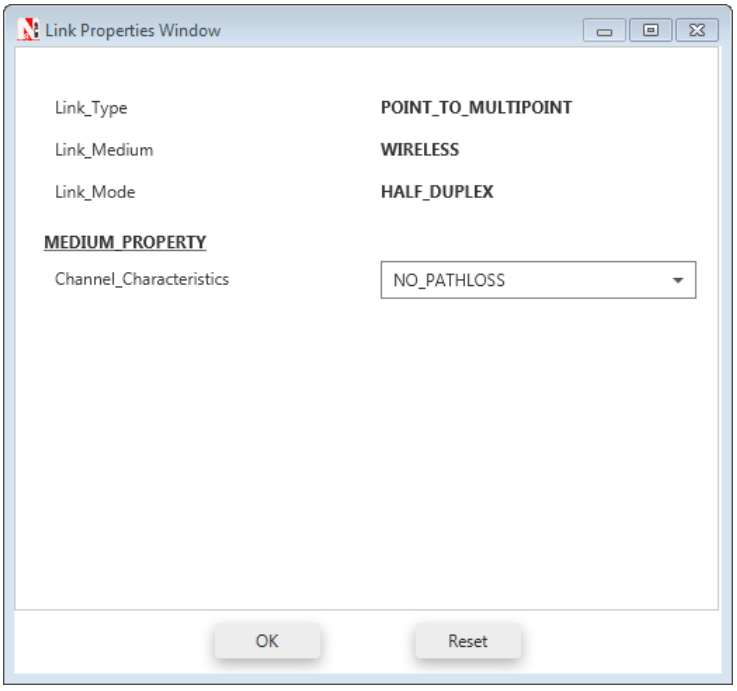
Wireless Node 4 → Right Click Properties → X-Coordinate: 300, Y-Coordinate: 100

Wireless Node 5 → Right Click Properties → X-Coordinate: 250, Y-Coordinate: 150

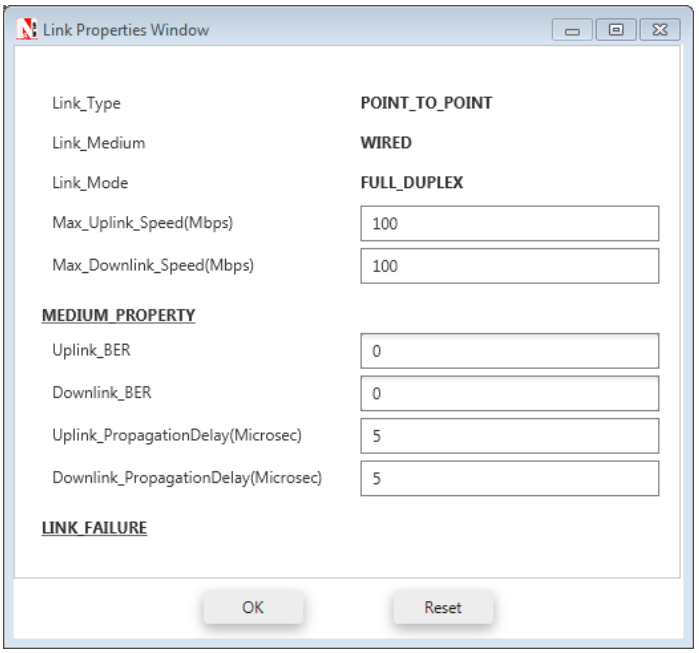
Node properties: Disable TCP in all nodes in Transport layer as follows:



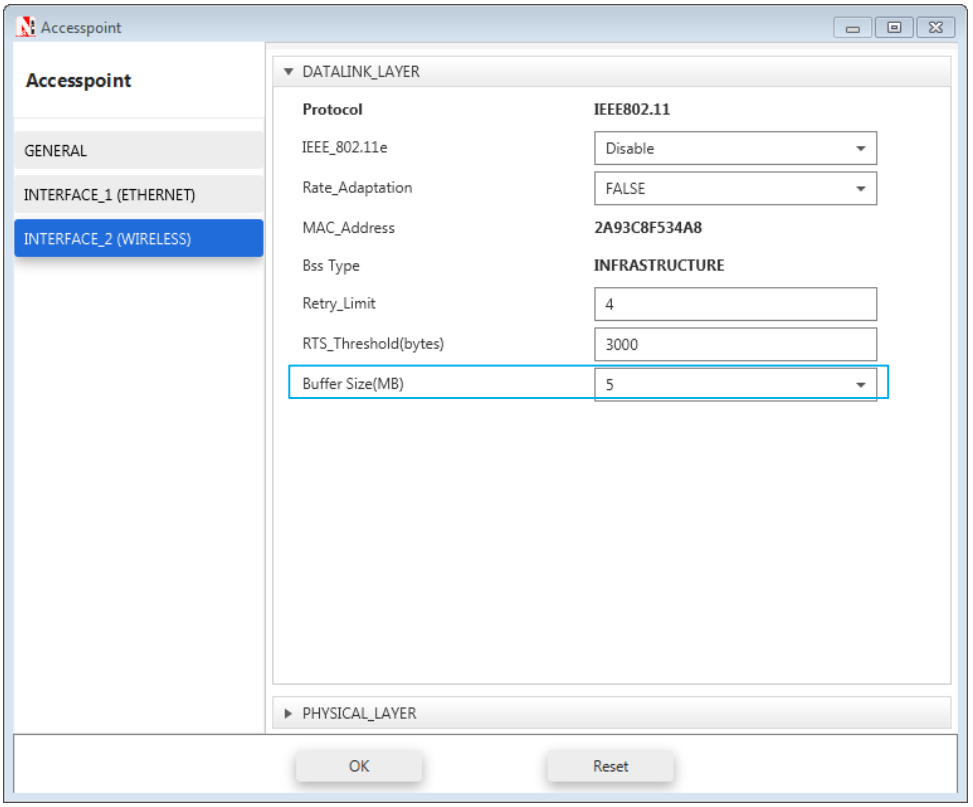
Wireless Link Properties: Right click on Wireless link and Change the channel characteristics as "No Path Loss"



Wired Link Properties: Right click on wired link and Change the Bit Error Rate as Zero (0), and leave the other properties at their default value.

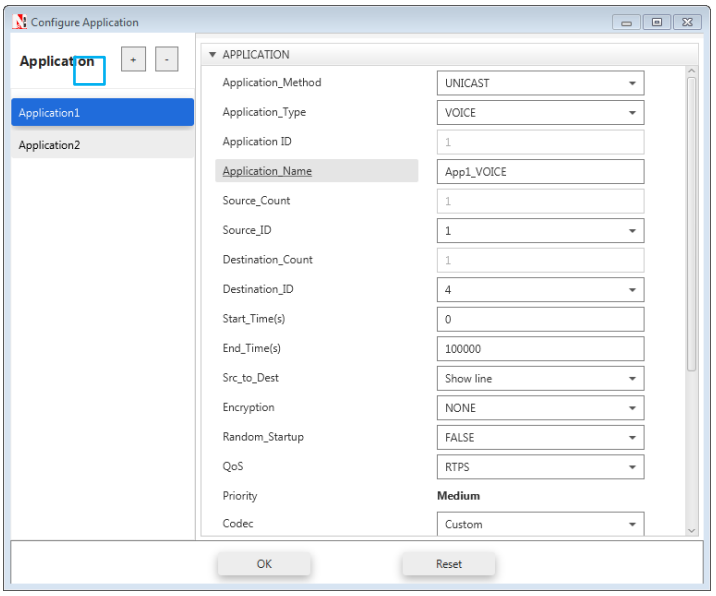


Access Point Properties: Right click on access point properties. In interface1_Wireless, enable IEEE802.11e and set the buffer size as 5 (MB) as shown in below figure:



Wireless Node Properties: Right click on Wireless Node and in Interface1_Wireless, enable IEEE802.11e.

Application Settings: Click on the Application icon present on the ribbon. Set properties as shown below:



Click on the + button to add multiple application

Application	No1: Voice (Codec-Custom)	No 2: CBR
Source ID	Wired Node 1	Wired Node 1
Destination ID	Wireless Node 4	Wireless Node 5
Packet Size		
Distribution	Constant	Constant
Value(Bytes)	1000	1000
Inter Arrival Time		
Distribution	Constant	Constant
Value(µs)	800	800

Run Simulation with Simulation Time set as10 sec. After completion of the experiment, “Save” the experiment in the current workspace as Sample 1. Note the application throughputs.

Sample 2:

Open Sample 1, and disable IEEE_802.11e in both Access point and Wireless Node properties and run the simulation for 10 seconds. Note the application throughputs.

14.3 Measurements and Outputs:

IEEE 802.11e	Application	Generation rate (Mbps)	Throughput (Mbps)	Delay (Micro. Sec.)
Enable (Sample 1)	Voice	10	3.15	964819.9
	CBR	10	2.19	6453921.6
Disable (Sample 2)	Voice	10	2.64	3673734.7
	CBR	10	2.64	3671227.4

14.4 Inference

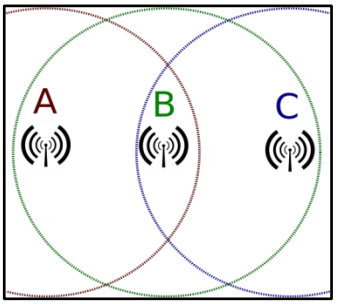
In sample 1, since QoS is enabled voice sees a higher priority than CBR. Hence voice packets in the queue are first transmitted before CBR packets are transmitted. In sample 2, since QoS has been disabled, priority is not considered for the applications. Hence they both see the same throughput.

As an additional note, when QoS is enabled the throughput for voice is 3.15 Mbps and for CBR it is 2.19, and when QoS is disabled the throughput for both is 2.64 Mbps per application or 5.28 Mbps for both applications put together. This value of around 5.5 Mbps is the maximum throughput an 802.11b access point can support. There is a slight drop in overall throughput when stations are present due to channel contention between the stations.

15. Study the hidden node problem in WLAN

15.1 Theory:

Hidden nodes in a wireless network are nodes that are out of range of other nodes or a collection of nodes. In a wireless network, it is likely that the node at the far edge of the access point's range, which is known as **A**, can see the access point, but it is unlikely that the same node can see a node on the opposite end of the access point's range, **C**. These nodes are known as *hidden*.

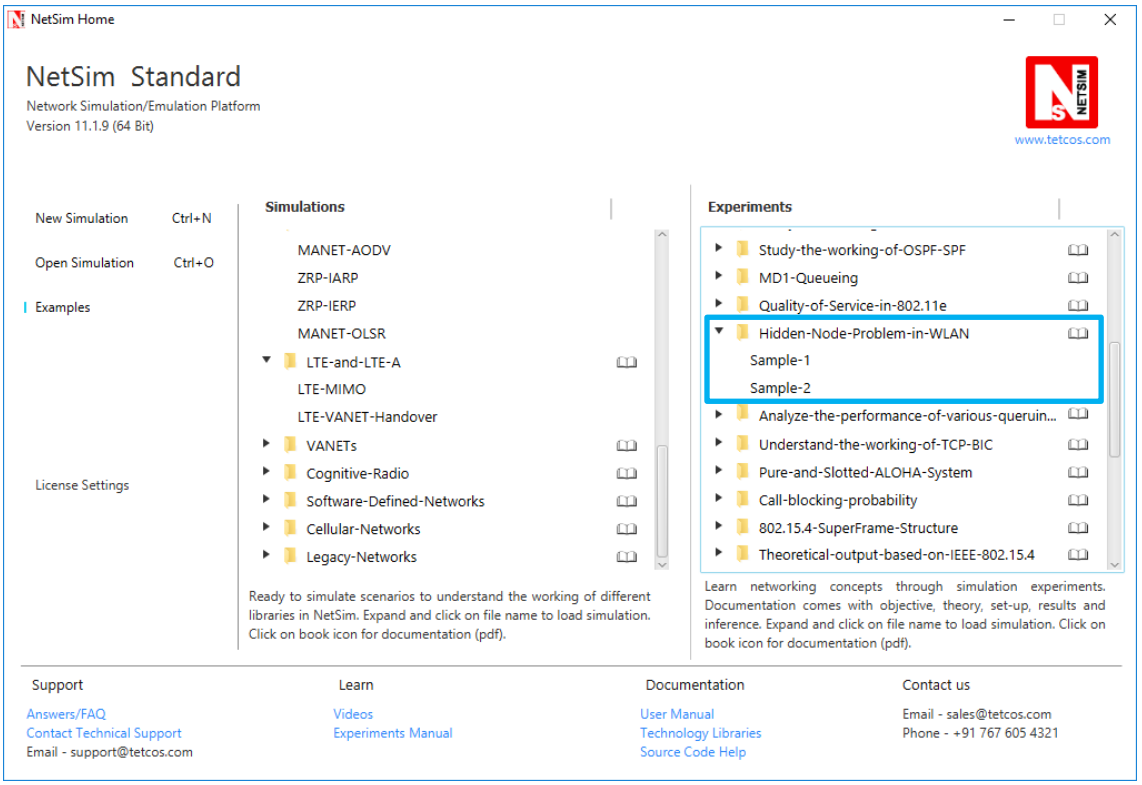


The problem is when nodes A and C start to send packets simultaneously to the access point B. Because the nodes A and C are out of range of each other and so cannot detect a collision while transmitting, Carrier sense multiple access with collision detection (CSMA/CD) does not work, and collisions occur, which then corrupt the data received by the access point.

To overcome the hidden node problem, RTS/CTS handshaking (IEEE 802.11 RTS/CTS) is implemented in conjunction with the Carrier sense multiple access with collision avoidance (CSMA/CA) scheme. The same problem exists in a MANET.

15.2 Procedure:

Open Examples → Hidden-Node-Problem-in-WLAN as shown below:



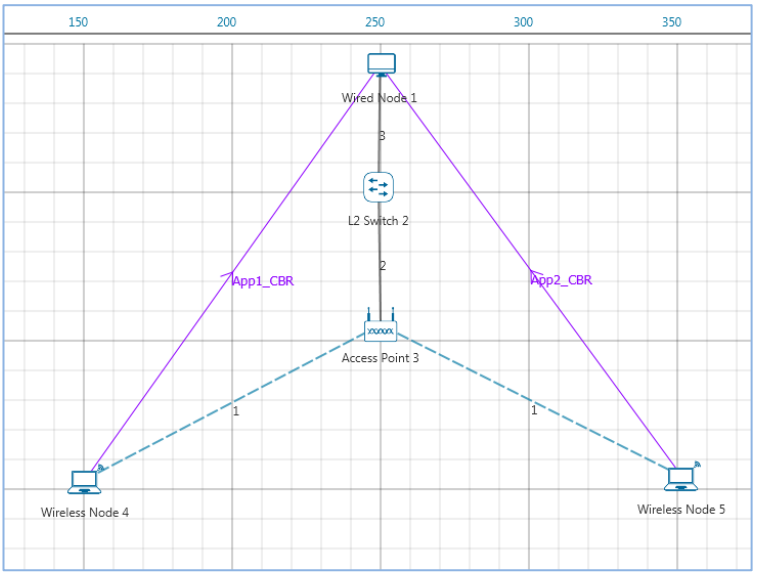
Sample Inputs:

Follow the steps given in the different samples to arrive at the objective.

In Sample 1,

- Total no of APs (Access Points) used: 1
- Total no of Wireless Nodes used: 2
- Total no of Switch used: 1
- Total no of Wired Node used: 1

The devices are interconnected as shown:



Also edit the following properties of AP 1, Wireless Node 4 and 5:

Wireless Node 5 Properties	
X/Lat	350
Y/Lon	150

Wireless Node 4 Properties	
X/Lat	150
Y/Lon	150

In NetSim RTS CTS mechanism can be enabled or disabled in WLAN using RTS THRESHOLD parameter. The RTS THRESHOLD parameter is available in the Datalink layer properties of the WLAN devices. RTS CTS mechanism is enabled if the RTS THRESHOLD is less than the packet size. RTS CTS mechanism is disabled if the RTS THRESHOLD is greater than the packet size.

Access Point Properties	
X/Lat	250
Y/Lon	100
Interface_1 (Wireless) Properties	
RTS Threshold(bytes)	3000

Edit Wireless link properties as shown:

Wireless Link Properties	
Channel Characteristics	Path Loss Only

Path Loss Model	LOG_DISTANCE
Path Loss Exponent(n)	2

Properties of Wired Links are default.

Disable TCP in both the Wireless Nodes and Wired Node.

Click on Application, set properties as shown below and run the simulation.

Application Properties	Application 1	Application 2
Application Method	Unicast	
Application Type	CBR	
Source_Id	4	5
Destination_Id	1	1
Packet Size		
Distribution	Constant	Constant
Value (bytes)	1460	1460
Inter Arrival Time		
Distribution	Constant	Constant
Value (micro sec)	20000	20000

(Note: Packet size here refers to the size of the packet along with the overheads added in the layers above and not the application layer packet size.)

Simulation Time - 10 Sec

Note: The Simulation Time can be selected only after the following two tasks,

- ***Set the properties for all the devices and links.***
- ***Click on Run Simulation button.***

Upon completion of the experiment, Save it using the save button (or ctrl + s) in the current workspace.

In Sample 2,

Set the properties of Access Point as follows:

Access Point Properties	
X/Lat	250
Y/Lon	100
Interface_1 (Wireless) properties	
RTS Threshold(bytes)	1000

Simulation Time - 10 Sec

(Note: The Simulation Time can be selected only after the following two tasks,

- ***Set the properties for all the devices and links.***
- ***Click on Run Simulation button.)***

Upon completion of the experiment, Save it using the save button (or ctrl + s) in the current workspace.

15.3 Output:

From Network metrics user can able to see data packet and control packet

Comparison Table:

Collided Packets	Without RTS/CTS	With RTS/CTS
Data Packets	22	0
Control Packets	0	20

15.4 Inference:

During simulations performed without RTS CTS mechanism enabled we notice data packet collisions. This is because nodes x and y transmit in parallel as they are out of range of each other. RTS CTS mechanism helps in avoiding data packet collisions with the help of RTS and CTS control packets that are exchanged before attempting transmissions. However, there can be control packet collisions which may involve, WLAN Ack's and RTS, CTS packets.

16. Analyze the performance of FIFO, Priority and WFQ Queuing Disciplines

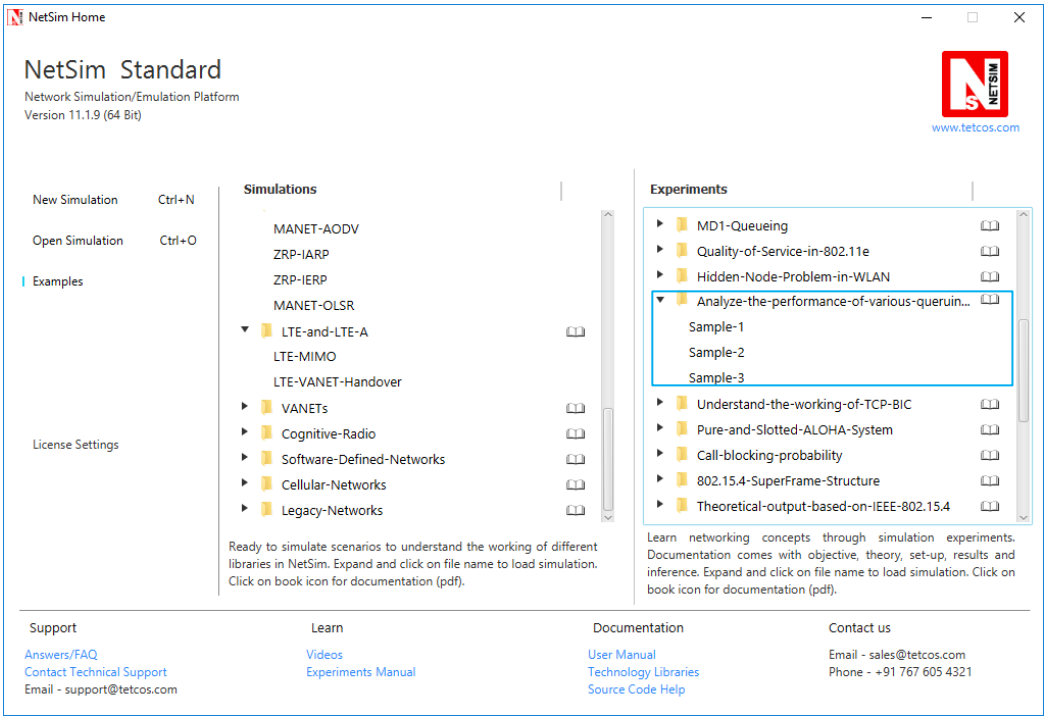
16.1 Introduction

As part of the resource allocation mechanisms, each router must implement some queuing discipline that governs how packets are buffered while waiting to be transmitted. Various queuing disciplines can be used to control which packets get transmitted (bandwidth allocation) and which packets get dropped (buffer space). The queuing discipline also affects the latency experienced by a packet, by determining how long a packet waits to be transmitted. Examples of the common queuing disciplines are first-in-first-out (FIFO) queuing, priority queuing (PQ), and weighted-fair queuing (WFQ).

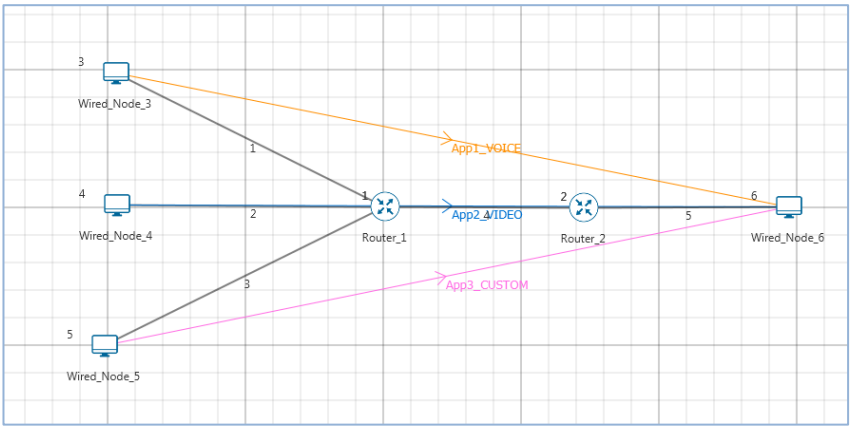
16.2 Network Set Up

Sample 1(FIFO)

Step 1: Go to Examples → Analyze-the-performance-of various-queuing-disciplines as shown:



Step 2: Click and drop 4 Wired Nodes and 2 Routers onto the Simulation Environment as shown and link them.



Disable TCP in all nodes. This can be done by Right Click on the device and unchecking TCP in transport layer properties.

Step 3: Right click on Wired Link and Select Properties and edit as shown below:

Link Properties	Link 1	Link 2	Link 3	Link 4	Link 5
Max Uplink Speed (Mbps)	10	10	10	5	5
Max Downlink Speed(Mbps)	10	10	10	5	5

Step 4: Right click on Router and Select Properties. In all the Interface_WAN properties, select Scheduling type as “FIFO”

Step 5: To run the simulation, Click on the Application icon present on the ribbon and set the properties as follows and click on Accept:

Application Properties	Application 1	Application 2	Application 3
Application Type	**Voice (Codec-Custom)**	**Video**	**Custom**
Source_Id	**3**	**4**	**5**
Destination_Id	**6**	**6**	**6**
QoS	**RTPS**	**NRTPS**	**BE**
Packet Size			
Distribution	**Constant**	**Frame_Per_Sec**	**Constant**
Value (bytes)	**1460**	**50**	**1000**
Inter Arrival Time			
Distribution	**Constant**	**Pixel_Per_Frame**	**Constant**
Value (micro secs)	**2336**	**100000**	**1333**

Note: For Voice application set codec as Custom

Step 6: Click on Run Simulation and set the Simulation Time as 10 sec

Save the experiment as say “FIFO-Sample-1”. Note down the Application throughputs

Sample 2(Priority): Open sample 1, and change the Scheduling type as Priority (in Router Properties). Upon completion of simulation, “Save” the experiment as say “Priority-Sample-2”. Note down the Application throughputs

Sample 3(WFQ): Open sample 2, and change the Scheduling type as WFQ (in Router Properties). Upon completion of simulation, “Save” the experiment as say “WFQ-Sample-3”. Note down the Application throughputs

16.3 Measurements and Outputs:

Comparison Table

Application	Traffic Generation Rate (Mbps)*	FIFO-Sample-1 Throughput (Mbps)	Priority-Sample-2 Throughput (Mbps)	WFQ-Sample-3 Throughput (Mbps)
Voice	5	1.75 ~ (5 / 13.6) *5	3.81	1.89
Video	2.6	0.89 ~ (2.6/13.6) *5	0.28	0.85
Custom	6	2.12 ~ (6/13.6) *5	0.70	2.02
Total	13.6	4.76 ~ 5	4.79 ~ 5	4.76 ~ 5

*The traffic generation rate is based on settings done in step 5.

The 5 mentioned above refers to 5 Mbps which is the data rate of link 4.

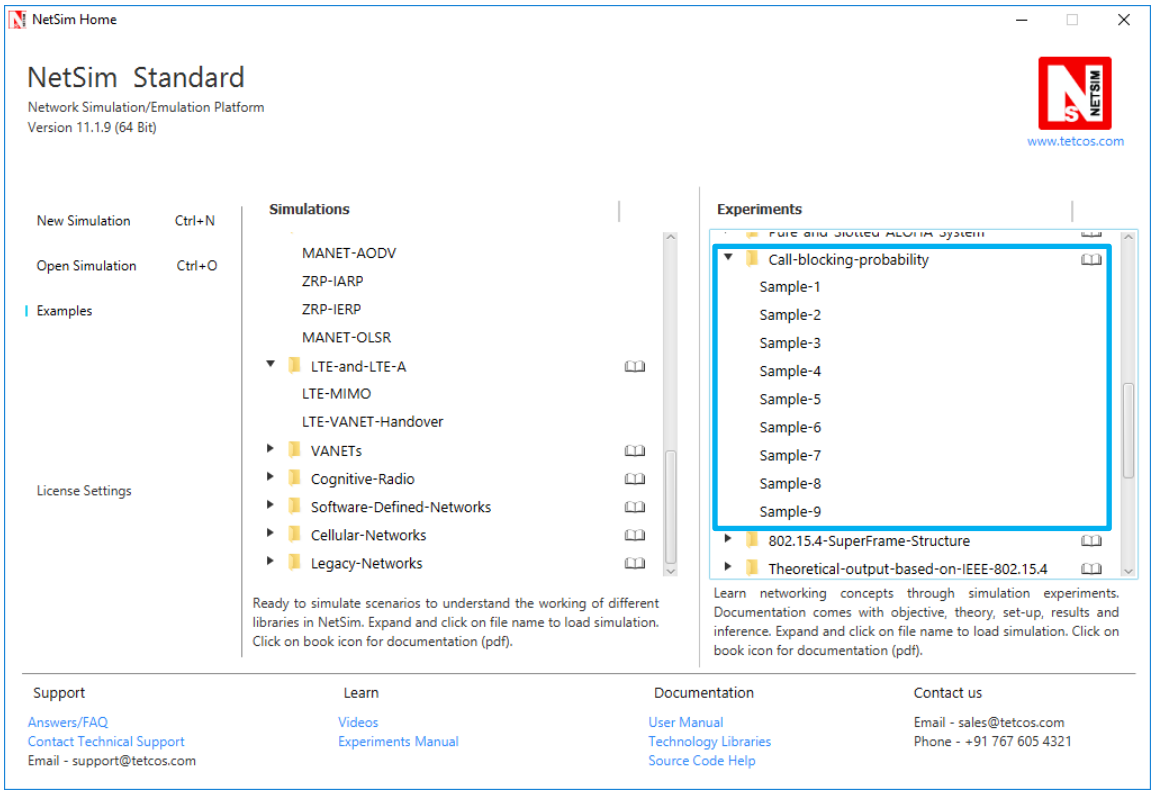
16.4 Inference

In FIFO, packets will get served based on their packet arrival time to router. Therefore, since link 4 is a 5 Mbps link, the throughputs of Voice, Video and Custom application is equal to the ratio of their generation rates. Priority scheduling technique processes packets based on their priority. Hence voice and video which have higher priority take up the complete bandwidth available. Weighted fair queuing (WFQ) assigns a weight to each application and hence gives a result between that is in between priority and FIFO.

17. Study how call blocking probability varies as the load on a GSM network is continuously increased

17.1 Procedure:

In NetSim, Open Examples → Call-blocking-probability as shown below:



Follow the steps given in the different samples to arrive at the objective.

In this Experiment,

- One BTS (BTS A) and one MSC (MSC B) is used
- Total no of MS used: Vary from 4 to 20 in steps of 2.

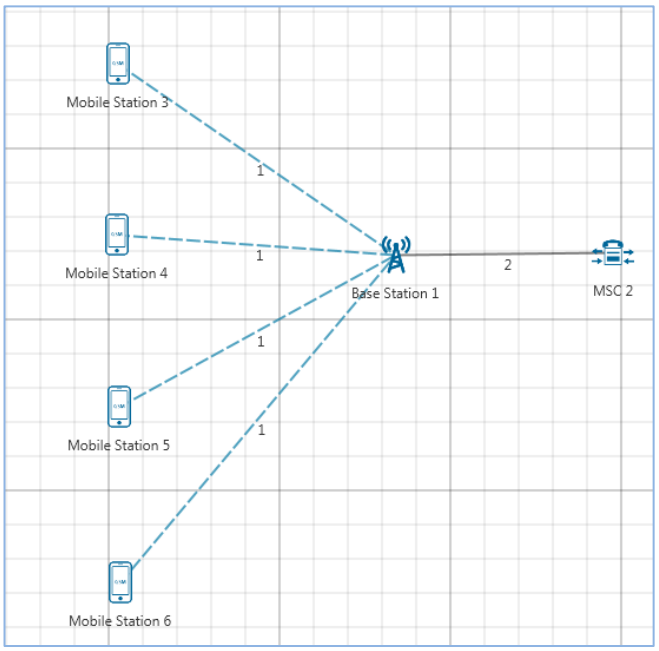
The devices are inter connected as given below,

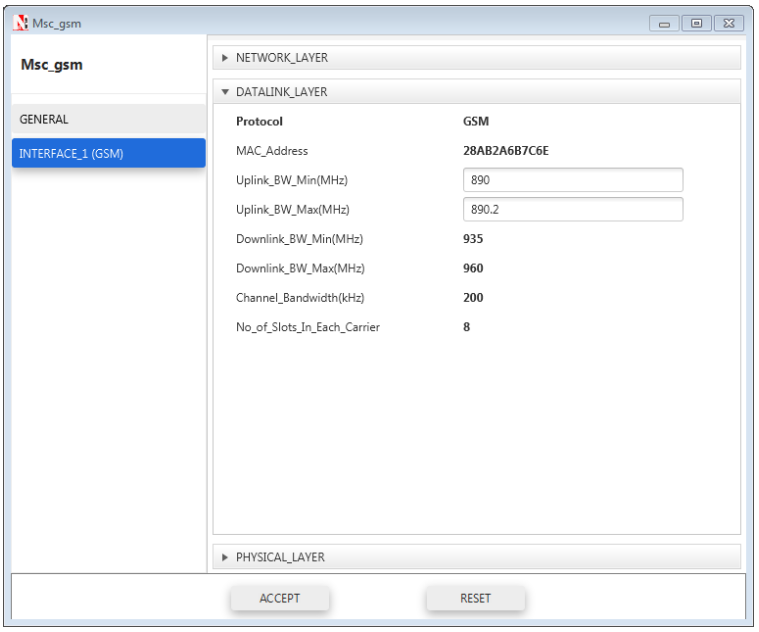
- All the MS are placed in the range of BTS 1

Set the properties by following the tables for each sample,

Inputs for Sample 1

Number of MS = 4





Accept default properties for BTS.

Edit **Uplink Bandwidth Min** to **890** MHz and **Uplink Bandwidth Max** to **890.2** MHz in MSC properties.

In the Sample 1, two Applications are run. Click on application Icon present on the ribbon and set properties as follows

To add another application click on “+” symbol

Application Properties	Application 1	Application2
Application type	Erlang_call	Erlang_call
Source_Id	3	5
Destination_Id	4	6
Call		
Duration_ Distribution	Exponential	Exponential
Duration(s)	60	60
Inter Arrival Time (sec)	10	10
IAT_ Distribution	Exponential	Exponential
Codec	Custom	Custom
Inter Arrival Time distribution	Constant	Constant
Packet Distribution	Constant	Constant
Service Type	CBR	CBR
Packet Size	33	33
Inter Arrival Time (μs)	20000	20000

Simulation Time – 100 sec

Inputs for Sample 2

Number of MS = 6. Add one more Application and set the properties as above with Source_Id as 7 and Destination_Id as 8.

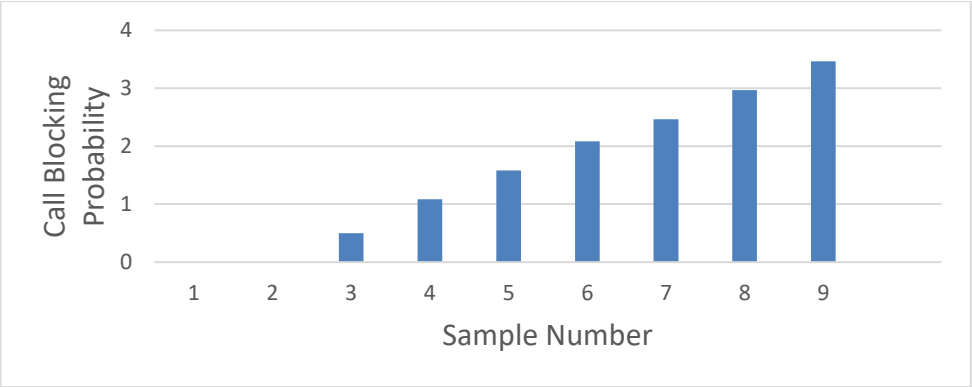
Likewise, increase the number of MS by 2 up to 20 and set properties for different Samples by adding an application every time and changing Source_Id and Destination_Id.

Simulation Time – 100 sec

17.2 Output

To view the output, go to the Cellular Metrics. In MS metrics, take sum of call blocking probability (It is the as ratio of Total call blocked to Total call generated).

Comparison Charts:



*** All the above plots highly depend upon the placement of Mobile station in the simulation environment. So, note that even if the placement is slightly different the same set of values will not be got but one would notice a similar trend.

17.3 Inference:

When the number of MS is increased from 4 to 20 the call blocking probability increases from 0 to 0.94. As we increase the number of mobile stations more calls are generated. This increases the traffic load on the system & more calls generated implies more channel requests arrive at the base station but the number of channels is fixed. So when the base station does not find any free channel the call is blocked. An additional observation is that the call blocking is zero until 8 MS. This is because the number of channels is sufficient to handle all call that 6 MS may generate. Only after this the base station does not find free channels and blocks calls.

18. Study the 802.15.4 SuperFrame Structure and analyze the effect of SuperFrame order on throughput

18.1 Introduction:

A coordinator in a PAN can optionally bound its channel time using a SuperFrame structure which is bound by beacon frames and can have an active portion and an inactive portion. The coordinator enters a low-power (sleep) mode during the inactive portion.

The structure of this SuperFrame is described by the values of macBeaconOrder and macSuperframeOrder. The MAC PIB attribute macBeaconOrder, describes the interval at which the coordinator shall transmit its beacon frames. The value of macBeaconOrder, BO, and the beacon interval, BI, are related as follows:

For $0 \leq BO \leq 14$, BI = aBaseSuperframeDuration * 2^{BO} symbols.

If BO = 15, the coordinator shall not transmit beacon frames except when requested to do so, such as on receipt of a beacon request command. The value of macSuperframeOrder, SO shall be ignored if BO = 15.

An example of a SuperFrame structure is shown in following Figure.

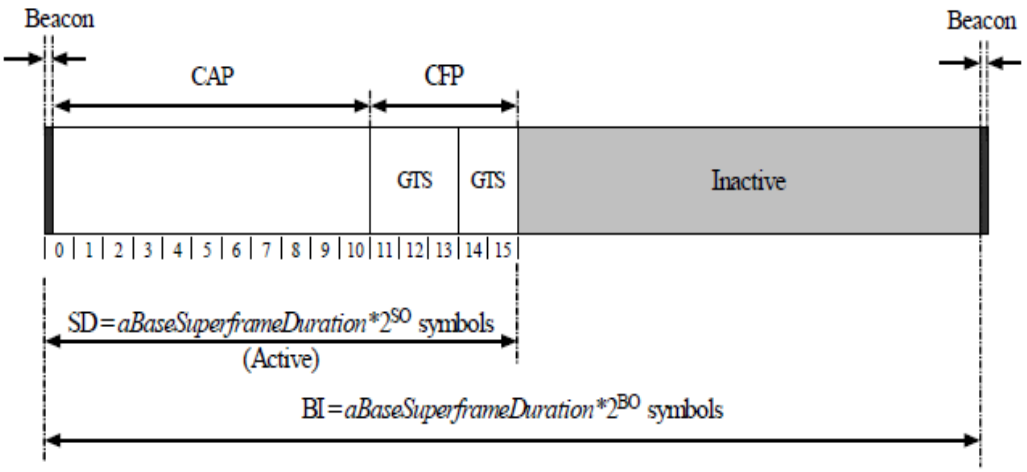


Fig: An example of the Super Frame structure

Theoretical Analysis:

From the above SuperFrame structure,

$$SuperFrame\ Duration = aBaseSuperframeDuration * 2^{BO}$$

$$Active\ part\ of\ SuperFrame = aBaseSuperframeDuration * 2^{SO}$$

$$Inactive\ part\ of\ SuperFrame = aBaseSuperframeDuration * (2^{BO} - 2^{SO})$$

If SuperFrame Order (SO) is same as Beacon Order (BO) then there will be no inactive period and the entire SuperFrame can be used for packet transmissions.

If BO=10, SO=9 half of the SuperFrame is inactive and so only half of SuperFrame duration is available for packet transmission. If BO=10, SO=8 then (3/4)th of the SuperFrame is inactive and so nodes have only (1/4)th of the SuperFrame time for transmitting packets and so we expect throughput to approximately drop by half of the throughput obtained when SO=9.

Percentage of inactive and active periods in SuperFrame for different SuperFrame Orders is given below:

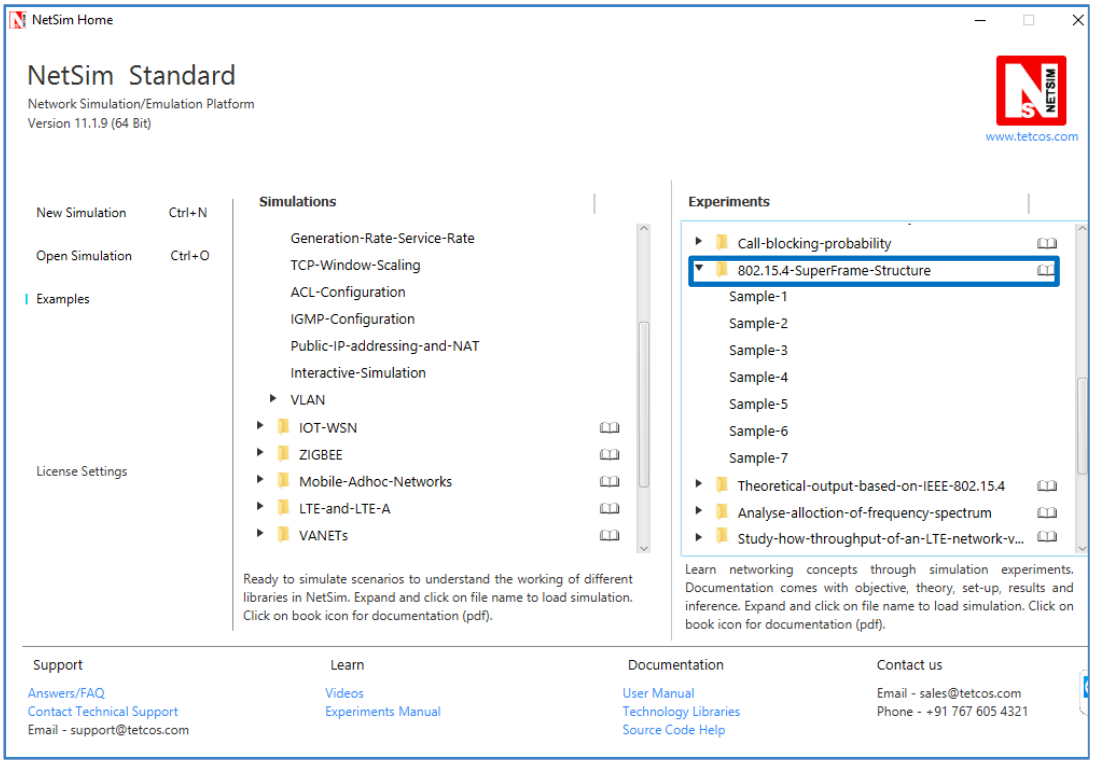
Beacon Order (BO)	Super Frame Order (SO)	Active part of SuperFrame(%)	Inactive part of SuperFrame (%)	Throughput estimated (%)
10	10	100	0	**> 200% of T**
10	9	50	50	**Say T = 21.07 (Got from simulation)**
10	8	25	75	**50 % T**
10	7	12.5	87.5	**25 % T**
10	6	6.25	93.75	**12.5 % of T**
10	5	3.125	96.875	**6.25 % of T**
10	4	1.5625	98.4375	**3.12% of T**
10	**3**	**0.78125**	**99.21875**	**1.56 % of T**

We expect throughput to vary in the active part of the SuperFrame as sensors can transmit a packet only in the active portion.

Simulation:

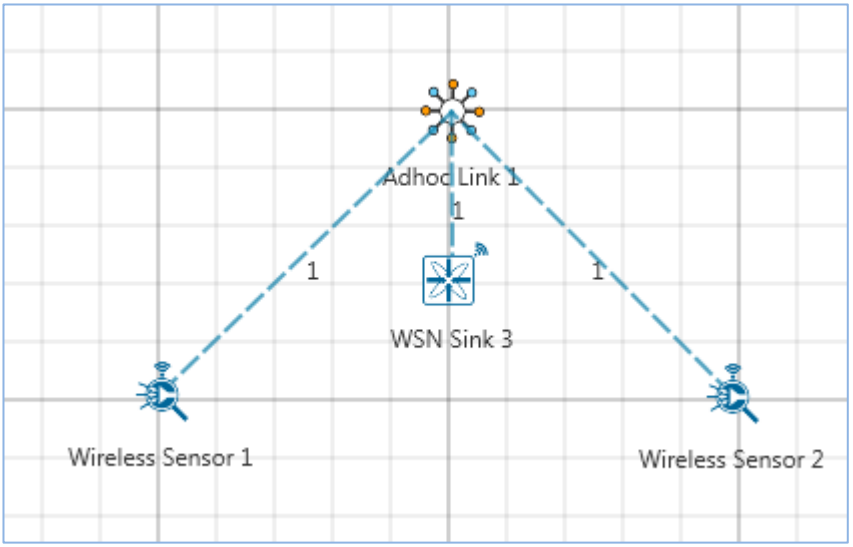
How to Create Scenario & Generate Traffic:

In NetSim, Open Examples → 802.15.4-Superframe-Structure as shown below:



In grid settings, enter side length as 100 m and select sensor placement strategy as via click & drop

Click & drop one Sensor Node, Sink Node onto the Simulation Environment:



Sink Node Properties: Change the following properties for WSN Sink.

Sink Node Properties	Values
Interface1_Zigbee: Data link Layer	
Beacon Mode	**Enable**
Beacon Order	**10**
Super frame Order	**10**

Disable TCP in all nodes

Click on Application Icon present on the ribbon and edit the following properties:

Application Properties	
Application Type	**Custom**
Source_Id	**Sensor 1**
Destination_Id	**Sensor 2**
Packet_Size	
Distribution	**Constant**
Value(Bytes)	**25**
Inter_Arrival_time	
Distribution	**Constant**
Value(micro sec)	**3000**

Simulation Time -30 Sec

(Note: The Simulation Time can be selected only after doing the following two tasks,

- ***Set the properties of Node, WSN Sink & Environment.***
- ***Click on Run Simulation and save the experiment)***

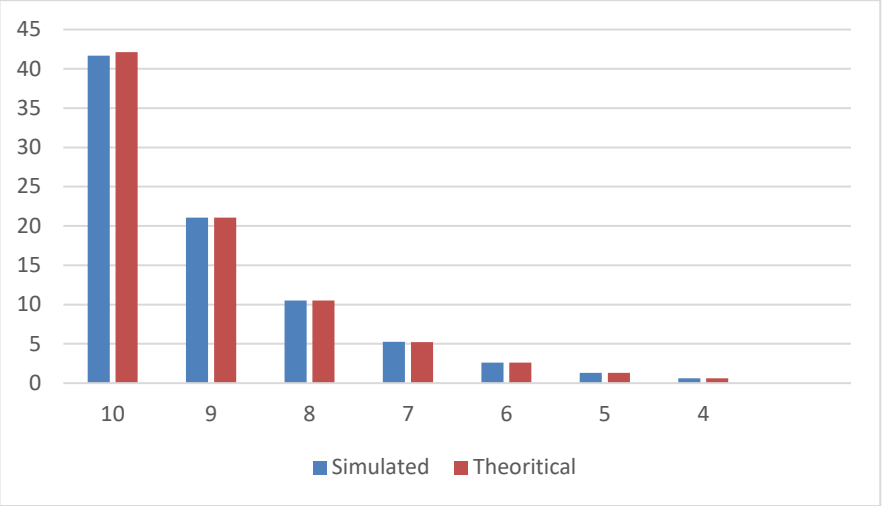
Sample 2 and so on: Vary the Super Frame Order for every sample (SuperFrame order = 9 for sample 2, and…, 4 for sample 7).

The following are the throughputs obtained from simulation for different SuperFrame Orders.

SuperFrame Order	Throughput (Kbps)
10	41.66
9	21.07
8	10.5
7	5.23
6	2.63
5	1.30
4	0.62

To obtain throughput from simulation, payload transmitted values will be obtained from Network statistics and calculated using following formula:

$$Throughput(Kbps) = \frac{Total\ Payload\ transmitted\ to\ destination(Bytes) * 8}{Simulation\ Time(MilliSeconds)}$$



Comparison Chart: All the above plots highly depend upon the placement of Sensor in the simulation environment. So, note that even if the placement is slightly different the same set of values will not be got but one would notice a similar trend.

18.2 Inference:

From the comparison chart both the simulation and theoretical throughputs match except for the case with no inactive period. A sensor will be idle if the last packet in its queue is transmitted. If a packet is generated in inactive period then the packet has to wait in the queue till the next SuperFrame so sensor has packets waiting in its queue and so it cannot be idle in the next SuperFrame, but if there is no inactive period then there might be no packets waiting in the queue and so sensor can be idle resulting in lesser throughput.

19. Understand the working of OSPF

19.1 Objective

To understand the working of OSPF and Shortest Path First (SPF) tree creation

19.2 Theory

OSPF:

Open Shortest Path First (OSPF) is an Interior Gateway Protocol (IGP) standardized by the Internet Engineering Task Force (IETF) and commonly used in large Enterprise networks. OSPF is a link-state routing protocol providing fast convergence and excellent scalability. Like all link-state protocols, OSPF is very efficient in its use of network bandwidth.

Shortest path First Algorithm:

OSPF uses a shorted path first algorithm in order to build and calculate the shortest path to all known destinations. The shortest path is calculated with the use of the Dijkstra algorithm. The algorithm by itself is quite complicated. This is a very high level, simplified way of looking at the various steps of the algorithm:

- Upon initialization or due to any change in routing information, a router generates a link-state advertisement. This advertisement represents the collection of all link-states on that router.
- All routers exchange link-states by means of flooding. Each router that receives a link-state update should store a copy in its link-state database and then propagate the update to other routers.
- After the database of each router is completed, the router calculates a Shortest Path Tree to all destinations. The router uses the Dijkstra algorithm in order to calculate the shortest path tree. The destinations, the associated cost and the next hop to reach those destinations form the IP routing table.
- In case no changes in the OSPF network occur, such as cost of a link or a network being added or deleted, OSPF should be very quiet. Any changes that occur are communicated through link-state packets, and the Dijkstra algorithm is recalculated in order to find the shortest path.

The algorithm places each router at the root of a tree and calculates the shortest path to each destination based on the cumulative cost required to reach that destination. Each router will have its own view of the topology even though all the routers will build a shortest path tree using the same link-state database.

Example:

Refer Pg. no.18 from OSPF RFC 2328 (https://tools.ietf.org/html/rfc2328#section-2.3)

The below network shows a sample map of an Autonomous System

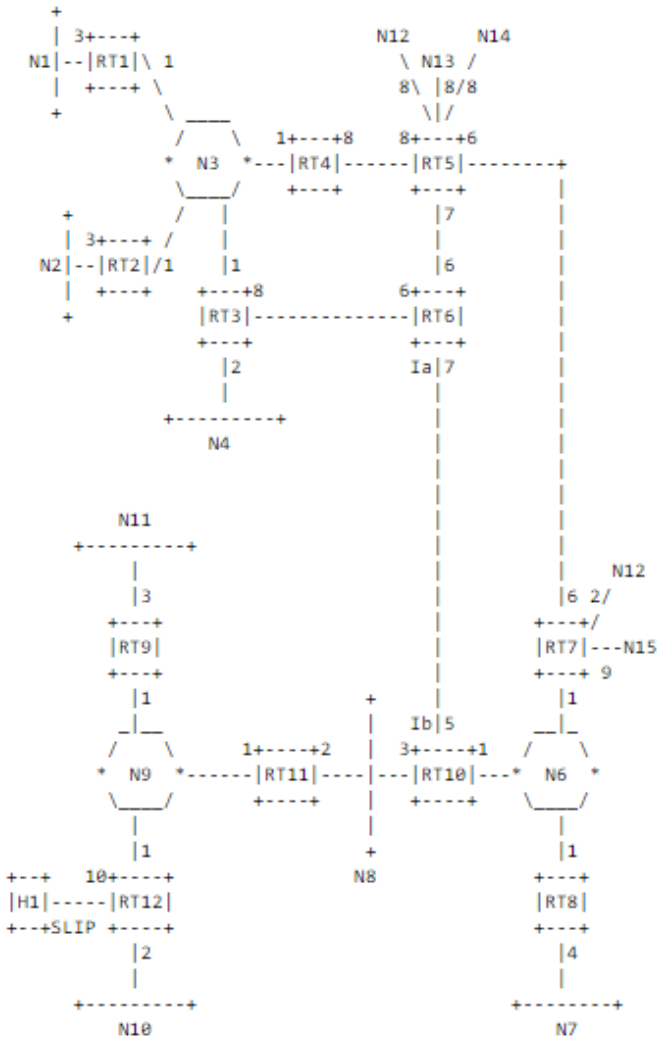


Fig 1. Sample Autonomous system

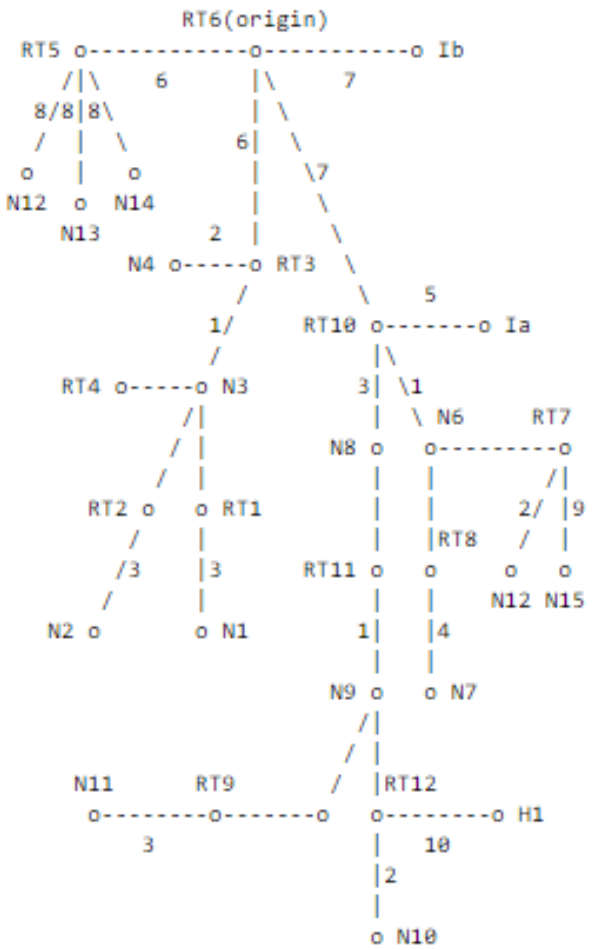
A cost is associated with the output side of each router interface. This cost is configurable by the system administrator. The lower the cost, the more likely the interface is to be used to forward data traffic. Costs are also associated with the externally derived routing data (e.g., the BGP-learned routes).
The directed graph resulting from the above network is depicted in the following table. Arcs are labelled with the cost of the corresponding router output interface. Arcs having no labelled cost have a cost of 0. Note that arcs leading from networks to routers always have cost 0.

	FROM															
		RT1	RT2	RT3	RT4	RT5	RT6	RT7	RT8	RT9	RT 10	RT 11	RT 12	N3	N6	N8	N9
	RT1													0			
	RT2													0			
	RT3						6							0			

TO	**RT4**					8								0			
	RT5				8		6	6									
	RT6			8		7											
	RT7					6									0		
	RT8														0		
	RT9																0
	RT10						7								0	0	
	RT11															0	0
	RT12																0
	N1	3															
	N2		3														
	N3	1	1	1	1												
	N4			2													
	N5																
	N6							1	1		1						
	N7								4								
	N8										3	2					
	N9									1		1	1				
	N10												2				
	N11									3							
	N12					8		2									
	N13					8											
	N14					8											
	N15							9									
	H1												10				

Table 1 Directed graph

A router generates its routing table from the above directed graph by calculating a tree of shortest paths with the router itself as root. Obviously, the shortest-path tree depends on the router doing the calculation. The shortest-path tree for Router RT6 in our example is depicted in the following figure.



SPF tree for Router 6

Routing Table

The tree gives the entire path to any destination network or host. However, only the next hop to the destination is used in the forwarding process. Note also that the best route to any router has also been calculated. For the processing of external data, we note the next hop and distance to any router advertising external routes. The resulting routing table for Router RT6 is shown in the following table

Destination	Next hop	Distance
N1	RT3	10
N2	RT3	10
N3	RT3	7
N4	RT3	8
N6	RT10	8
N7	RT10	12
N8	RT10	10
N9	RT10	11
N10	RT10	13
N11	RT10	14
H1	RT10	21
RT5	RT5	6
RT7	RT10	8
N12	RT10	10
N13	RT5	14

N14	RT5	14
N15	RT10	17

Routing Table for RT6

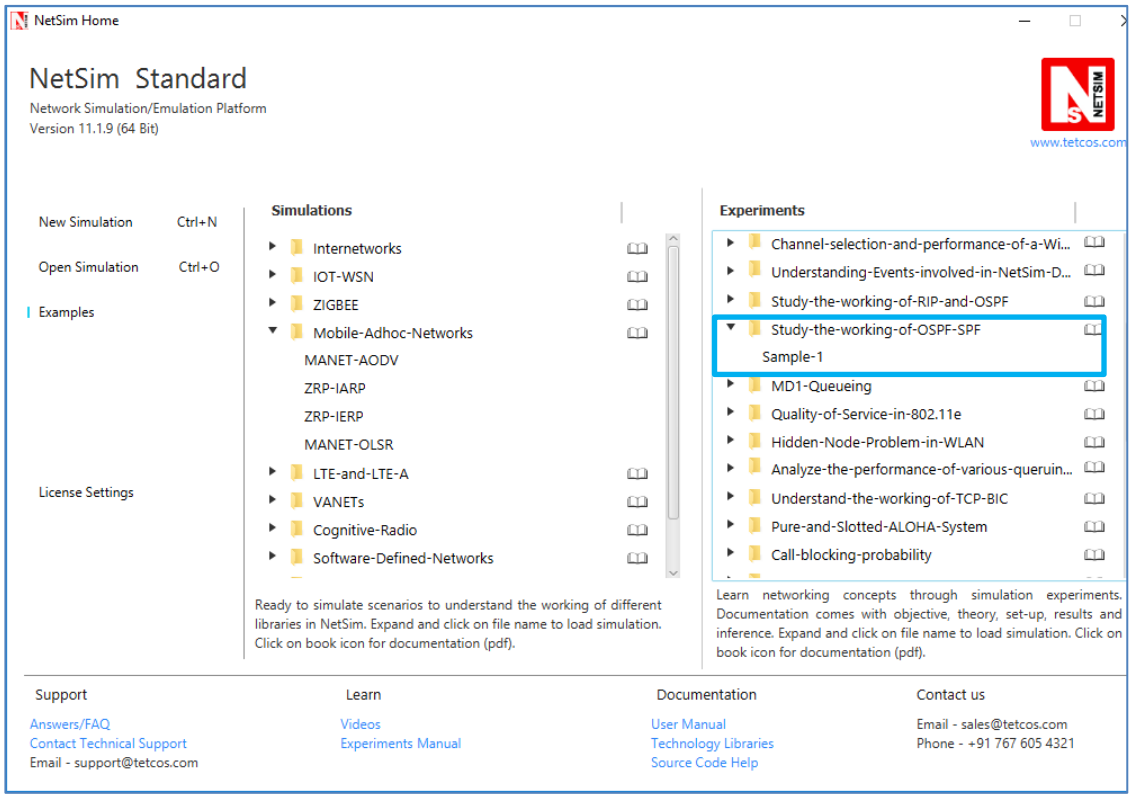
Distance calculation:

Router6 has 3 interfaces i.e. RT3, RT5 and RT10. The distance obtained is 10 for destination N1 via RT3 interface. The packets from Router6 would reach N1 via RT3, N3 and RT1. The cost assigned to routers in this path is 6+1+3 = 10 (cost can be seen in SPF tree for Router6). This is how distance is calculated.

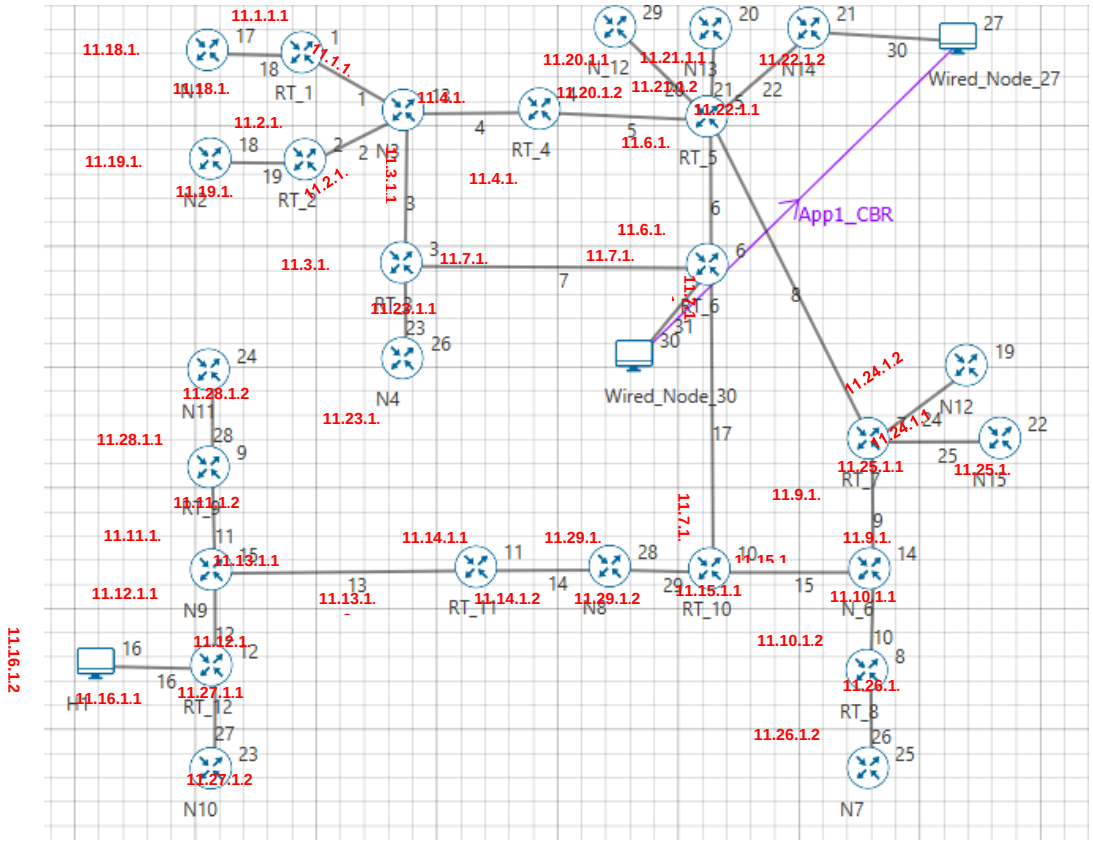
19.3 Procedure

Network Setup

Open Examples → Study-the-working-of-OSPF-SFP as shown below:



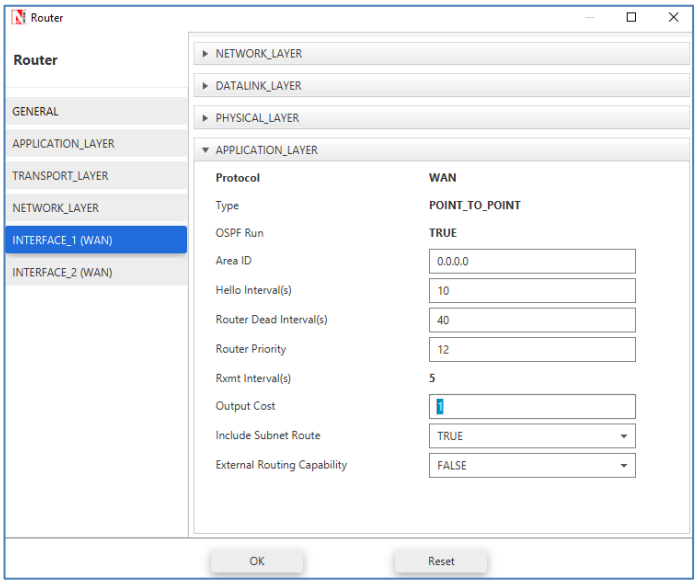
Create a network as explained in the figure 1



The above network was created in NetSim and it is similar to the network as per the OSPF RFC 2328 (Refer Pg. no. 19 - https://tools.ietf.org/html/rfc2328#page-23)

Settings done in the network

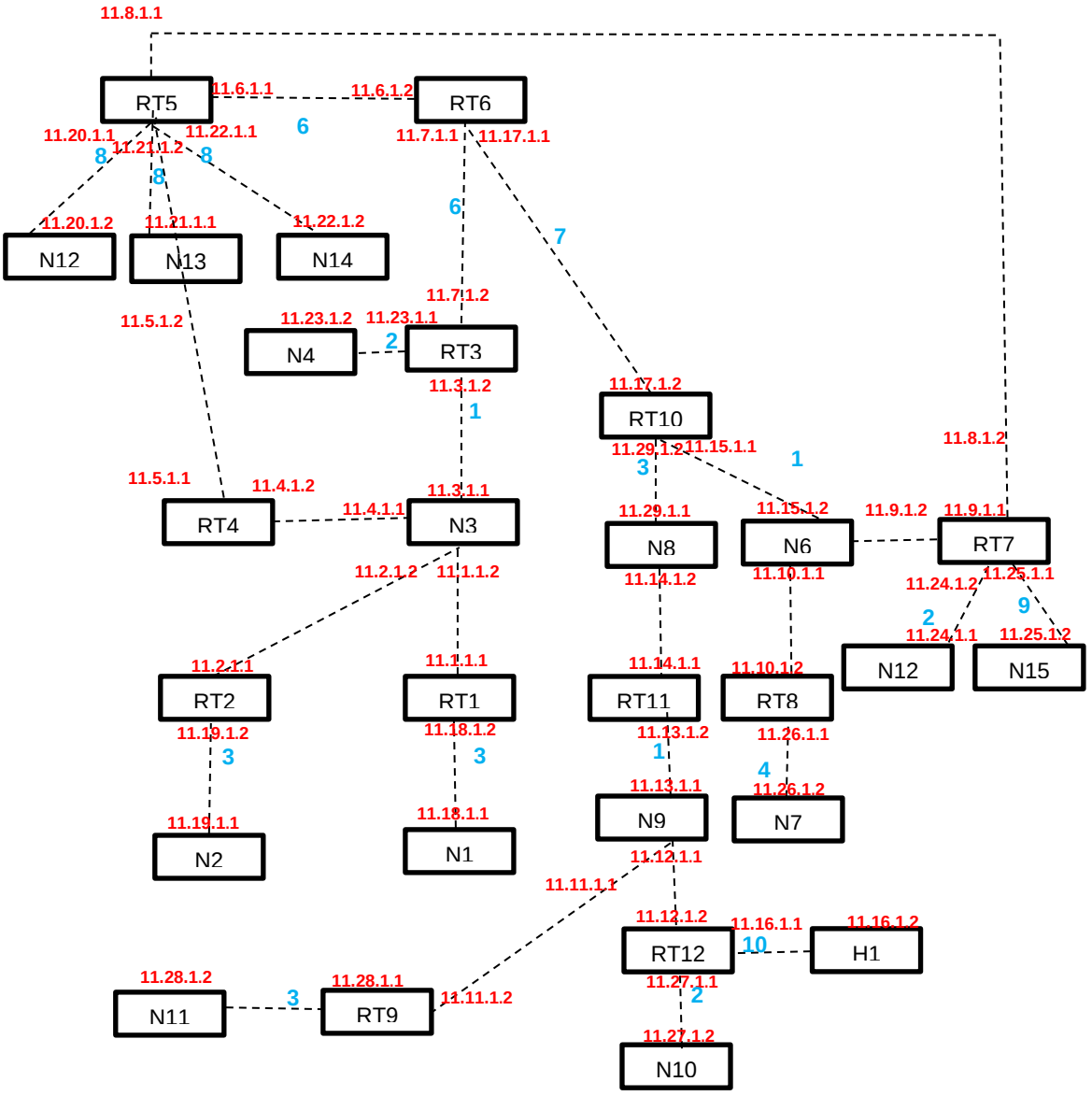
1. Configure CBR application with Source as 30, Destination as 27 and Application start time as 30s
2. Set the Output Cost for all the Routers in the network under router interface properties as per the network shown in Figure 1



3. Enable Packet Trace and run simulation for 50s

19.4 Output

The following is the shortest path first tree created in NetSim



SPF tree for Router 6

In the above screenshot, red color information represents the interface ip addresses of routers and the blue color represents the cost.

NOTE: NetSim, does not implement Link type3 (Link to Stub Network). Hence users would notice a slight difference between the SPF trees of RFC and NetSim.

The IP forwarding table formed in the routers can be accessed from the IP_Forwarding_Table list present in the Simulation Results window as shown below:

RT_6_Table

RT_6 Detailed View

Network Destination	Netmask/Prefix len	Gateway	Interface	Metrics	Type
11.3.1.2	255.255.0.0	11.7.1.2	11.7.1.1	7	OSPF
11.1.1.2	255.255.0.0	11.7.1.2	11.7.1.1	7	OSPF
11.2.1.2	255.255.0.0	11.7.1.2	11.7.1.1	7	OSPF
11.3.1.1	255.255.0.0	11.7.1.2	11.7.1.1	7	OSPF
11.4.1.1	255.255.0.0	11.7.1.2	11.7.1.1	7	OSPF
11.15.1.1	255.255.0.0	11.17.1.2	11.17.1.1	8	OSPF
11.23.1.1	255.255.0.0	11.7.1.2	11.7.1.1	8	OSPF
11.9.1.2	255.255.0.0	11.17.1.2	11.17.1.1	8	OSPF
11.10.1.1	255.255.0.0	11.17.1.2	11.17.1.1	8	OSPF
11.15.1.2	255.255.0.0	11.17.1.2	11.17.1.1	8	OSPF
11.4.1.2	255.255.0.0	11.7.1.2	11.7.1.1	8	OSPF
11.1.1.1	255.255.0.0	11.7.1.2	11.7.1.1	8	OSPF
11.23.1.2	255.255.0.0	11.7.1.2	11.7.1.1	8	OSPF
11.2.1.1	255.255.0.0	11.7.1.2	11.7.1.1	8	OSPF
11.9.1.1	255.255.0.0	11.17.1.2	11.17.1.1	9	OSPF
11.10.1.2	255.255.0.0	11.17.1.2	11.17.1.1	9	OSPF
11.29.1.2	255.255.0.0	11.17.1.2	11.17.1.1	10	OSPF
11.18.1.2	255.255.0.0	11.7.1.2	11.7.1.1	10	OSPF
11.19.1.2	255.255.0.0	11.7.1.2	11.7.1.1	10	OSPF
11.24.1.2	255.255.0.0	11.17.1.2	11.17.1.1	10	OSPF
11.14.1.2	255.255.0.0	11.17.1.2	11.17.1.1	10	OSPF
11.29.1.1	255.255.0.0	11.17.1.2	11.17.1.1	10	OSPF
11.18.1.1	255.255.0.0	11.7.1.2	11.7.1.1	10	OSPF
11.19.1.1	255.255.0.0	11.7.1.2	11.7.1.1	10	OSPF
11.24.1.1	255.255.0.0	11.17.1.2	11.17.1.1	10	OSPF
11.13.1.2	255.255.0.0	11.17.1.2	11.17.1.1	11	OSPF
11.8.1.1	255.255.0.0	11.6.1.1	11.6.1.2	12	OSPF
11.17.1.2	255.255.0.0	11.17.1.2	11.17.1.1	12	OSPF
11.26.1.1	255.255.0.0	11.17.1.2	11.17.1.1	12	OSPF
11.14.1.1	255.255.0.0	11.17.1.2	11.17.1.1	12	OSPF
11.26.1.2	255.255.0.0	11.17.1.2	11.17.1.1	12	OSPF
11.6.1.1	255.255.0.0	11.6.1.1	11.6.1.2	13	OSPF
11.7.1.2	255.255.0.0	11.7.1.2	11.7.1.1	14	OSPF
11.5.1.2	255.255.0.0	11.6.1.1	11.6.1.2	14	OSPF
11.20.1.2	255.255.0.0	11.6.1.1	11.6.1.2	14	OSPF
11.21.1.2	255.255.0.0	11.6.1.1	11.6.1.2	14	OSPF
11.22.1.1	255.255.0.0	11.6.1.1	11.6.1.2	14	OSPF
11.8.1.2	255.255.0.0	11.17.1.2	11.17.1.1	14	OSPF
11.20.1.1	255.255.0.0	11.6.1.1	11.6.1.2	14	OSPF
11.21.1.1	255.255.0.0	11.6.1.1	11.6.1.2	14	OSPF
11.22.1.2	255.255.0.0	11.6.1.1	11.6.1.2	14	OSPF
11.5.1.1	255.255.0.0	11.7.1.2	11.7.1.1	15	OSPF
11.25.1.1	255.255.0.0	11.17.1.2	11.17.1.1	17	OSPF
11.25.1.2	255.255.0.0	11.17.1.2	11.17.1.1	17	OSPF
11.31.0.0	255.255.0.0	on-link	11.31.1.1	300	LOCAL
11.17.0.0	255.255.0.0	on-link	11.17.1.1	300	LOCAL
11.7.0.0	255.255.0.0	on-link	11.7.1.1	300	LOCAL

In this network, Router6 has 3 interfaces with IP's 11.7.1.1, 11.6.1.2 and 11.17.1.1 and its network addresses are 11.7.0.0, 11.6.0.0 and 11.17.0.0 since its network mask is 255.255.0.0

From the above screenshot, the router forwards packets intended to the subnet:

- 11.1.1.2, 11.2.1.2, 11.3.1.2, 11.3.1.1, 11.4.1.1 via interface 11.7.1.1 with cost 7 (6+1)
- Similarly 11.23.1.1, 11.4.1.2, 11.1.1.1, 11.2.1.1, 11.23.1.2 via interface 11.7.1.1 with cost 8 (6+1+1)
- 11.15.1.1, 11.19.1.2, 11.10.1.1, 11.15.1.2 via interface 11.17.1.1 with cost 8 (7+1)
- 11.9.1.1, 11.10.1.2 via interface 11.17.1.1 with cost 9 (7+1+1)
- 11.29.1.2, 11.29.1.1 and 11.14.1.2 via interface 11.17.1.1 with cost 10 (7+3)
- 11.24.1.2, 11.24.1.1 via interface 11.17.1.1 with cost 10 (7+1+2)
- 11.18.1.2, 11.18.1.1, 11.19.1.2 and 11.19.1.1 via interface 11.7.1.1 with cost 10 (6+1+3)
- 11.13.1.2 via interface 11.17.1.1 with cost 11 (7+3+1)
- 11.8.1.1 via interface 11.6.1.2 with cost 12 (6+6)
- 11.17.1.2 via interface 11.17.1.1 with cost 12 (7+5)
- 11.26.1.1 and 11.26.1.2 via interface 11.17.1.1 with cost 12 (7+1+4)
- 11.14.1.1 via interface 11.17.1.1 with cost 12 (7+3+2)
- 11.6.1.1 via interface 11.6.1.2 with cost 13 (7+6)
- 11.7.1.2 via interface 11.7.1.1 with cost 14 (8+6)
- 11.5.1.2 via interface 11.6.1.2 with cost 14 (6+8)
- 11.20.1.2, 11.20.1.1, 11.21.1.1, 11.21.1.2, 11.22.1.1, 11.22.1.2 via interface 11.6.1..2 with cost 14 (8+6)
- 11.8.1.2 via interface 11.17.1.1 with cost 14 (7+1+6)
- 11.25.1.1, 11.25.1.2 via interface 11.17.1.1 with cost 17 (7+1+9)
- 11.5.1.1 via interface 11.7.1.1 with cost 15 (6+1+8)

We are thus able to simulate the exact example as provided in the RFC and report that SPF Tree obtained and the routing costs match the analysis provided in the RFC

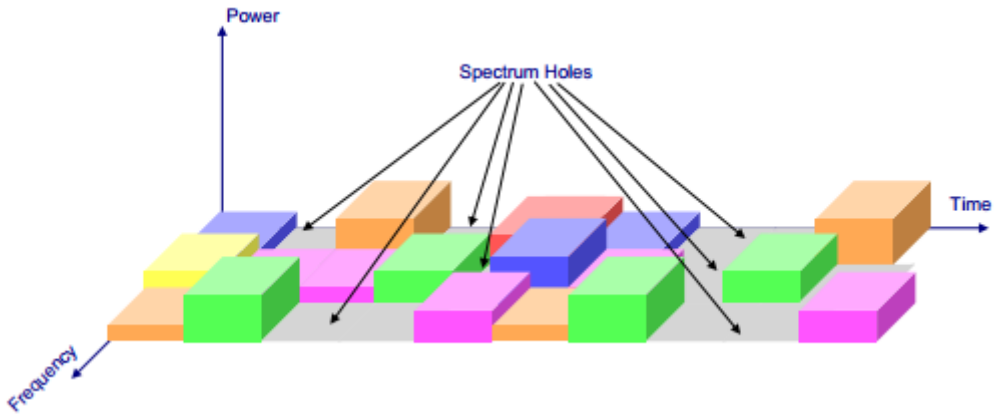
20. To analyze how the allocation of frequency spectrum to the Incumbent (Primary) and CR CPE (Secondary User) affects throughput

20.1 Introduction:

An important component of the cognitive radio concept is the ability to measure, sense, learn, and be aware of the parameters related to the radio channel characteristics, availability of spectrum and power, radio's operating environment, user requirements and applications, available networks (infrastructures) and nodes, local policies and other operating restrictions.

NetSim simulator models IEEE 802.22 Cognitive Radio per the theory explained below.

A spectrum hole has been defined as a band of frequencies assigned to a primary user, but at a particular time and specific geographic location, the band is not being utilized by that user. Cognitive Radio was proposed as the means to promote the efficient use of spectrum by exploiting the existence of spectrum holes.



These spectrum holes are used by the SU for its transmission. This scheme is often referred to as opportunistic spectrum access (OSA). No concurrent transmission of the PU and the SU is allowed. The SU must vacate the channel as soon as the PU reappears, which leads to the forced termination of the SU connection (if there is no other available channel for the SU). Since the SU has no control over the resource availability, the transmission of the SU is blocked when the channel is occupied by the PU. The forced termination and blocking of a SU connection is shown in the below figure. The forced termination probability and blocking probability are the key parameters which determine the throughput of the SU, and thus its viable existence. The forced termination depends on the traffic behavior of the PUs and the SUs (e.g. arrival rates, service time etc.). In the case of multiple SU groups with

different traffic statistics, the forced termination and blocking probabilities lead to unfairness among the SU groups.

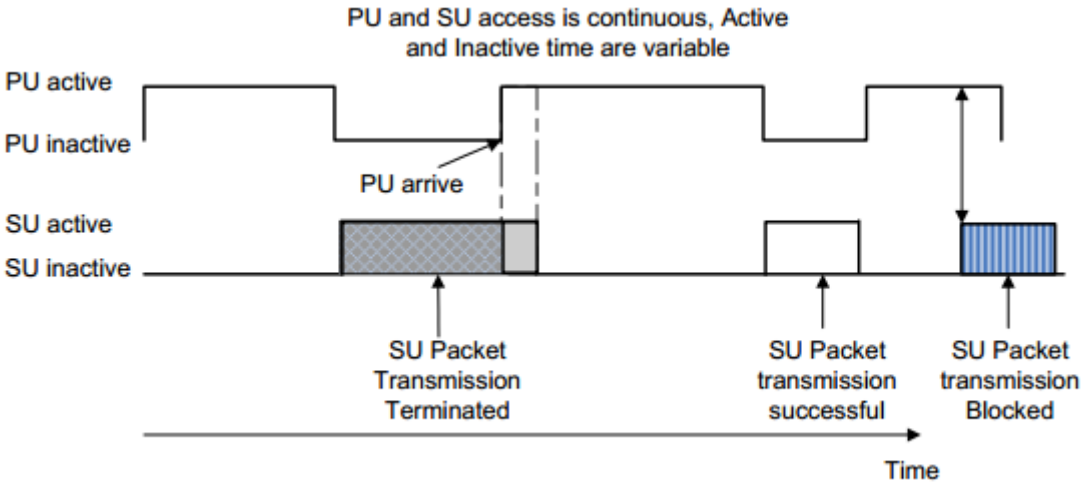


Illustration of forced termination and blocking

Performance metrics:

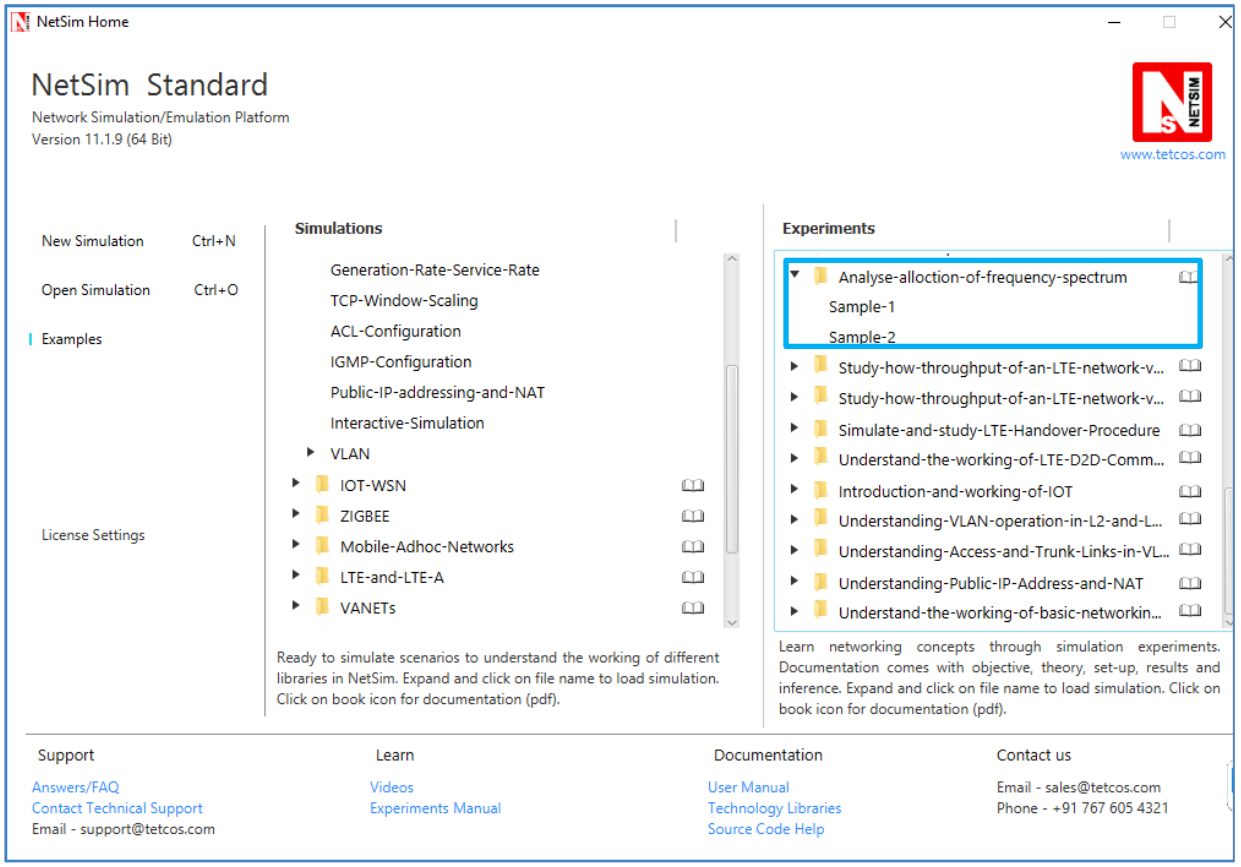
The different parameters used to analyze the performance are explained as follows:

- **Throughput**: It is the rate of successfully transmitted data packets in unit time in the network during the simulation.
- **Spectral Efficiency:** It refers to the information rate that can be transmitted over a given bandwidth in a specific communication system. It is a measure of how efficiently a limited frequency spectrum is utilized by the physical layer protocol, and sometimes by the media access control protocol.

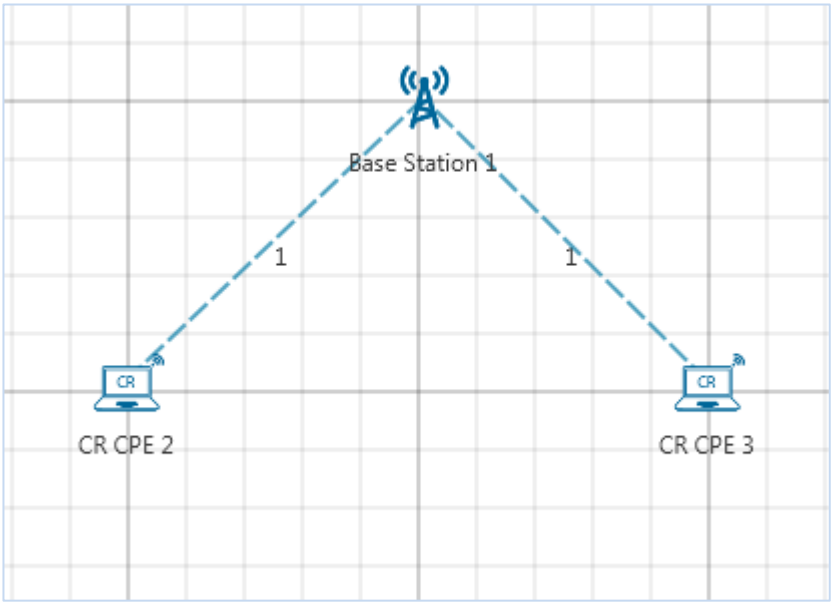
20.2 Network Set Up

Sample 1:

Step 1: In NetSim, Open Examples →Analyze-allocation-of-frequency-spectrum as shown below:



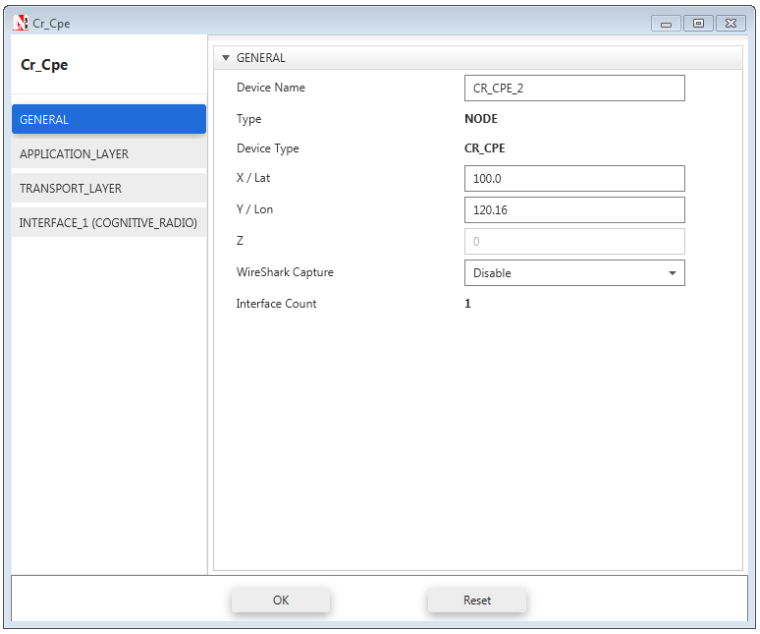
Step 2: Click & drop 1 Base Station, Adhoc link and 2 CR CPE onto the Simulation Environment. To edit the position, change the (x, y) co-ordinates in Global Properties as shown:



Editing Device Co-ordinates:

- Go to device properties by right-clicking on the device and selecting properties.
- Under General, X/Lat and Y/Lat parameters can be modified as required.

Disable TCP in all Node



Arrange the positions of the nodes as per the following table:

BS	X-Coordinate	Y-Coordinate
1	100	100

CR CPE	X-Coordinate	Y-Coordinate
2	100	120
3	120	100

Step 3: Base Station Properties: In Interface_CR properties, under the Incumbent1 section, set

Incumbent	X-Coordinate	Y-Coordinate
1	90	90

Operational_Frequency_Start (MHz) = 54

Operational_Frequency_End (MHz) = 60

ON_Duration (s) = 10

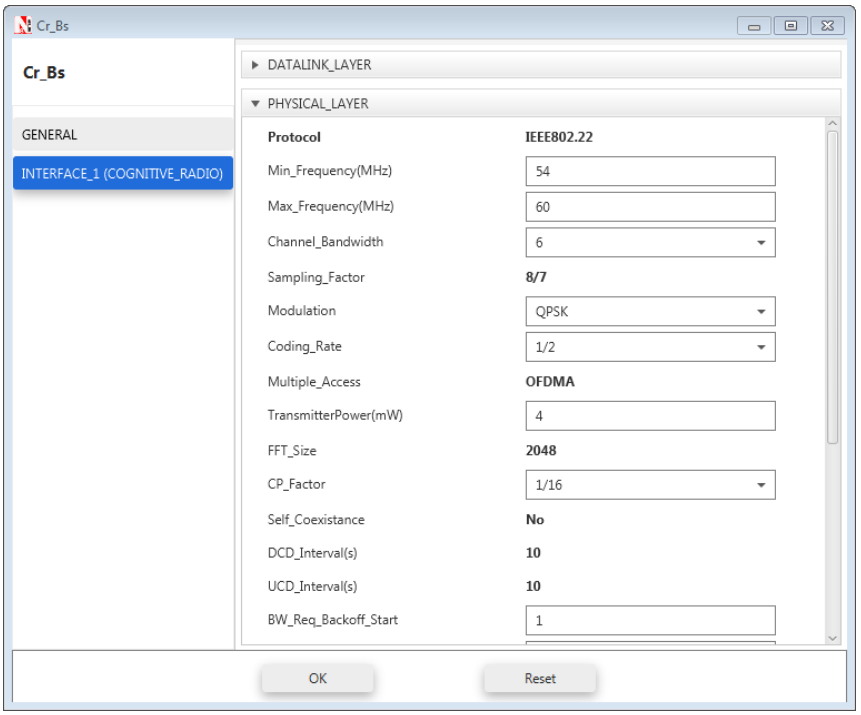
OFF_Duration (s) = 0

Keepout_Distance (m) = 100

Under the **Physical Layer** section, Set

Min_Frequency (MHz) = 54

Max_Frequency (MHz) = 60



Step 4:

Application Properties:

To add application, click on the Application icon. Edit the Application properties as given in table. All other properties are default.

Application Properties	Application 1
Application Type	Custom
Source ID	2
Destination ID	3

Note: ***Click on the Run Simulation icon and set simulation time only after doing the following tasks:***

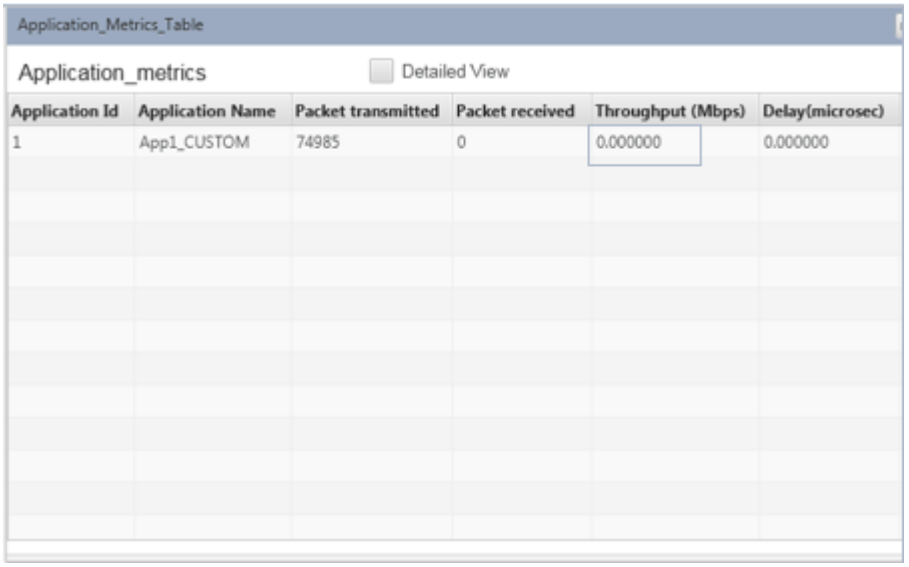
- ***Set the properties of Nodes and***
- ***Configure Applications***.

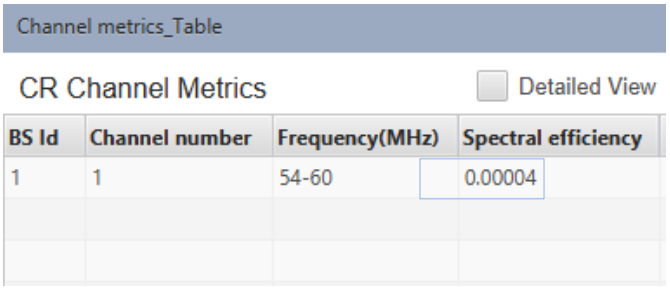
Simulation Time- 100 Seconds

Output:

- Go to Application metrics and check the value of throughput.
- **Throughput = Throughput of the application.**

- Go to CR Channel metrics at the end in metrics and check the **Spectral Efficiency**.





Sample 2:

Perform the same steps as in Sample 1 with following changes:

Change1: Base Station Properties

In the **Physical Layer** section, Set Min_Frequency = 54 and Max_Frequency = 90

Note: Open the saved experiments (Please Refer at the end of experiment) to obtain the various Performance metrics and note the values obtained to compare as shown in table.

Comparison Table:

	Sample 1	Sample 2
Throughput(Mbps)	0.000000	0.583

Spectral Efficiency-

Sample 1

Frequency(MHz)	Spectral efficiency
54-60	0.00004

Sample 2

Frequency(MHz)	Spectral efficiency
54-60	0.00004
60-66	0.00000
66-72	0.00010
72-78	0.19720
78-84	0.00510
84-90	0.15571

20.3 Inference:

In both the samples, the Secondary User (CR-CPE) lies within the operational region of Primary User (Incumbent), hence the frequency spectrum used by operational Primary User (Incumbent) will not be used by Secondary User (CR-CPE). Also the Operational Interval under Incumbent is set to zero, i.e., the Incumbent will continuously use the channel allocated to it.

In the first sample, both the Primary User (Incumbent) and the Secondary User (CR-CPE) has been allocated the same channel (frequency band of 54 - 60 MHz). As Incumbent will continuously use the channel allocated to it, so there will be no Spectrum Hole, hence the secondary user will not be able to transmit any data in an opportunistic manner. Therefore the throughput of the application in the CR-CPE and the spectral efficiency is almost equal to zero.

In the second sample, the Primary User (Incumbent) has been allocated frequency band of 54 - 60 MHz and the Secondary User (CR-CPE) has been allocated the frequency band of 54 - 90 MHz. Incumbent will continuously use the channel allocated to it, but the rest channels will remain free i.e. there will be Spectrum Hole, which the CR-CPE will utilize to transmit data.

NOTE: The results are highly dependent on position/velocity/ traffic etc. Any modifications with the above mentioned input parameters will change the final output result.

21. Study how the throughput of LTE network varies as the distance between the ENB and UE (User Equipment) is increased

21.1 Theory:

LTE or Long Term Evolution, commonly known as 4G LTE, is a standard for wireless communication of high-speed data for mobile phones and data terminals. It is based on the GSM/EDGE and UMTS/HSPA network technologies, increasing the capacity and speed using a different radio interface.

The path loss in LTE is the decay of the signal power dissipated due to radiation on the wireless channels. Path loss may be due to many effects, such as free space loss, refraction, diffraction, reflection, aperture-medium coupling loss, and absorption.

Received power (**P_r**) can be calculated as:

Case 1: When no path loss Received power is same as Transmitted power, i.e., Pr = Pt

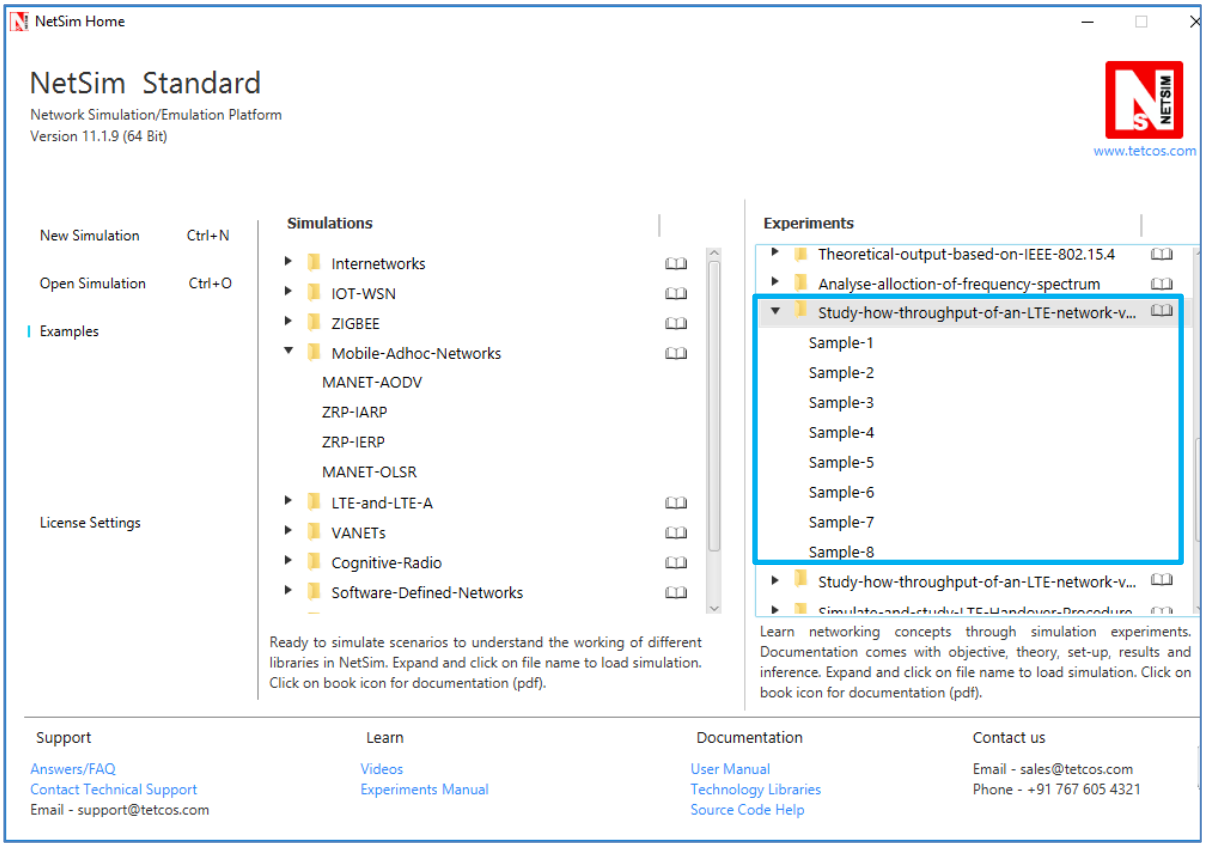
Case 2: When **Line of Sight** is there, Received power P_r is

$$P_r = P_t + G_t + G_r + 20\ \log_{10}(\frac{\lambda}{4\pi d}) + 10\ n\ \log_{10}\frac{d0}{d}$$

Where **G_t** and **G_r** are gains of transmitting and receiving antenna respectively. Here **d** is the distance between transmitter and receiver, **λ** is the wavelength of the transmitted signal and $d0$ is reference distance at which channel gain becomes 1. **n** is path loss exponent and **P_t** is transmitted power.

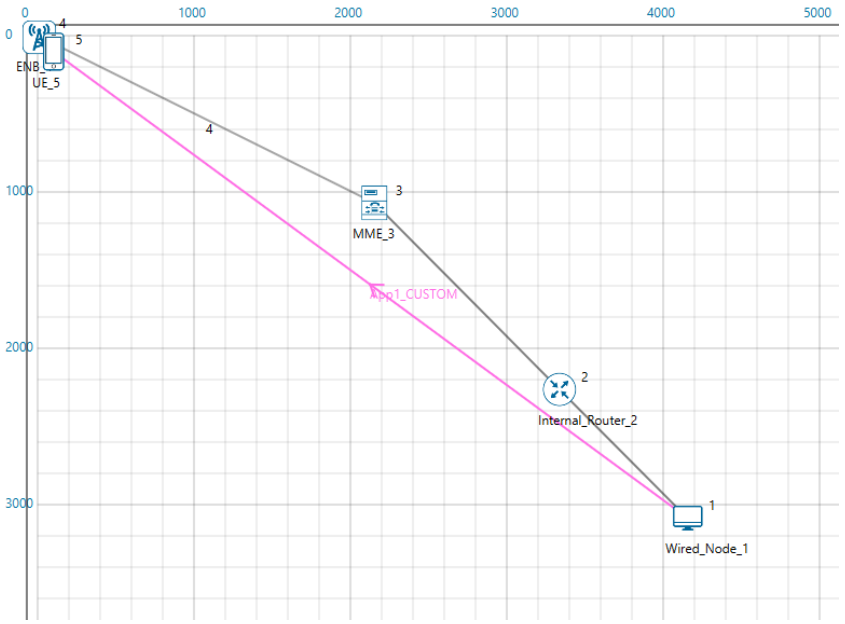
21.2 Procedure:

Open Examples → Study-how-throughput-of-an-LTE-varies-with-Distance as shown below:



Sample Inputs:

In this experiment, 1 Wired Node, 1 Router, 1 MME , 1 ENB and 1 UE is clicked and dropped onto the Simulation environment from tool bar as shown below.



Note: Before placement of any device grid length should be increased and it should be 10000 meters X 10000 meters. Click on Environment Settings present in the ribbon and set grid length as 10000.

Sample 1:

These properties can be set only after devices are linked to each other as shown above.

Wired Node Properties: In Wired Node 1, go to Transport Layer and set TCP as Disable.

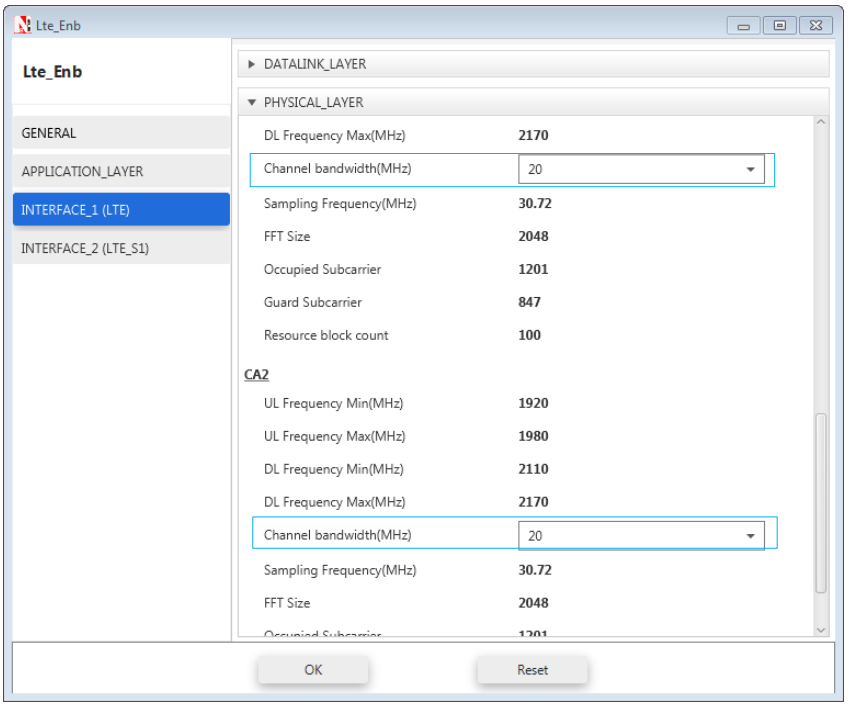
Router Properties: Default properties.

MME Properties: Default properties.

ENB Properties:

ENB Properties	ENB 4
General Properties	
X_Coordinate	**0**
Y_Coordinate	**0**

Set the Channel Bandwidth to 20MHz for both carriers shown below



UE Properties:

UE Properties	UE 5
Global Properties	
X_Coordinate	**50**
Y_Coordinate	**50**
Velocity(m/s)	**0**

To run the simulation, click on the Application icon and change the following properties:

Application Properties	
Application Type	Custom
Source ID	Wired Node 1
Destination ID	UE 5
Packet Size	
Distribution	Constant
Value(Bytes)	1460
Inter Arrival Time	
Distribution	Constant
Value(µs)	165

Wired Link Properties:

Link Properties	Wired Link 2	Wired Link 3	Wired Link 4
Uplink Speed (Mbps)	100	100	100
Downlink Speed (Mbps)	100	100	100
Uplink BER	0	0	0
Downlink BER	0	0	0
Down Time	0	0	0

Wireless Link Properties:

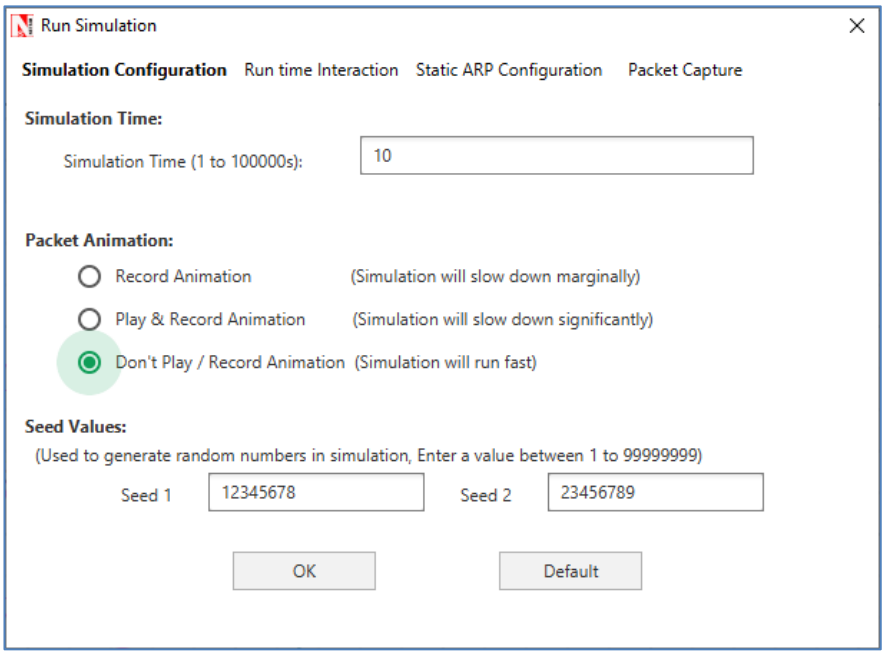
Link Properties	Wireless Link 1
Channel characteristics	**Path Loss Only**
Path Loss Model	**Log Distance**
Path loss Exponent(n)	**4**

Simulation Time – 10 Sec

Note: The Simulation Time can be selected only after the following two tasks,

- ***Set the properties for all the devices and links.***
- ***Click on Run Simulation button***

Set Packet Animation to **Don't Play/Record Animation (Simulation will run fast)** and click Accept. If record animation option is selected, the simulation may take a long time to complete.



Upon completion of the experiment "Save" the experiment in current workspace and note down the Application throughput which is available in Application metrics for each sample case.

Sample 2:

Change the following properties in UE and run the simulation for 10 seconds as above. All other properties are default.

UE Properties:

UE Properties	UE 5
Global Properties	
X_Coordinate	**100**
Y_Coordinate	**100**
Velocity(m/s)	**0**

From Sample 3 to Sample 9:

Change the UE property every time (for all samples) by varying the (x, y) coordinates values as follows:

Change in UE Properties: (x, y)	
Sample 3	(150,150)
Sample 4	(200,200)
Sample 5	(250,250)
Sample 6	(300,300)
Sample 7	(350,350)
Sample 8	(400,400)

And note down the throughput values from the Application metrics in each sample case.

21.3 Output:

Step 1: Distance calculation:

Calculate the Distance between ENB (x_1, y_1) and UE(x_2, y_2) as follows: $\sqrt{(x_2-x_1)^2+ (y_2-y_1)^2}$

For example for Sample 1:

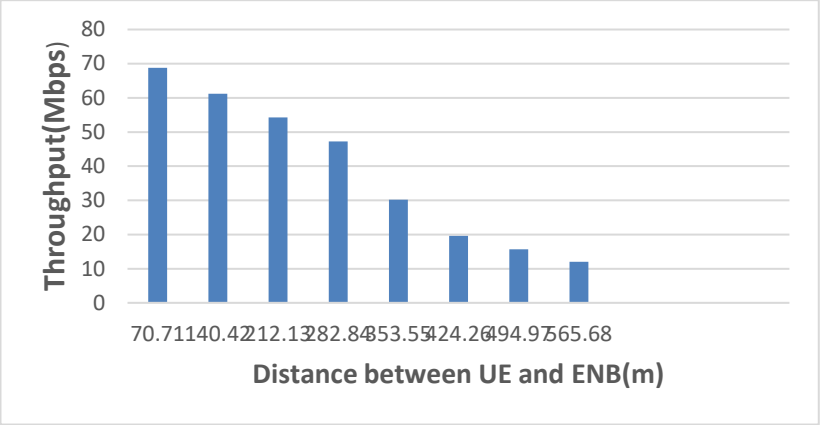
ENB (x_1, y_1) = (0, 0); UE(x_2, y_2) = (50, 50);

Distance = $\sqrt{(50-0)^2+ (50-0)^2}$ **=** $\sqrt{2} \times 50 =$ **$50\sqrt{2}$** meters.

Step 2: Open the Excel file and note down the distance between UE and ENB and throughput values as shown in below table.

Sample	Distance between UE and ENB (meters)			Throughput (Mbps)
1	$50\sqrt{2}$	=	70.71	68.8
2	$100\sqrt{2}$	=	140.42	61.1
3	$150\sqrt{2}$	=	212.13	54.3
4	$200\sqrt{2}$	=	282.84	47.3
5	$250\sqrt{2}$	=	353.55	30.2
6	$300\sqrt{2}$	=	424.26	19.6
7	$350\sqrt{2}$	=	494.97	15.6
8	$400\sqrt{2}$	=	565.68	12.1

Comparison Chart:



To draw these graphs by using Excel “Insert →Chart” option and then select chart type as “Line chart”.

21.4 Inference

As the distance increases between ENB and UE, throughput decreases. The reason is that as the distance increases between the devices, the received signal power decreases, and the LTE Phy Rate drops as the signal power reduces.

22. Study how the throughput of LTE network varies as the Channel bandwidth changes in the ENB (Evolved node)

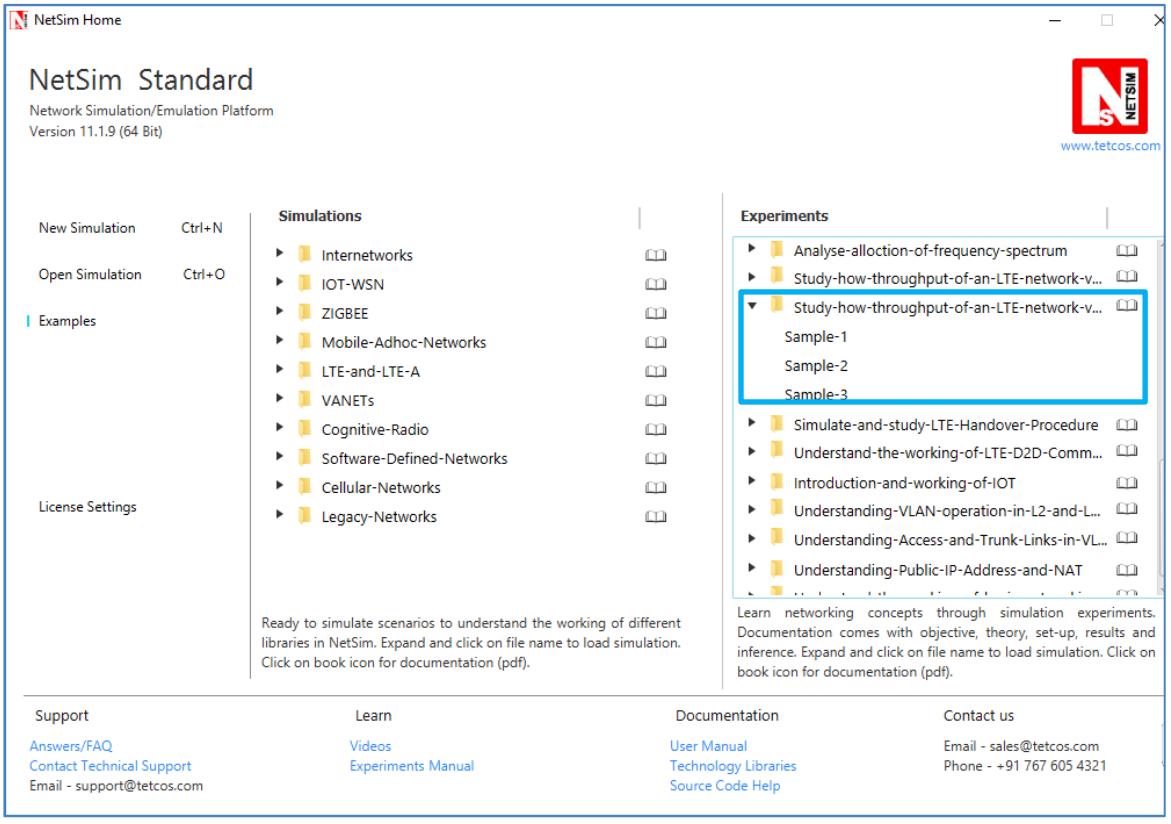
22.1 Theory:

LTE or Long Term Evolution, commonly known as 4G LTE, is a standard for wireless communication of high-speed data for mobile phones and data terminals. It is based on the GSM/EDGE and UMTS/HSPA network technologies, increasing the capacity and speed using a different radio interface.

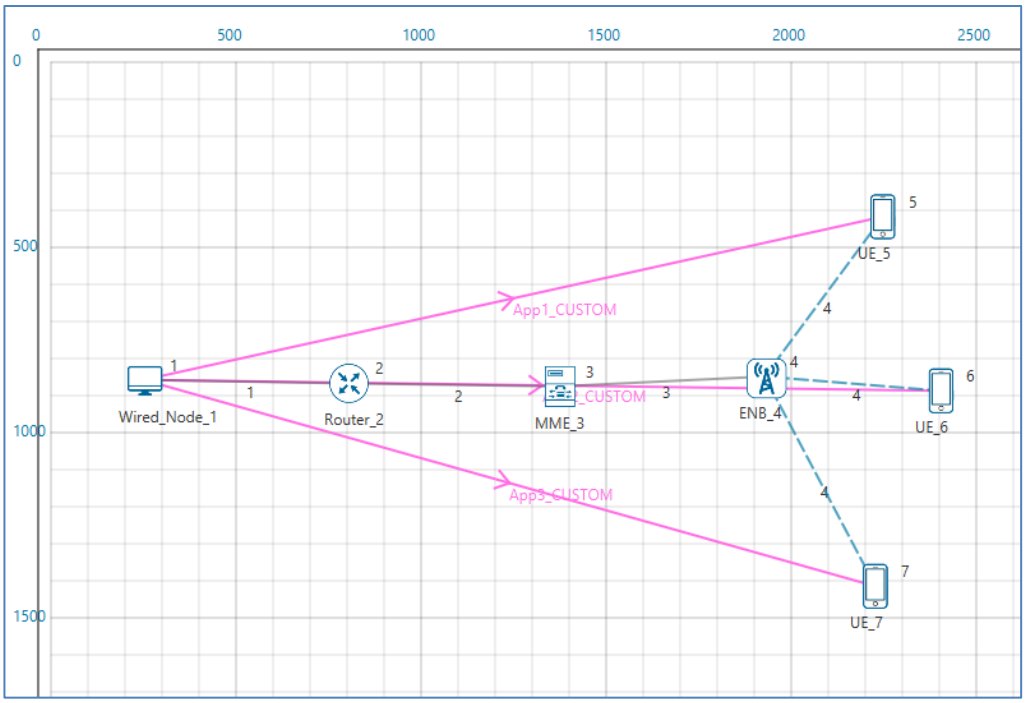
LTE supports flexible carrier bandwidths, from 1.4 MHz up to 20 MHz as well as both FDD and TDD. LTE designed with a scalable carrier bandwidth from 1.4 MHz up to 20 MHz which bandwidth is used depends on the frequency band and the amount of spectrum available with a network operator.

22.2 Procedure:

Step 1: Open Examples → Study-how-throughput-of-an-LTE-varies-with-Channel-Bandwidth as shown below:



Step 2: Click and drop 1 Wired Node ,1 Router, 1 MME , 1 ENB and 3 UE onto the Simulation environment from tool bar as shown below and connect them using wired/wireless links.



Sample 1:

Step 3: Disable TCP in transport layer properties of Wired Node and set default properties for Router and MME.

Step 4: Set ENB properties according to the table given below.

ENB Properties	
Interface_LTE (Physical Layer)	
Carrier aggregation	Inter_Band_Noncontingious_CA
CA1	
Channel Bandwidth (MHz)	10
CA2	
Channel Bandwidth (MHz)	10

Step 5: Set UE properties as shown below. The remaining properties in UE are set to default.

UE Properties	UE 5	UE 6	UE 7
Global Properties			
Velocity(m/s)	**0**	**0**	**0**

Step 6: Set Wired Link properties as per the table given below.

Link Properties	Wired Link 1	Wired Link 2	Wired Link 3
Uplink Speed (Mbps)	1000	1000	1000
Downlink Speed (Mbps)	1000	1000	1000
Uplink BER	0	0	0
Downlink BER	0	0	0

Step 7: Set Wireless Link properties as per the table given below.

Link Properties	Wireless Link 4
Channel characteristics	**No Path Loss**

Step 8: Click on the Application icon present on the ribbon and change the following properties

Application Properties	Application 1	Application 2	Application 3
Application Type	Custom	Custom	Custom
Source ID	Wired Node 1	Wired Node 1	Wired Node 1
Destination ID	UE 5	UE 6	UE 7
Packet Size			
Distribution	Constant	Constant	Constant
Value(Bytes)	1460	1460	1460
Inter Arrival Time			
Distribution	Constant	Constant	Constant
Value(µs)	146	146	146

Step 9: Click on Run Simulation and set simulation time to 10 seconds.

Step 10: Upon completion of the experiment “Save” the experiment in the current workspace and note down the Application throughputs of all applications which is available in Application metrics for each sample case.

Sample 2:

Set ENB properties according to the table given below. All other steps remain the same as per Sample 1.

ENB Properties	
Interface_LTE (Physical Layer)	
Carrier Aggregation	Inter_Band_Noncontingious_CA
CA1	
Channel Bandwidth (MHz)	10
CA2	
Channel Bandwidth (MHz)	5

Sample 3:

Set ENB properties according to the table given below. All other steps remain the same as per Sample 1.

ENB Properties	
Interface_LTE (Physical Layer)	
Carrier aggregation	Inter_Band_Noncontingious_CA
CA1	
Channel Bandwidth (MHz)	5
CA2	
Channel Bandwidth (MHz)	5

22.3 Output

Add the sum of all throughput values in each sample case: Example Sample 1

Application ID	Throughput (mbps)
1	23.177
2	23.177
3	23.177
Sum =	69.531 mbps

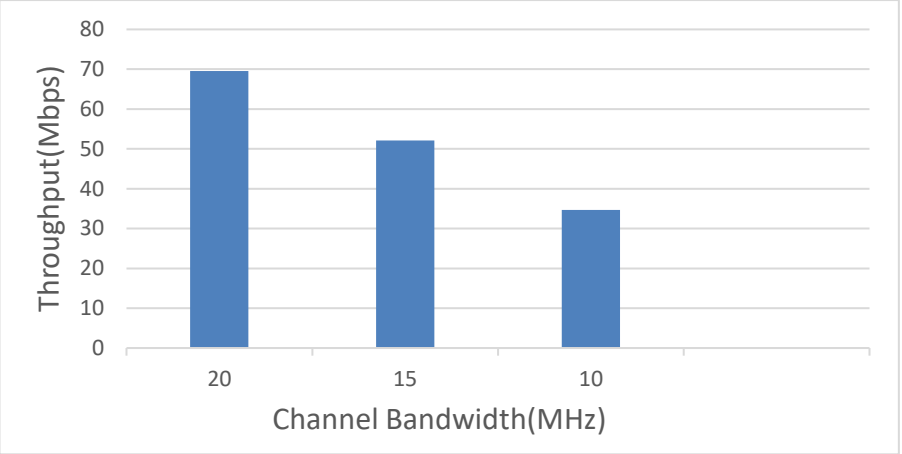
Same procedure can be followed for the other samples.

Open the Excel file and note down the sum of applications throughput values as shown in below table.

Sample	Channel Bandwidth(MHz)	Throughput (Mbps)
1	20	69.53
2	15	52.11
3	10	34.7

Comparison Chart:

To draw these graphs by using Excel “Insert →Chart” option and then select chart type as “Line chart”.



22.4 Inference

LTE provides spectrum flexibility with scalable transmission bandwidth between 1.4 MHz and 20 MHz depending on the available spectrum for flexible radio planning. The 20 MHz bandwidth can provide up to 150 Mbps downlink user data rate and 75 Mbps uplink peak data rate with 2×2 MIMO, and 300 Mbps with 4×4 MIMO.

As the channel bandwidth decreases the number of resource blocks also decreases. If more resource blocks are available then more number of packets can be transmitted.

Channel Bandwidth (MHz)	1.4	3	5	10	15	20
Transmission Bandwidth Configuration NRB: (1 resource block = 180kHz in 1ms TTI)	6	15	25	50	75	100

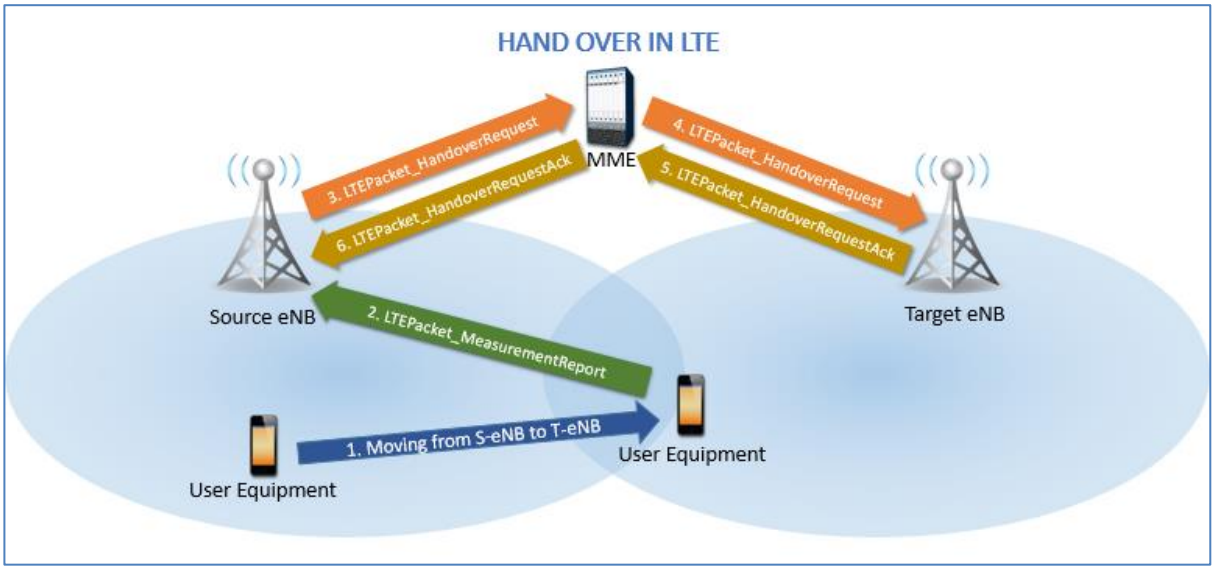
23. Simulate and study LTE Handover procedure

23.1 Introduction

As defined by 3GPP, handover is a procedure for changing the serving cell of a UE. The two eNodeBs involved in the process are typically called the source eNB (S-eNB) and the target eNB (T-eNB). In NetSim, handover procedure is triggered “automatically” by the serving eNodeB of the UE.

23.2 Description and Definitions

1. A data call is established between the UE, S-eNB (Source-eNB) and the network elements. Data packets are transferred to/from the UE to/from the network in both directions (Downlink as well as Uplink)
2. The network sends the MEASUREMENT CONTROL REQ message to the UE to set the parameters to measure and set thresholds for those parameters. Its purpose is to instruct the UE to send a measurement report to the network as soon as it detects the thresholds.
3. The UE sends the MEASUREMENT REPORT to the Serving eNB, which contains the RQRS from all the nearby eNBs. The Serving eNB makes the decision to hand off the UE to a T-eNB (Target-eNB) using the handover algorithm mentioned in the Introduction
4. The S-eNB then initiates the decision to handover using the X2 interface.
5. The S-eNB issues a HANDOVER REQUEST message to the T-eNB passing necessary information to prepare the handover at the target side
6. The T-eNB sends back the HANDOVER REQUEST ACKNOWLEDGE message including a transparent container to be sent to the UE as an RRC message to perform the handover.
7. The S-eNB generates the RRC (Radio resource control used for signaling transfer) message to perform the handover, i.e., RRC CONNECTION RECONFIGURATION message including the mobility Control Information.
8. The S-eNB starts forwarding the downlink data packets to the T-eNB for all the data bearers which are being established in the T-eNB during the HANDOVER REQ message processing.
9. The T-eNB now requests the S-eNB to release the resources. With this, the handover procedure is complete.

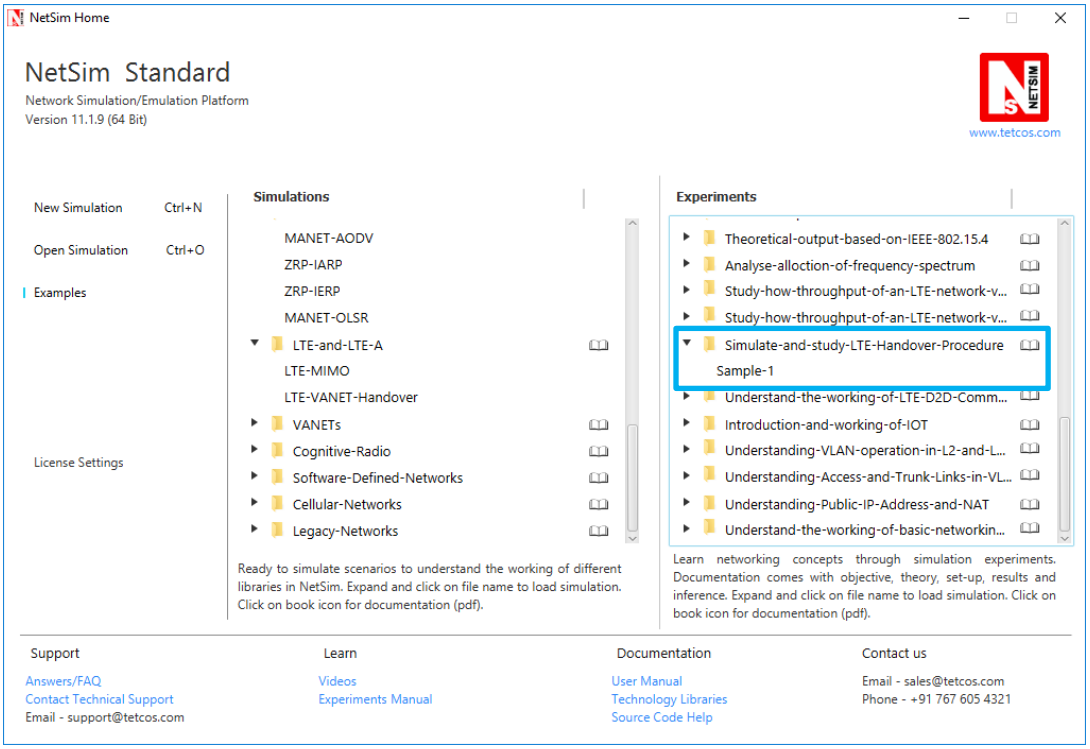


23.3 Analysis/Algorithm

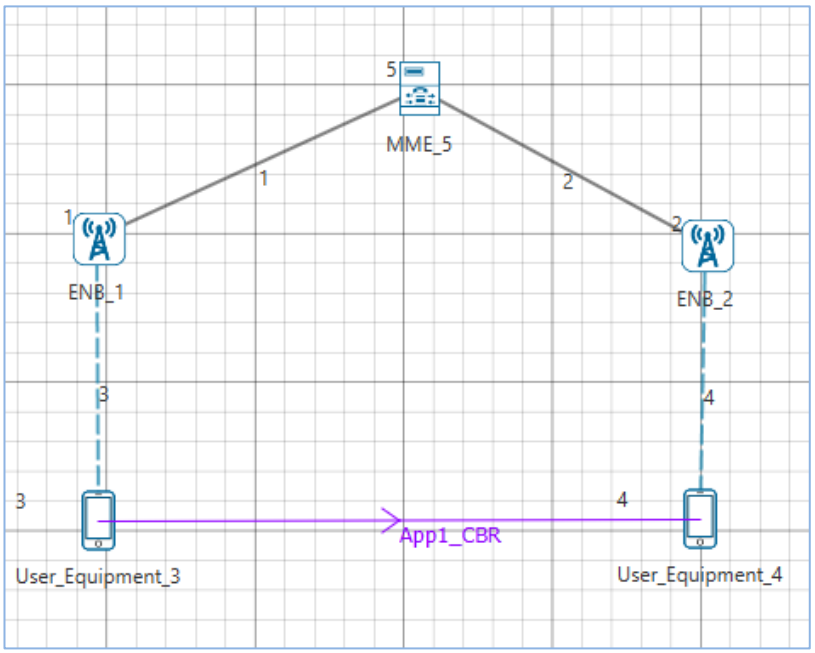
NetSim handover algorithm utilizes the Reference Signal Received Quality (RSRQ) measurements, to trigger the handover. When the target eNB's RSRQ crosses the serving eNB's RSRQ by a factor know as margin of handover (equal to 3dB), hand over is triggered.

23.4 Network Set-up

Step 1: Open Examples → Simulate-and-Study-LTE-Handover-Procedure as shown below:



Step 2: Set Grid length to 5000m*5000m. Click & drop 2 eNB's, 2 UE's and 1 MME onto the Simulation Environment and connect them as per the following:



eNB Properties:

eNB 1: X-Co-ordinate = 1000m, Y-Co-ordinate = 1500m

eNB 2: X-Co-ordinate = 4000m, Y-Co-ordinate = 1500m

Accept default properties for eNB and MME.

Wireless Link Properties:

Propagation model->Path Loss only, Path Loss Model→Log Distance and Path loss Exponent→3

Application Properties:

To add application, click on the Application icon present on the grid. Edit the Application properties and set the Application Type as CBR with the Source ID = 3 and Destination ID = 4. All other properties are default. Set Start Time=20

UE Properties:

UE 3: X-Co-ordinate = 1000m, Y-Co-ordinate = 3000m, Mobility_Model = FILE_BASED_MOBILITY

UE 4: X-Co-ordinate = 4000m, Y-Co-ordinate = 3000m, Mobility_Model = FILE_BASED_MOBILITY

Note: In File Based Mobility, users can write their own custom mobility models and define the movement of the mobile users. Create a mobility.txt file for UE's involved in mobility with each step

equal to 0.5 sec with velocity 50 m/s or copy the file from the Docs folder of NetSim Install directory < C:\Program Files\NetSim Standard\Docs\Sample_Configuration\NetSim_Experiment_Manual\Experiment-23-LTE-Handover\Sample-1> and place it inside the bin folder of NetSim <C:\Program Files\NetSim Standard\bin>. For more information, please refer section 3.3 “Mobility models in NetSim” under MANETs Technology Library as shown below:

The NetSim Mobility File format is as follows:

mobility.txt

#Initial position of the UE 3

$node_(2) set X_ 1000.0

$node_(2) set Y_ 3000.0

$node_(2) set Z_ 0.0

#Initial position of the UE 4

$node_(3) set X_ 4000.0

$node_(3) set Y_ 3000.0

$node_(3) set Z_ 0.0

#Positions of the UE 3 at specific time

$time 0.0 "$node_(2) 1000.0 3000.0 0.0"

$time 0.5 "$node_(2) 1050.0 3000.0 0.0"

$time 1.0 "$node_(2) 1100.0 3000.0 0.0"

$time 1.5 "$node_(2) 1150.0 3000.0 0.0"

$time 2.0 "$node_(2) 1200.0 3000.0 0.0"

$time 2.5 "$node_(2) 1250.0 3000.0 0.0"

$time 3.0 "$node_(2) 1300.0 3000.0 0.0"

```
$time 3.5 "$node_(2) 1350.0 3000.0 0.0"

$time 4.0 "$node_(2) 1400.0 3000.0 0.0"

$time 4.5 "$node_(2) 1450.0 3000.0 0.0"

$time 5.0 "$node_(2) 1500.0 3000.0 0.0"

$time 5.5 "$node_(2) 1550.0 3000.0 0.0"

$time 6.0 "$node_(2) 1600.0 3000.0 0.0"

$time 6.5 "$node_(2) 1650.0 3000.0 0.0"

$time 7.0 "$node_(2) 1700.0 3000.0 0.0"

$time 7.5 "$node_(2) 1750.0 3000.0 0.0"

$time 8.0 "$node_(2) 1800.0 3000.0 0.0"

$time 8.5 "$node_(2) 1850.0 3000.0 0.0"

$time 9.0 "$node_(2) 1900.0 3000.0 0.0"

$time 9.5 "$node_(2) 1950.0 3000.0 0.0"

$time 10.0 "$node_(2) 2000.0 3000.0 0.0"

$time 10.5 "$node_(2) 2050.0 3000.0 0.0"

$time 11.0 "$node_(2) 2100.0 3000.0 0.0"

$time 11.5 "$node_(2) 2150.0 3000.0 0.0"

$time 12.0 "$node_(2) 2200.0 3000.0 0.0"

$time 12.5 "$node_(2) 2250.0 3000.0 0.0"

$time 13.0 "$node_(2) 2300.0 3000.0 0.0"

$time 13.5 "$node_(2) 2350.0 3000.0 0.0"

$time 14.0 "$node_(2) 2400.0 3000.0 0.0"

$time 14.5 "$node_(2) 2450.0 3000.0 0.0"

$time 15.0 "$node_(2) 2500.0 3000.0 0.0"

$time 15.5 "$node_(2) 2550.0 3000.0 0.0"
```

$time 16.0 "$node_(2) 2600.0 3000.0 0.0"

$time 16.5 "$node_(2) 2650.0 3000.0 0.0"

$time 17.0 "$node_(2) 2700.0 3000.0 0.0"

$time 17.5 "$node_(2) 2750.0 3000.0 0.0"

$time 18.0 "$node_(2) 2800.0 3000.0 0.0"

$time 18.5 "$node_(2) 2850.0 3000.0 0.0"

$time 19.0 "$node_(2) 2900.0 3000.0 0.0"

$time 19.5 "$node_(2) 2950.0 3000.0 0.0"

$time 20.0 "$node_(2) 3000.0 3000.0 0.0"

$time 20.5 "$node_(2) 3050.0 3000.0 0.0"

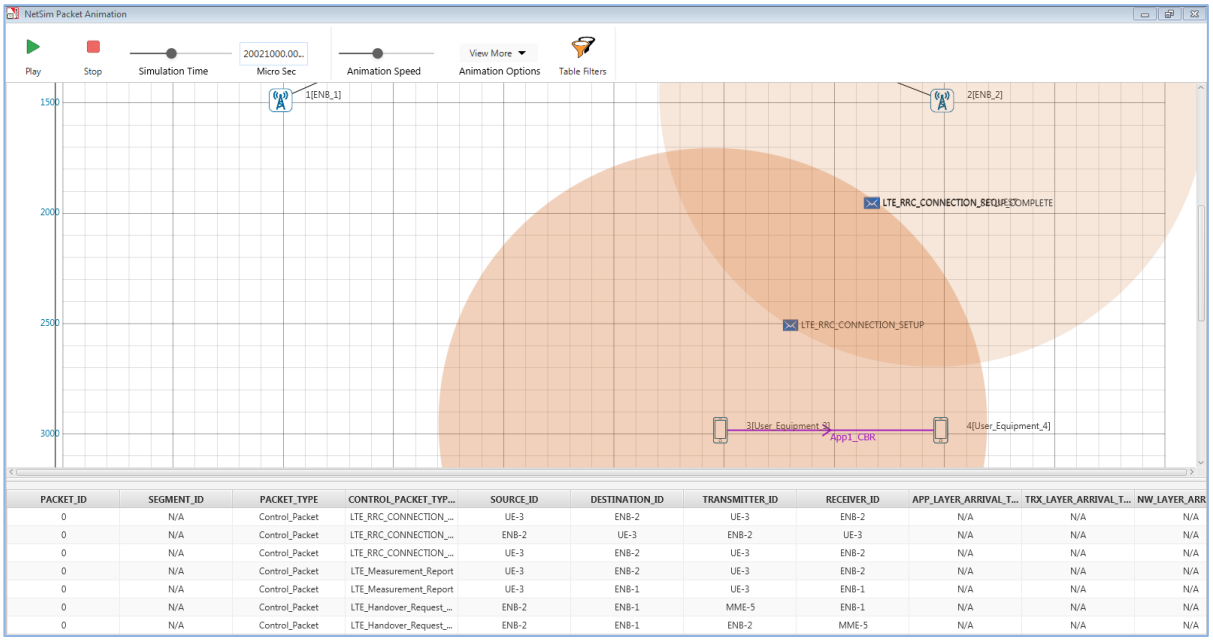
$time 21.0 "$node_(2) 3100.0 3000.0 0.0"

Step 3: Enable Packet trace and run simulation for 50s.

23.5 Measurements and Outputs

Open Packet Animation:

As UE moves from one position to another it sends measurement report to each ENB in range. As it moves SNR received by each ENB keeps on changing based on distance between ENB and UE. If the difference between SNR received by new ENB to that of old ENB to which it is connected is greater at that point handover occurs.



PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYP...	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_T...	TRX_LAYER_ARRIVAL_T...	NW_LAYER_ARR
0	N/A	Control_Packet	LTE_RRC_CONNECTION_...	UE-3	ENB-2	UE-3	ENB-2	N/A	N/A	N/A
0	N/A	Control_Packet	LTE_RRC_CONNECTION_...	ENB-2	UE-3	ENB-2	UE-3	N/A	N/A	N/A
0	N/A	Control_Packet	LTE_RRC_CONNECTION_...	UE-3	ENB-2	UE-3	ENB-2	N/A	N/A	N/A
0	N/A	Control_Packet	LTE_Measurement_Report	UE-3	ENB-2	UE-3	ENB-2	N/A	N/A	N/A
0	N/A	Control_Packet	LTE_Measurement_Report	UE-3	ENB-1	UE-3	ENB-1	N/A	N/A	N/A
0	N/A	Control_Packet	LTE_Handover_Request_...	ENB-2	ENB-1	MME-5	ENB-1	N/A	N/A	N/A
0	N/A	Control_Packet	LTE_Handover_Request_...	ENB-2	ENB-1	ENB-2	MME-5	N/A	N/A	N/A

23.6 Inference

1. As shown in the above packet animation table, UE 3 connected to eNB 1 and UE 4, connected to eNB 2
2. UE 3 is moving from eNB 1 to eNB 2 due to mobility
3. Then UE 3 sends the LTE_Measurement_Report to eNB 1
4. The eNB 1 sends a LTE_Handover_Request message to the eNB B, if the received SNR by eNB 2 is greater than eNB 1, by 3dB (margin of handover)
5. eNB 2 checks for resource availability and sends a LTE_Handover_Request_Ack message to the eNB 1
6. Now UE 3 starts communicating with eNB 2 shown in the above screenshot

23.7 Additional Notes & References

1. To calculate and print SNR for each pair of eNB-UE combination please refer NetSim knowledgebase article (https://tetcos.freshdesk.com/solution/articles/14000037296-how-can-i-print-snr-cqi-mcs-index-and-tbs-index-value-to-a-file-)
2. If the wireless links have no path loss set, then there will never be any handovers because the received power from all eNB's will be the same

24. Understand the working of LTE Device to Device Communication

24.1 Theory:

LTE D2D communication is a peer to peer link which does not use the cellular network infrastructure, but enables LTE based devices to communicate directly with one another when they are in close proximity.

D2D would enable the direct link of a device user equipment UE to another device using the cellular spectrum. This could allow large volumes of media or other data to be transferred from one device to another over short distances and using a direct connection. This form of device to device transfer would enable the data to be transferred without the need to run it via the cellular network itself, thereby avoiding problems with overloading the network.

The D2D model can be summarized as follows:

- Each UE produces its D2D identity and transmits it to the eNB during its first access to the network.
- UE's make D2D spectrum requests including the D2D identity of the target D2D receiver.
- eNB launches a peer discovery procedure for the requested D2D pair.
- eNB allocates cellular resources to valid D2D pairs and informs both D2D peers, tuning them indirectly at the same spectrum portion.
- The UE transmitter sends its data using the spectrum region that has been allocated by the eNB, while the UE receiver tunes to the same spectrum region to receive the transmitted data.
- The UE receiver acknowledges the reception of the data through the eNB.

24.2 Benefits of D2D communications

Direct communications between devices can provide several benefits to users in various applications where the devices are in close proximity:

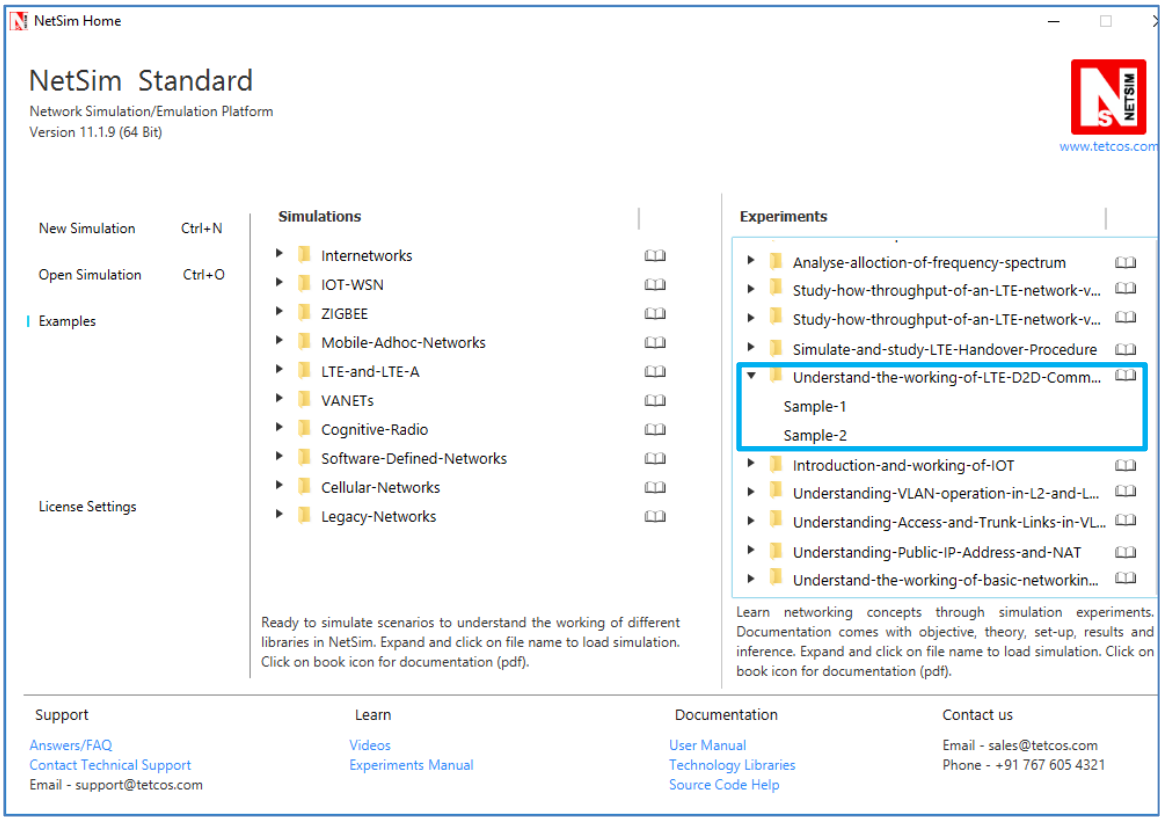
- **Reliable communications:** LTE Device to Device can be used to communicate locally between devices to provide highly reliable communications especially if the LTE network has failed for any reason - even as a result of the disaster.
- **Instant communications:** As the D2D communications does not rely on the network infrastructure the devices could be used for instant communications between a set

numbers of devices in the same way that walkie-talkies are used. This is particularly applicable to the way communications may be used by the emergency services.

- **Interference reduction:** By not having to communicate directly with a base station, fewer links are required (i.e. essentially only between devices) and this has an impact of the amount of data being transmitted within a given spectrum allocation. This reduces the overall level of interference.
- **Power saving:** Using device to device communication provides energy saving, if the two devices are in close proximity then lower transmission power levels are required

24.3 Procedure:

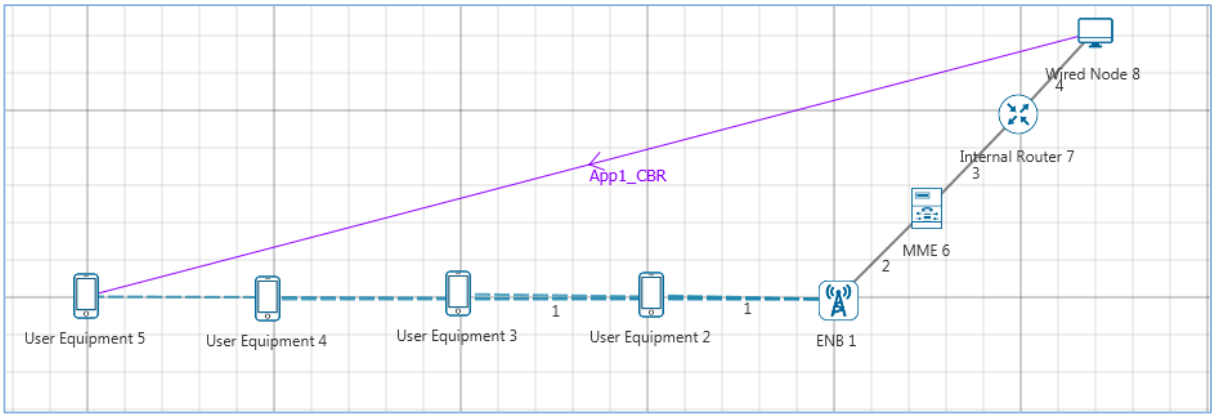
Step 1: Open Examples → Understand-the-working-of-LTE-D2D-Communication as shown below:



Step 2: Click on Environment Settings and change Grid length to 5000 meters

SAMPLE1:

Click & drop 1 eNB, 1 MME, 4 UE's, 1 Router and 1 Wired Node onto the Simulation Environment and connect as per the following.



Set the coordinates of wireless devices as per the following

Device Type	X - Coordinate	Y - Coordinate
eNB 1	4500	1000
UE 2	3500	1000
UE 3	2500	1000
UE 4	1500	1000
UE 5	500	1000

Step 3: Properties

UE Properties:

UE	Transport Layer Properties	Interface_LTE Properties
	TCP	**D2D**
UE 2	disable	FALSE
UE 3	disable	FALSE
UE 4	disable	FALSE
UE 5	disable	FALSE

Wireless Link Properties:

Wireless Link Properties	
Channel Characteristics	Path loss only
Path loss Model	Log Distance
Path loss exponent	3.2

Application Properties:

Application Type	Source ID	Destination ID
CBR	8	5

Set Default properties for all other parameters / devices and enable Packet Trace

Step 4: Set all the properties explained above and then click on Run Simulation

Set Simulation time = 10 seconds

After completion, users can save this experiment

SAMPLE 2

Open the saved scenario and set the properties given below

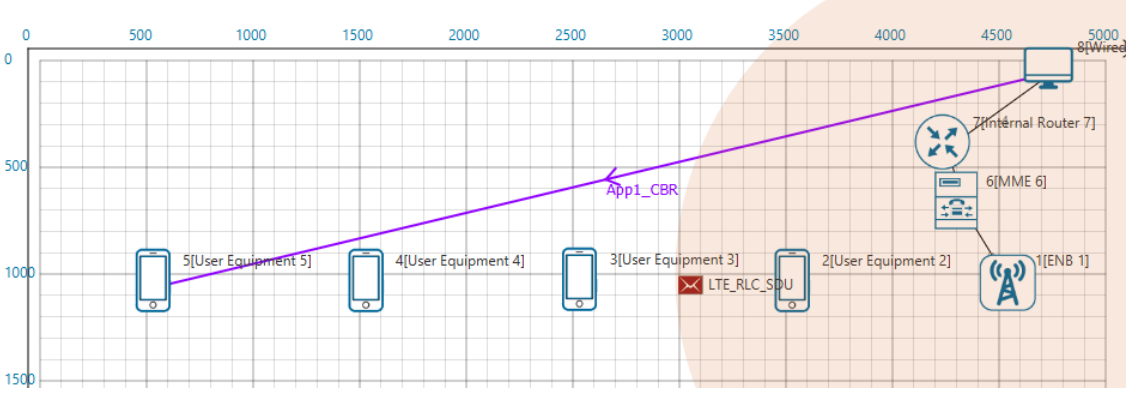
UE Properties:

UE	Transport Layer Properties	Interface_LTE Properties
	TCP	**D2D**
UE 2	disable	TRUE
UE 3	disable	TRUE
UE 4	disable	TRUE
UE 5	disable	TRUE

Enable Packet Trace and run simulation for 10 s.

24.4 Output

Sample 1: Without D2D

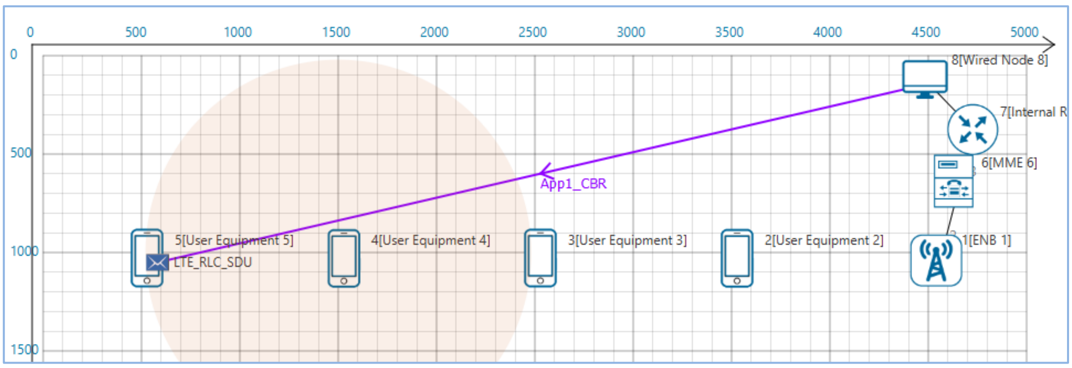


As shown in above figure, application is set from Wired Node 8 to UE 5. As UE 5 is far away from eNB 1, there is too much of attenuation and packets from eNB 1 to UE 5 get errored. We can observe this in Packet Animation. This results in a very low or zero throughput. Users can also observe in the packet animation that only LTE_RLC_SDUs are errored (in red color). The same can also be seen from the Packet trace by filtering CONTROL_PACKET_TYPE to LTE_RLC_SDU packets. For doing this refer section 7.5 in NetSim's user manual.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACK	SOURCE	DESTI	TRANSMITTER	RECEIVER	PACKET_STATUS
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-5	ENB-1	UE-5	Errored

Sample 2: With D2D

In second case, even though UE E is far away from eNB 1, packets will reach to UE 5 via intermediate UEs (in this case UE 4). Users can observe this in Animation and Packet Trace. As shown in the figure below, eNB 1 is first transmitting the LTE_RLC_SDU packets to UE 4 and then UE 4 is transmitting to UE 5 using LTE Device to device communication. In this case, we get considerably higher throughput since the errored packets are less.



Users can also observe this in Packet trace by filtering CONTROL_PACKET_TYPE to LTE_RLC_SDU packets.

PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_P	SOURCE_	DEST	TRANSM	RECEIVER_I	APP_LAYE	PACKET_STATUS
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-4	ENB-1	UE-4	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-4	ENB-1	UE-4	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	UE-4	UE-5	UE-4	UE-5	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-4	ENB-1	UE-4	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-4	ENB-1	UE-4	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	UE-4	UE-5	UE-4	UE-5	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-4	ENB-1	UE-4	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-4	ENB-1	UE-4	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	UE-4	UE-5	UE-4	UE-5	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-4	ENB-1	UE-4	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-4	ENB-1	UE-4	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	UE-4	UE-5	UE-4	UE-5	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-4	ENB-1	UE-4	N/A	Successful
0	N/A	Control_Packet	LTE_RLC_SDU	ENB-1	UE-4	ENB-1	UE-4	N/A	Successful

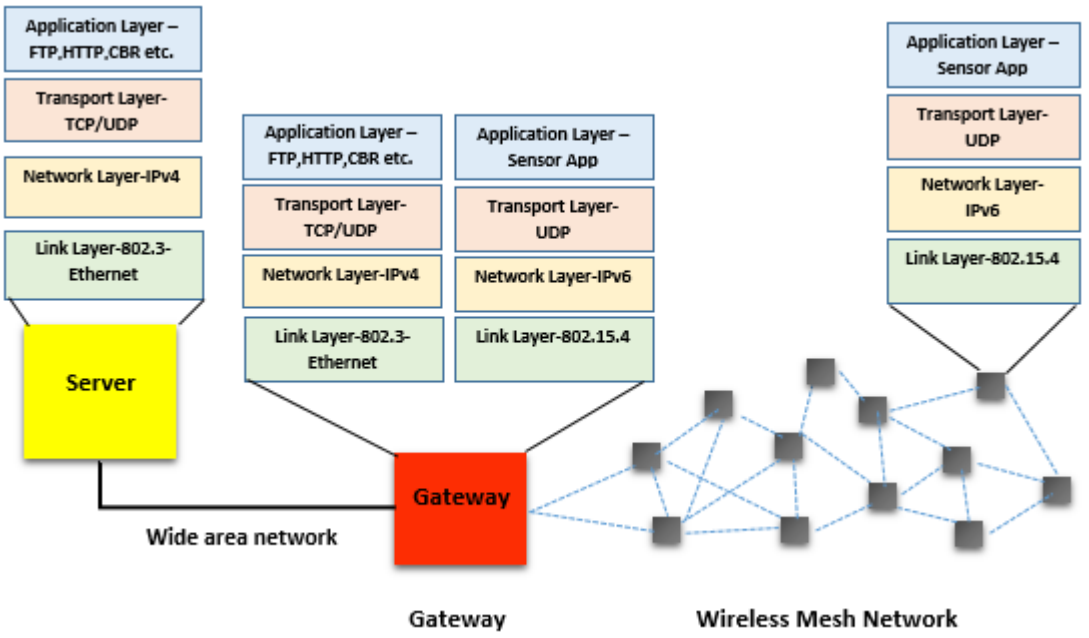
From the above figure, users can observe that eNB-1 is transmitting LTE_RLC_SDU packet to UE-4 and then UE-4 is transmitting to UE-5.

25. Introduction and working of Internet of Things (IoT)

25.1 Introduction

Internet of Things (**IoT**) is a network of physical devices, vehicles, buildings and other items embedded with electronics, software, sensors, actuators, and network connectivity that enable these objects to collect and exchange data. An IoT network allows objects to be sensed and/or controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit.

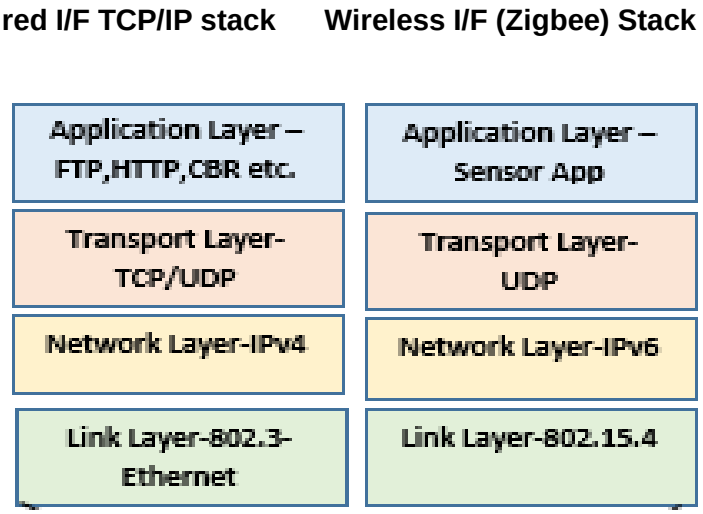
25.1.1 Simple IOT scenario



25.1.2 Components

1. **Sensors:** Sensors are used to detect physical phenomena such as light, heat, pressure, temperature, humidity etc. Sensors are regarded as a revolutionary information gathering method to build the information and communication system which will greatly improve the reliability and efficiency of infrastructure systems. It follows IPv6 addressing system. IP addresses are the backbone to the entire IoT ecosystem. IPv6's huge increase in address space is an important factor in the development of the Internet of Things.
2. **LowPAN Gateway:** These are the Gateways to Internet for all the things/devices that we want to interact with. Gateway help to bridge the internal network of sensor nodes with the external Internet i.e., it will collect the data from sensors and transmitting it to the internet infrastructure.

A 6LowPAN Gateway will have 2 interfaces, one is Zigbee interface connected to sensors (follows 802.15.4 MAC and PHY) and the other is WAN interface connected to ROUTER.

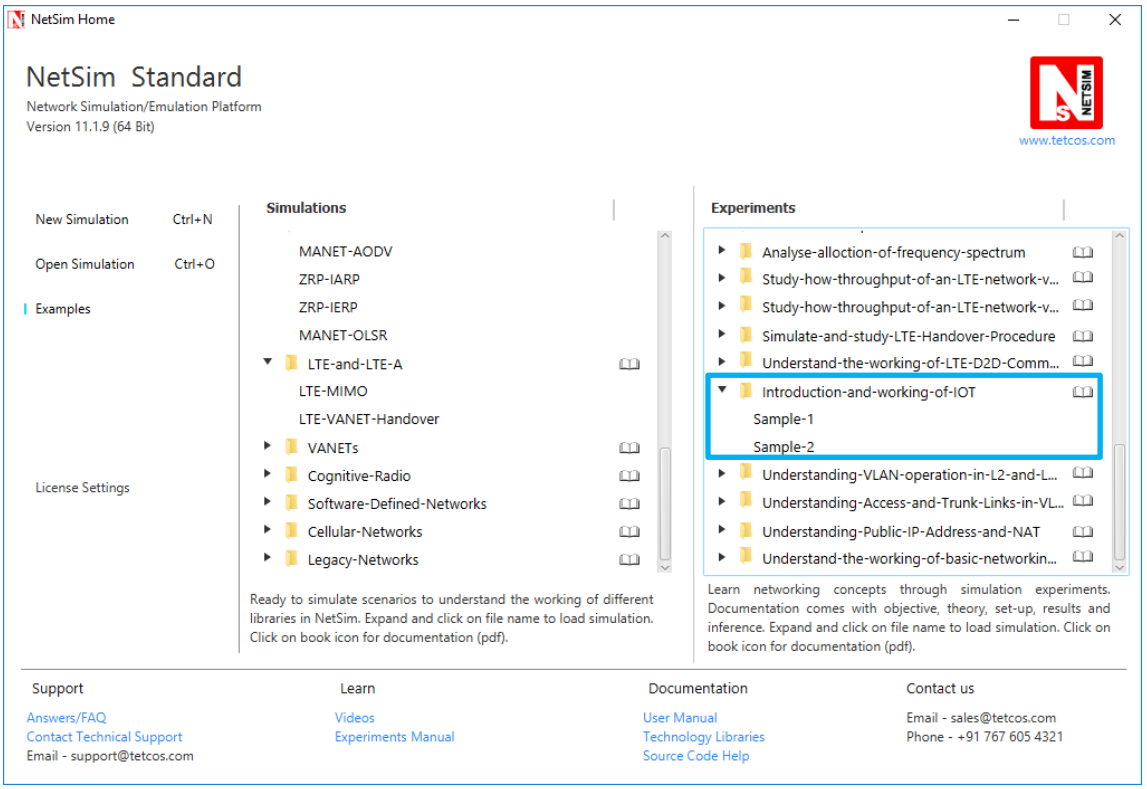


6LowPAN Gateway Stack at wired and wireless Interfaces

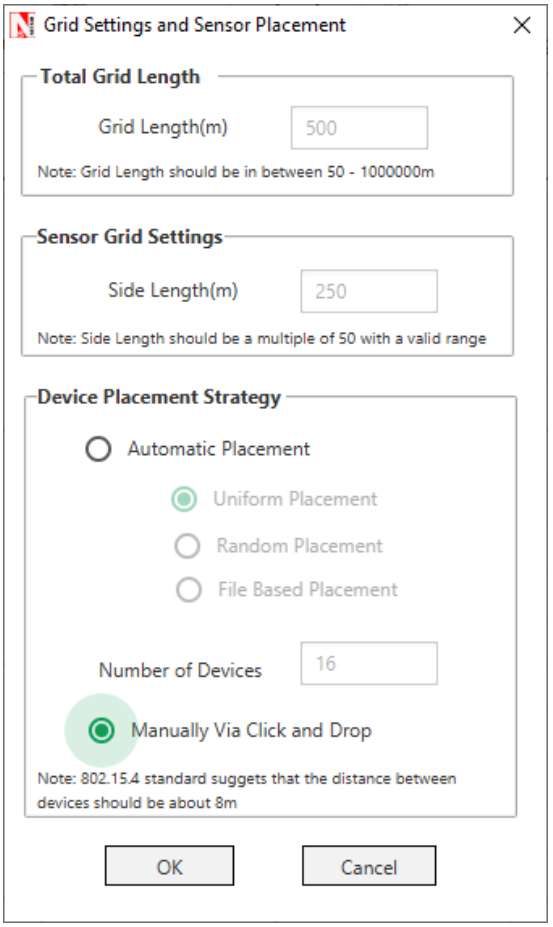
6LoWPAN is an acronym of IPv6 over Low Power Wireless Personal Area Network. The 6LoWPAN concept originated from the idea that "the Internet Protocol should be applied even to the smallest devices, and that low-power devices with limited processing capabilities should be able to participate in the Internet of Things.

25.2 PART A: To Model and Simulate an IoT Network Scenario in NetSim

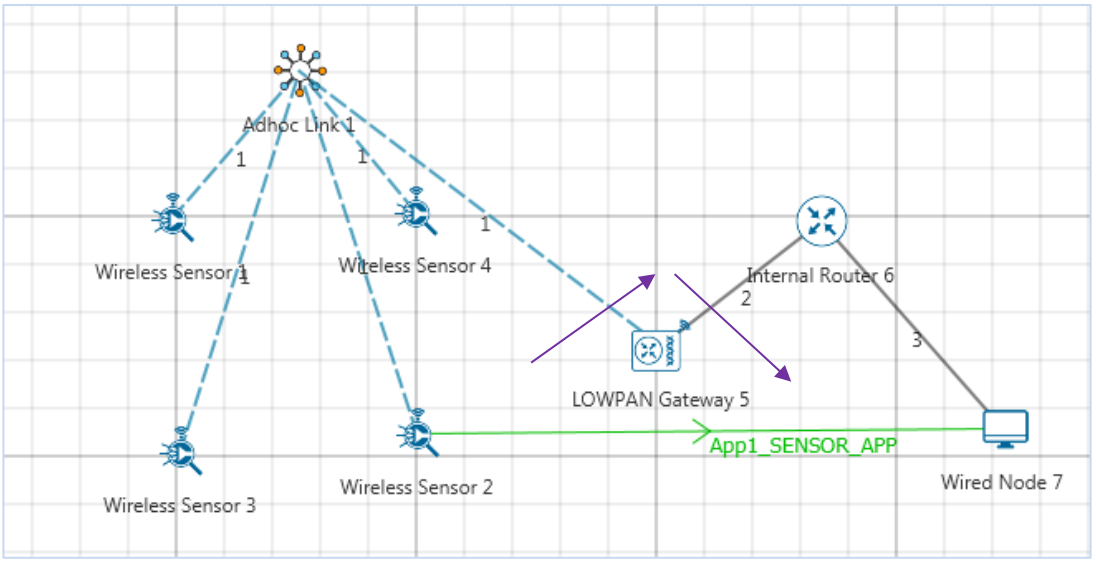
Step 1: Open Examples → Introduction-and-working-of-IOT as shown below:



Step 2: NetSim provides a fast configuration option for creating networks in IOT/WSN. Select Manually via Click and Drop and Click on OK.



Step 3: Click and drop 4 Sensors, 1 LowPAN Gateway, 1 Router and 1 Wired Node. Connect Router, LowPAN Gateway and Wired Node using Wired Links



Step 4: Similarly click and drop **Adhoc** link onto the grid. Then click on Adhoc link icon (in grid) and on sensors (in grid) to connect sensors to the Adhoc link. Then click on Adhoc link icon (in grid) and on Sink node (in grid) to connect Adhoc link to Sink Node. Shown above is a screen shot what a finally connected network would look like.

Step 5: Click on the Application icon present on the ribbon and set properties as given in table. All other properties are default.

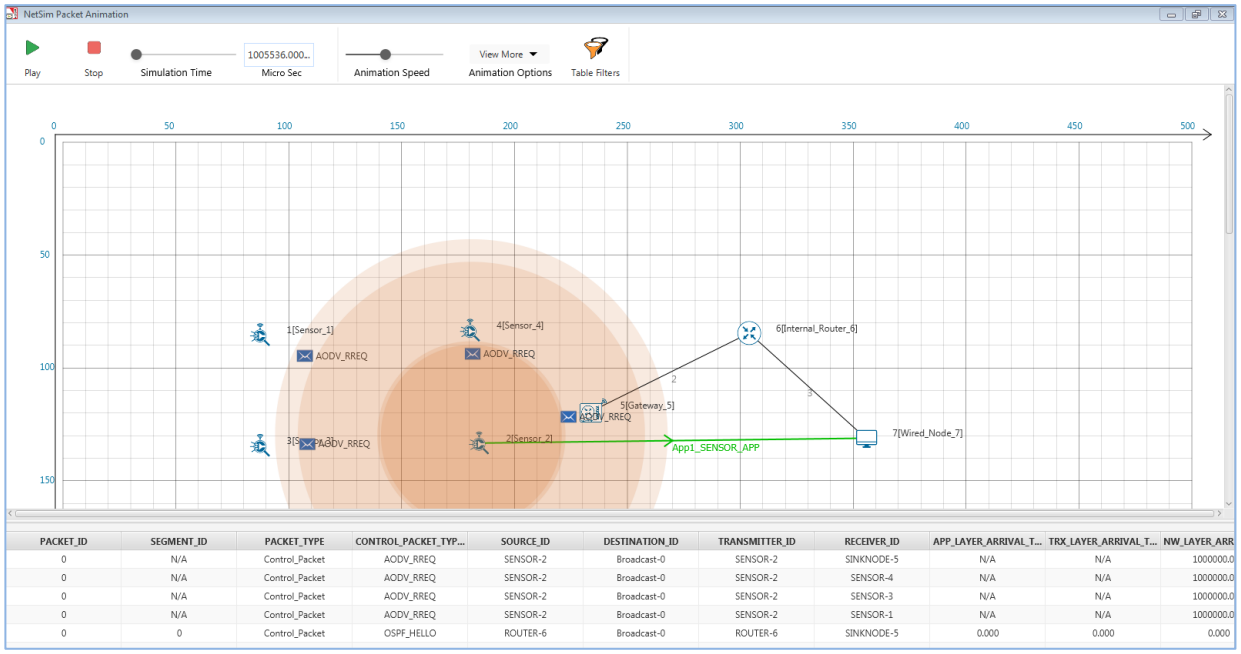
Application Type	Sensor_App
Source ID	2
Destination ID	7

Step 4: Click on Run Simulation icon. Set the Simulation Time as 100 seconds and click on OK.

25.3 Output

At the end of simulation, NetSim provides various performance metrics such as Network Metrics, Link Metrics, Application Metrics, and Protocol Metrics etc.

In Packet Animation Table, users can see the packet flow.

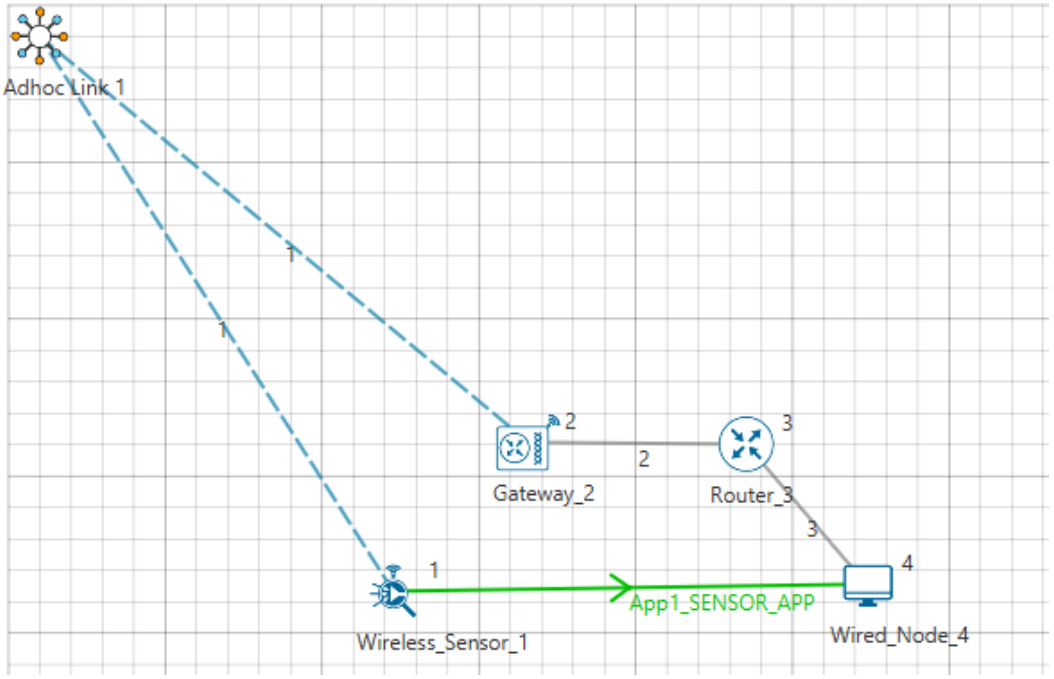


PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYP...	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID	APP_LAYER_ARRIVAL_T...	TRX_LAYER_ARRIVAL_T...	NW_LAYER_ARR
0	N/A	Control_Packet	AODV_RREQ	SENSOR-2	Broadcast-0	SENSOR-2	SINKNODE-5	N/A	N/A	1000000.0
0	N/A	Control_Packet	AODV_RREQ	SENSOR-2	Broadcast-0	SENSOR-2	SENSOR-4	N/A	N/A	1000000.0
0	N/A	Control_Packet	AODV_RREQ	SENSOR-2	Broadcast-0	SENSOR-2	SENSOR-3	N/A	N/A	1000000.0
0	N/A	Control_Packet	AODV_RREQ	SENSOR-2	Broadcast-0	SENSOR-2	SENSOR-1	N/A	N/A	1000000.0
0	0	Control_Packet	OSPF_HELLO	ROUTER-6	Broadcast-0	ROUTER-6	SINKNODE-5	0.000	0.000	0.000

25.4 PART B: Understand the flow of packet from IPv6 to IPv4 network using Packet Trace

Follow the steps till step 2 from the above.

Step 3: Click and drop 1 sensor, 1 LowPAN Gateway, 1 Router and 1 Wired Node. Connect Router, LowPAN Gateway and Wired Node using Wired Links.



Step 4: Similarly click and drop **Adhoc** link onto the grid. Then click on Adhoc link icon (in grid) and on sensor (in grid) to connect sensor to the Adhoc link. Then click on Adhoc link icon (in grid) and on Sink node (in grid) to connect Adhoc link to Sink Node. Shown above is a screen shot what a finally connected network would look like.

Step 5: Click on the Application icon present on the ribbon and set properties as given in table. All other properties are default.

Application Type	Sensor_App
Source ID	1
Destination ID	4

Step 6: Click Packet Trace icon in the tool bar and check Enable Packet Trace check box and click OK.

Step 7: Click on Run Simulation icon. Set the Simulation Time as 10 seconds and click on OK.

25.5 Output

Users can understand how the IP addresses are changing from IPv6 to IPv4 and vice versa with the help of packet trace file.

After simulation, open packet trace and filter PACKET_TYPE to Sensing and observe the columns SOURCE_IP, DESTINATION_IP, GATEWAY_IP and NEXT_HOP_IP

SOURCE_IP – source node IP

DESTINATION_IP – gateway IP

GATEWAY_IP – IP of the device which is transmitting a packet

NEXT_HOP_IP – IP of the next hop

1. Sensor and 6_LOWPAN_Gateways 1st interface follows IPv6 addressing.
2. 6_LOWPAN_Gateways 2nd interface, Router and Wired Node follows IPv4 addressing.
3. From the screenshot below, users can identify the changing of IP addresses from source to destination.

	PACKET_T	CONTROL_PACKET	SOURCE_II	DESTINATIC	TRANSMITT	RECEIVE	OURCE_IP	DESTINATION_IP	GATEWAY_IP	NEXT_HOP_IP
7	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SENSOR-1	SINKNODE-	DEC:3017:E256:9BB8:1FE7:	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5	FDEC:3017:E256:9BB8:1FE7:9481:58A1:D94B	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5
8	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SINKNODE-2	ROUTER-3	DEC:3017:E256:9BB8:1FE7:	11.2.1.2	11.2.1.1	11.2.1.2
9	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	ROUTER-3	NODE-4	DEC:3017:E256:9BB8:1FE7:	11.3.1.2	11.3.1.1	11.3.1.2
11	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SENSOR-1	SINKNODE-	DEC:3017:E256:9BB8:1FE7:	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5	FDEC:3017:E256:9BB8:1FE7:9481:58A1:D94B	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5
12	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SINKNODE-2	ROUTER-3	DEC:3017:E256:9BB8:1FE7:	11.2.1.2	11.2.1.1	11.2.1.2
13	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	ROUTER-3	NODE-4	DEC:3017:E256:9BB8:1FE7:	11.3.1.2	11.3.1.1	11.3.1.2
17	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SENSOR-1	SINKNODE-	DEC:3017:E256:9BB8:1FE7:	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5	FDEC:3017:E256:9BB8:1FE7:9481:58A1:D94B	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5
18	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SINKNODE-2	ROUTER-3	DEC:3017:E256:9BB8:1FE7:	11.2.1.2	11.2.1.1	11.2.1.2
19	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	ROUTER-3	NODE-4	DEC:3017:E256:9BB8:1FE7:	11.3.1.2	11.3.1.1	11.3.1.2
23	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SENSOR-1	SINKNODE-	DEC:3017:E256:9BB8:1FE7:	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5	FDEC:3017:E256:9BB8:1FE7:9481:58A1:D94B	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5
24	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SINKNODE-2	ROUTER-3	DEC:3017:E256:9BB8:1FE7:	11.2.1.2	11.2.1.1	11.2.1.2
25	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	ROUTER-3	NODE-4	DEC:3017:E256:9BB8:1FE7:	11.3.1.2	11.3.1.1	11.3.1.2
29	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SENSOR-1	SINKNODE-	DEC:3017:E256:9BB8:1FE7:	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5	FDEC:3017:E256:9BB8:1FE7:9481:58A1:D94B	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5
30	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SINKNODE-2	ROUTER-3	DEC:3017:E256:9BB8:1FE7:	11.2.1.2	11.2.1.1	11.2.1.2
31	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	ROUTER-3	NODE-4	DEC:3017:E256:9BB8:1FE7:	11.3.1.2	11.3.1.1	11.3.1.2
33	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SENSOR-1	SINKNODE-	DEC:3017:E256:9BB8:1FE7:	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5	FDEC:3017:E256:9BB8:1FE7:9481:58A1:D94B	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5
34	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SINKNODE-2	ROUTER-3	DEC:3017:E256:9BB8:1FE7:	11.2.1.2	11.2.1.1	11.2.1.2
35	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	ROUTER-3	NODE-4	DEC:3017:E256:9BB8:1FE7:	11.3.1.2	11.3.1.1	11.3.1.2
39	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SENSOR-1	SINKNODE-	DEC:3017:E256:9BB8:1FE7:	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5	FDEC:3017:E256:9BB8:1FE7:9481:58A1:D94B	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5
40	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SINKNODE-2	ROUTER-3	DEC:3017:E256:9BB8:1FE7:	11.2.1.2	11.2.1.1	11.2.1.2
41	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	ROUTER-3	NODE-4	DEC:3017:E256:9BB8:1FE7:	11.3.1.2	11.3.1.1	11.3.1.2
45	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SENSOR-1	SINKNODE-	DEC:3017:E256:9BB8:1FE7:	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5	FDEC:3017:E256:9BB8:1FE7:9481:58A1:D94B	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5
46	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SINKNODE-2	ROUTER-3	DEC:3017:E256:9BB8:1FE7:	11.2.1.2	11.2.1.1	11.2.1.2
47	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	ROUTER-3	NODE-4	DEC:3017:E256:9BB8:1FE7:	11.3.1.2	11.3.1.1	11.3.1.2
51	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SENSOR-1	SINKNODE-	DEC:3017:E256:9BB8:1FE7:	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5	FDEC:3017:E256:9BB8:1FE7:9481:58A1:D94B	FDEC:3017:E256:9BB8:1FE7:A31F:17B1:87F5
52	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	SINKNODE-2	ROUTER-3	DEC:3017:E256:9BB8:1FE7:	11.2.1.2	11.2.1.1	11.2.1.2
53	Sensing	App1_SENSOR_APP	SENSOR-1	NODE-4	ROUTER-3	NODE-4	DEC:3017:E256:9BB8:1FE7:	11.3.1.2	11.3.1.1	11.3.1.2

26. Understand the working of TCP BIC Congestion control algorithm, simulate and plot the TCP congestion window

26.1 Theory

In BIC congestion control is viewed as a searching problem in which the system can give yes/no feedback through packet loss as to whether the current sending rate (or window) is larger than the network capacity. The current minimum window can be estimated as the window size at which the flow does not see any packet loss. If the maximum window size is known, we can apply a binary search technique to set the target window size to the midpoint of the maximum and minimum. As increasing to the target, if it gives any packet loss, the current window can be treated as a new maximum and the reduced window size after the packet loss can be the new minimum. The midpoint between these new values becomes a new target. Since the network incurs loss around the new maximum but did not do so around the new minimum, the target window size must be in the middle of the two values. After reaching the target and if it gives no packet loss, then the current window size becomes a new minimum, and a new target is calculated. This process is repeated with the updated minimum and maximum until the difference between the maximum and the minimum falls below a preset threshold, called the minimum increment (S_{min}). This technique is called binary search increase.

Additive Increase

In order to ensure faster convergence and RTT-fairness, binary search increase is combined with an additive increase strategy. When the distance to the midpoint from the current minimum is too large, increasing the window size directly to that midpoint might add too much stress to the network. When the distance from the current window size to the target in binary search increase is larger than a prescribed maximum step, called the maximum increment (Smax) instead of increasing window directly to that midpoint in the next RTT, we increase it by Smax until the distance becomes less than Smax, at which time window increases directly to the target. Thus, after a large window reduction, the strategy initially increases the window linearly, and then increases logarithmically. This combination of binary search increase and additive increase is called as binary increase. Combined with a multiplicative decrease strategy, binary increase becomes close to pure additive increase under large windows. This is because a larger window results in a larger reduction by multiplicative decrease and therefore, a longer additive increase period. When the window size is small, it becomes close to pure binary search increase – a shorter additive increase period.

Slow Start:

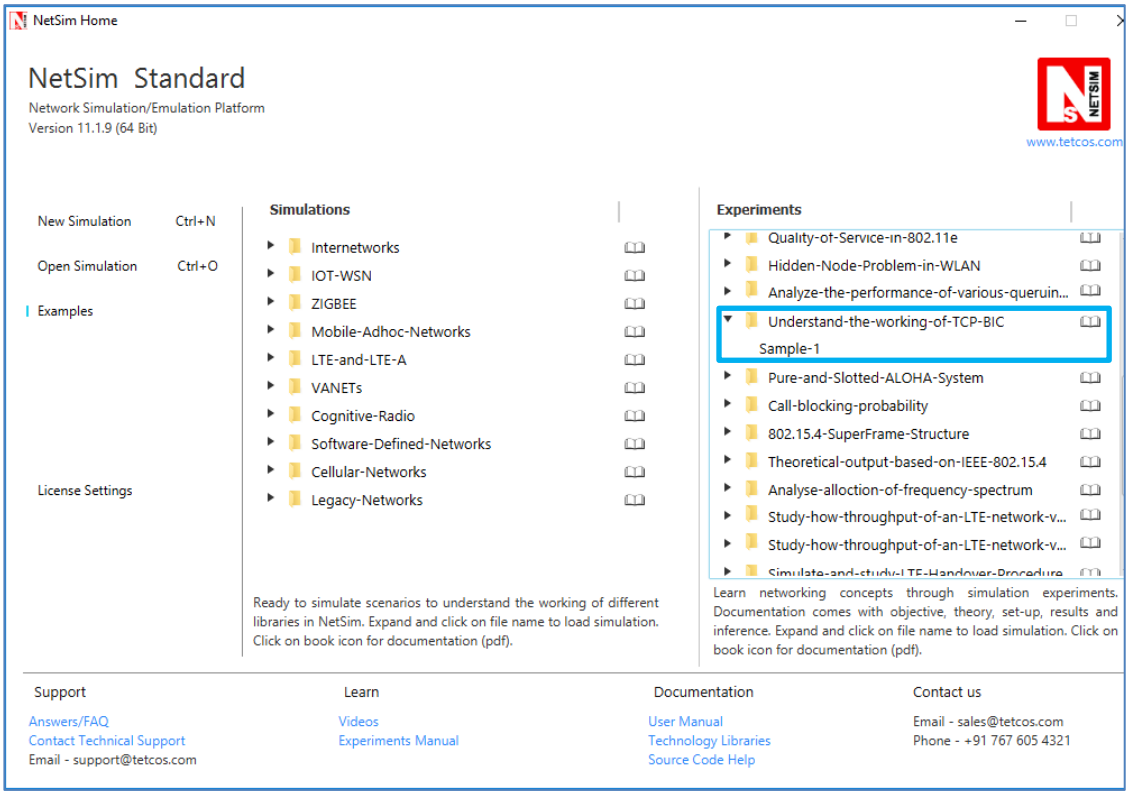
After the window grows past the current maximum, the maximum is unknown. At this time, binary search sets its maximum to be a default maximum (a large constant) and the current window size to be the minimum. So the target midpoint can be very far. According to binary increase, if the target midpoint is very large, it increases linearly by the maximum increment. Instead, run a “slow start” strategy to probe for a new maximum up to Smax. So if cwnd is the current window and the maximum increment is Smax, then it increases in each RTT round in steps cwnd+1, cwnd+2, cwnd+4,…, cwnd+Smax. The rationale is that since it is likely to be at the saturation point and also the maximum is unknown, it probes for available bandwidth in a “slow start” until it is safe to increase the window by Smax. After slow start, it switches to binary increase.

Fast Convergence:

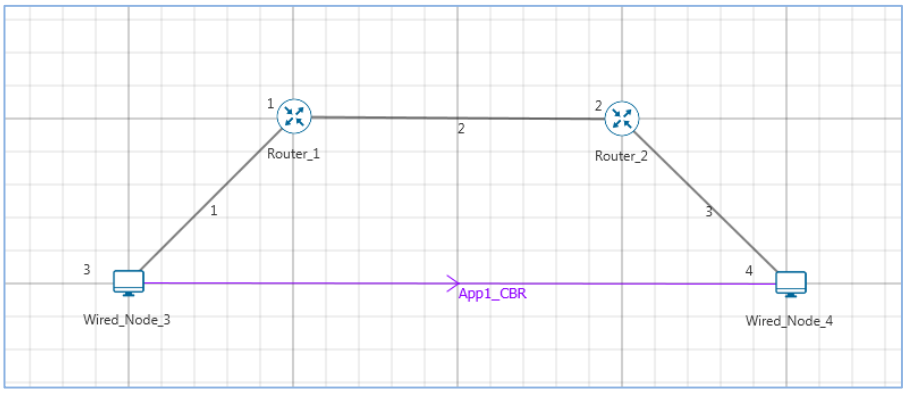
It can be shown that under a completely synchronized loss model, binary search increase combined with multiplicative decrease converges to a fair share. Suppose there are two flows with different window sizes, but with the same RTT. Since the larger window reduces more in multiplicative decrease (with a fixed factor β), the time to reach the target is longer for a larger window. However, its convergence time can be very long. In binary search increase, it takes $log(d) - log(Smin)$ RTT rounds to reach the maximum window after a window reduction of d. Since the window increases in a log step, the larger window and smaller window can reach back to their respective maxima very fast almost at the same time (although the smaller window flow gets to its maximum slightly faster). Thus, the smaller window flow ends up taking away only a small amount of bandwidth from the larger flow before the next window reduction. To remedy this behaviour, binary search increase is modified as follows. After a window reduction, new maximum and minimum are set. Suppose these values are max_wini and min_wini for flow i (i =1, 2). If the new maximum is less than the previous, this window is in a downward trend. Then, readjust the new maximum to be the same as the new target window (i.e. max_wini = (max_wini-min_wini)/2), and then readjust the target. After that apply the normal binary increase. This strategy is called fast convergence.

26.2 Network setup

Open Examples → Understand-the-working-of-TCP-BIC as shown below:



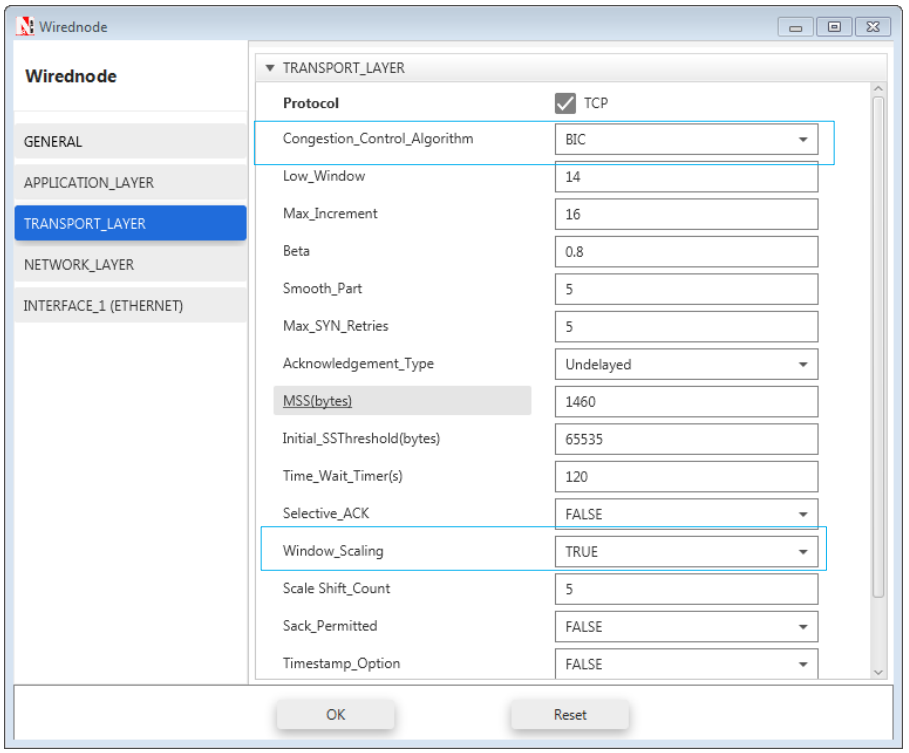
Create a network scenario in Internetworks with 2 routers and 2 wired nodes as shown below:



Set the Properties as mentioned below:

Enable Wireshark Capture in Wired Node 3

Enable BIC in all devices and set Window Scaling as TRUE in Wired Node3



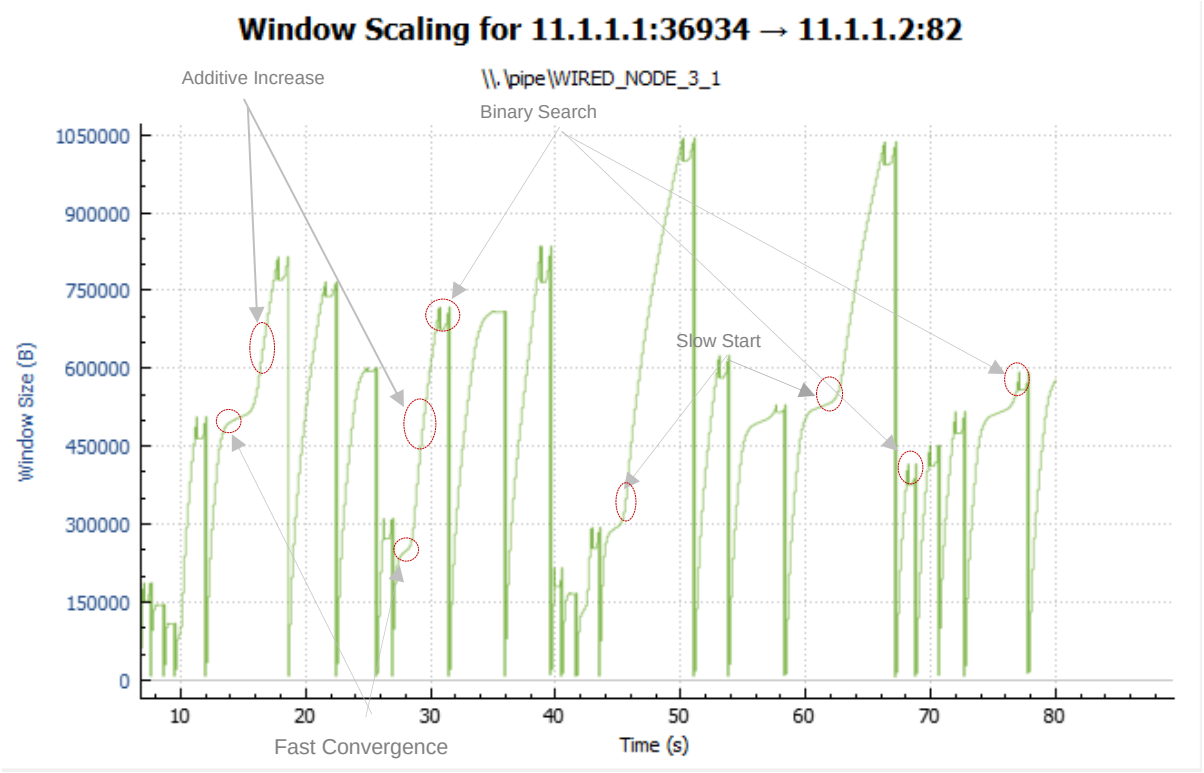
Application Properties	
Application Type	CBR
Source ID	3
Destination ID	4
Packet Size (Bytes)	1460
Start Time	20
Inter Arrival Time (µs)	400

Link Properties	Wired Link 1	Wired Link 2	Wired Link 3
Uplink Speed (Mbps)	20	100	20
Downlink Speed (Mbps)	20	100	20
Uplink propagation delay	5	1000	5
Downlink propagation delay	5	1000	5
Bit error rate (all links)	10 e(-8)	10 e(-8)	10 e(-8)

After setting all properties, Run Simulation for 100s.

26.3 Output

Wireshark Window Scaling Plot



Click on data packet i.e. <None>. Go to Statistics→ TCP Stream Graphs→Window Scaling.

Click on Switch Direction in the window scaling graph window to view the graph.

(For more guidance refer “section - 7.7.5 Window Scaling” in user manual)

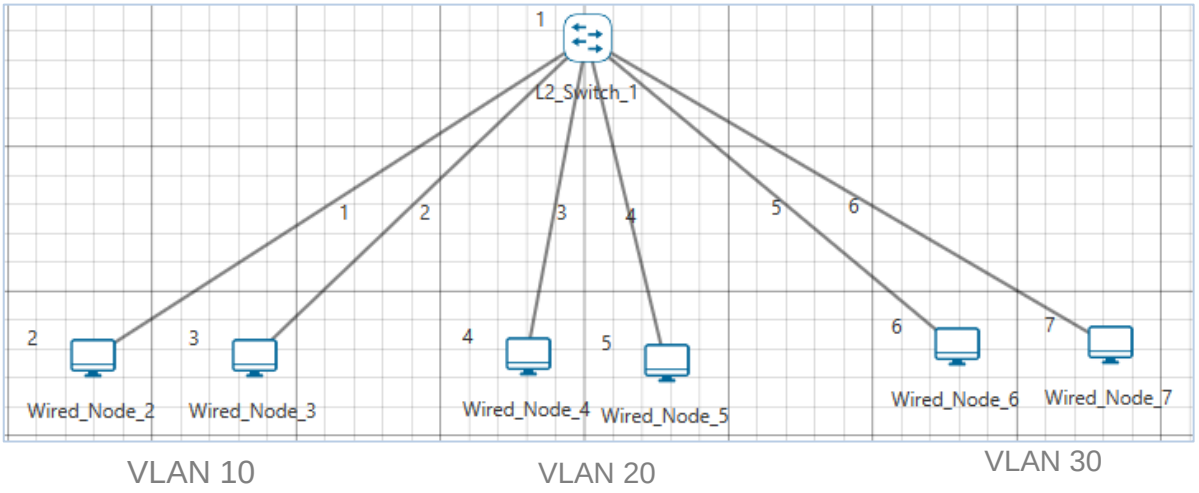
The graph shown above is a plot of Congestion Window vs Time of BIC for the scenario shown above. Each point on the graph represents the congestion window at the time when the packet is sent. You can observe Binary Search, Additive Increase, Fast Convergence, Slow Start phases in the above graph.

27. Understanding VLAN operation in L2 and L3 Switches

27.1 Introduction to VLAN

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

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

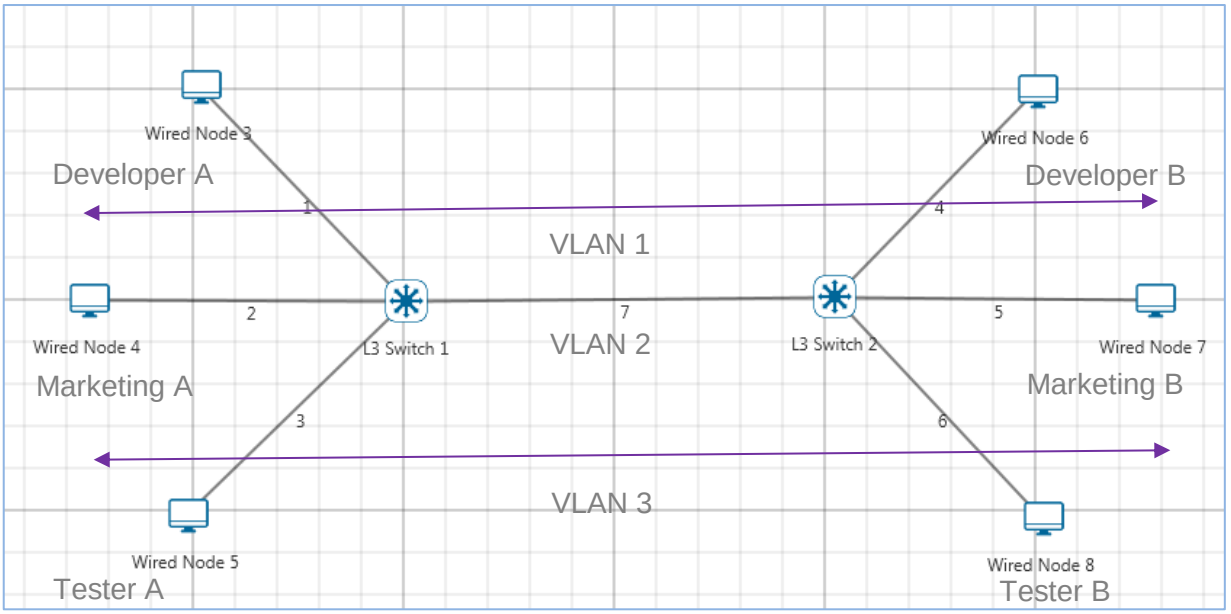


A VLAN behaves just like a LAN in all respects but with additional flexibility. By using VLAN technology, it is possible to subdivide a single physical switch into several logical switches. VLANs are implemented by using the appropriate switch configuration commands to create the VLANs and assign specific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

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

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

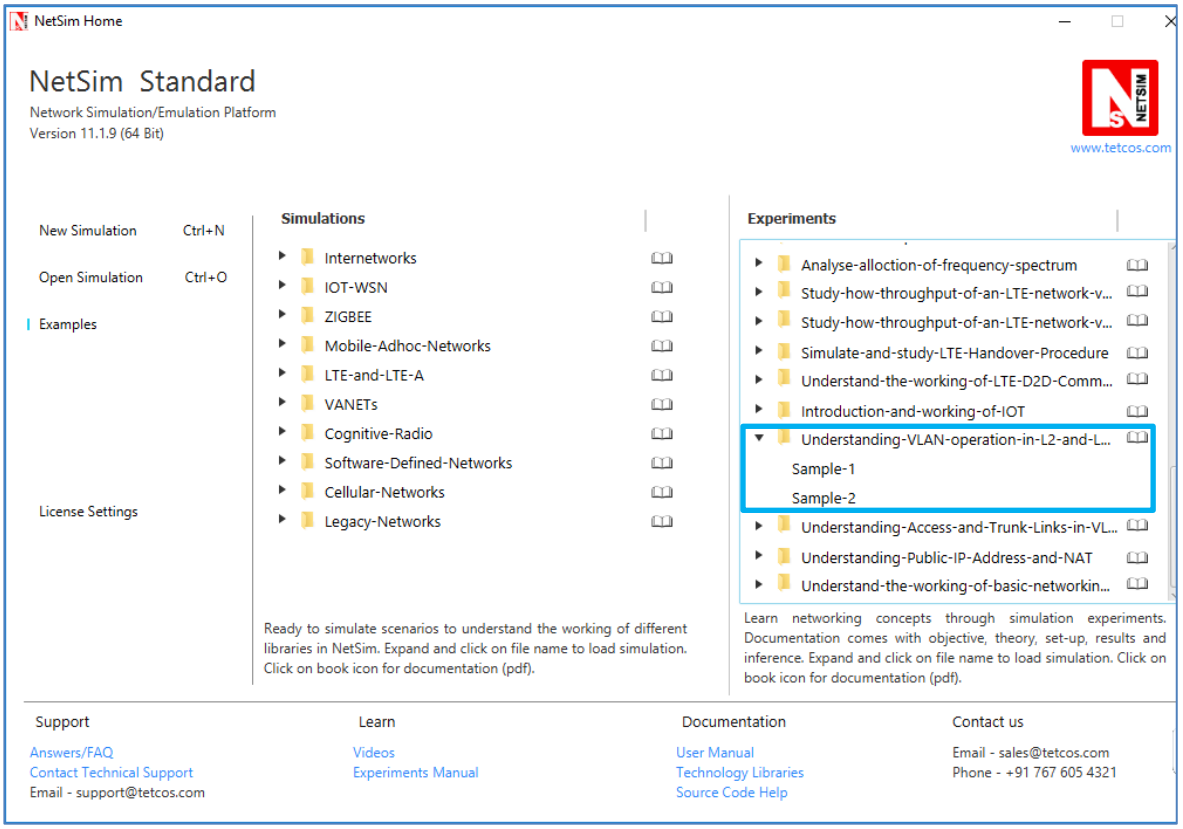
In the below screenshot, the stations in the development department are assigned to one VLAN, the stations in the marketing department are assigned to another VLAN, and the stations in the testing department are assigned to another VLAN.



27.2 Sample 1- Intra-VLAN

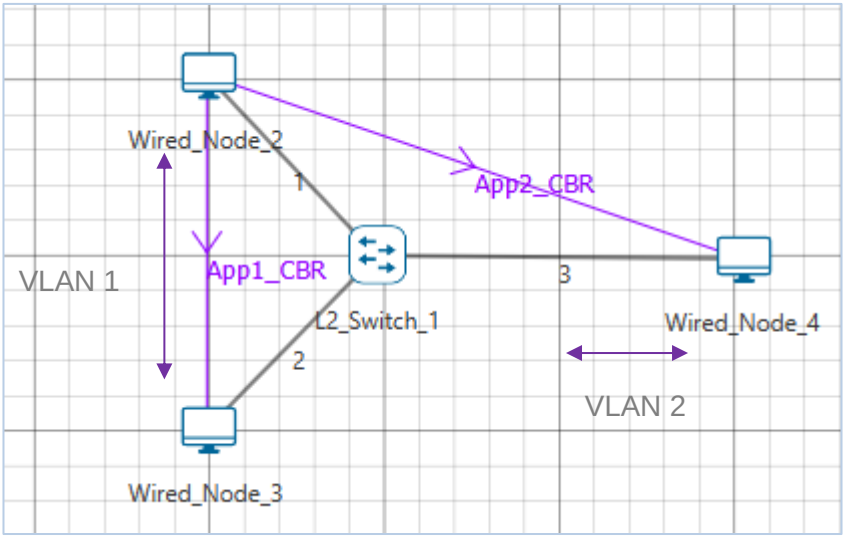
Intra-VLAN is a mechanism in which hosts in same VLAN can communicate to each other

Open Examples → Understanding-VLAN-Operation-in-L2-and-L3-Switches as shown below:



27.2.1 Procedure

Create a network in internetworks as per the below screenshot

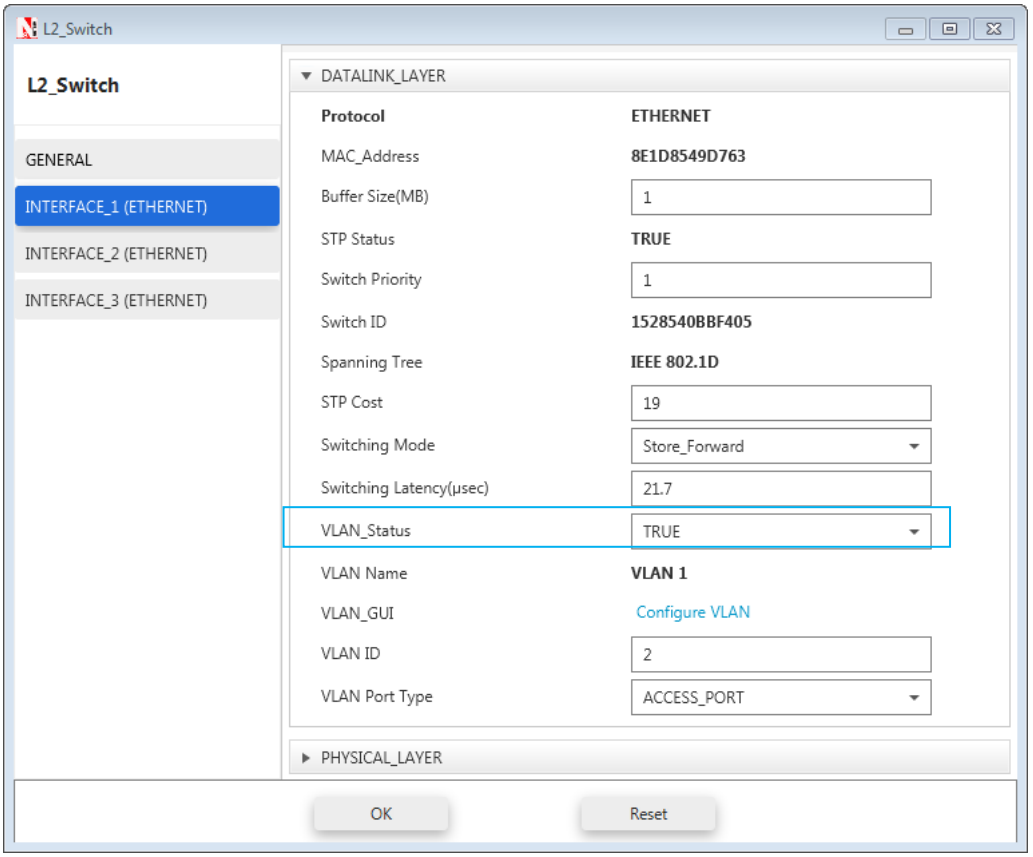


Edit the properties of L2 Switch as per the screenshots below

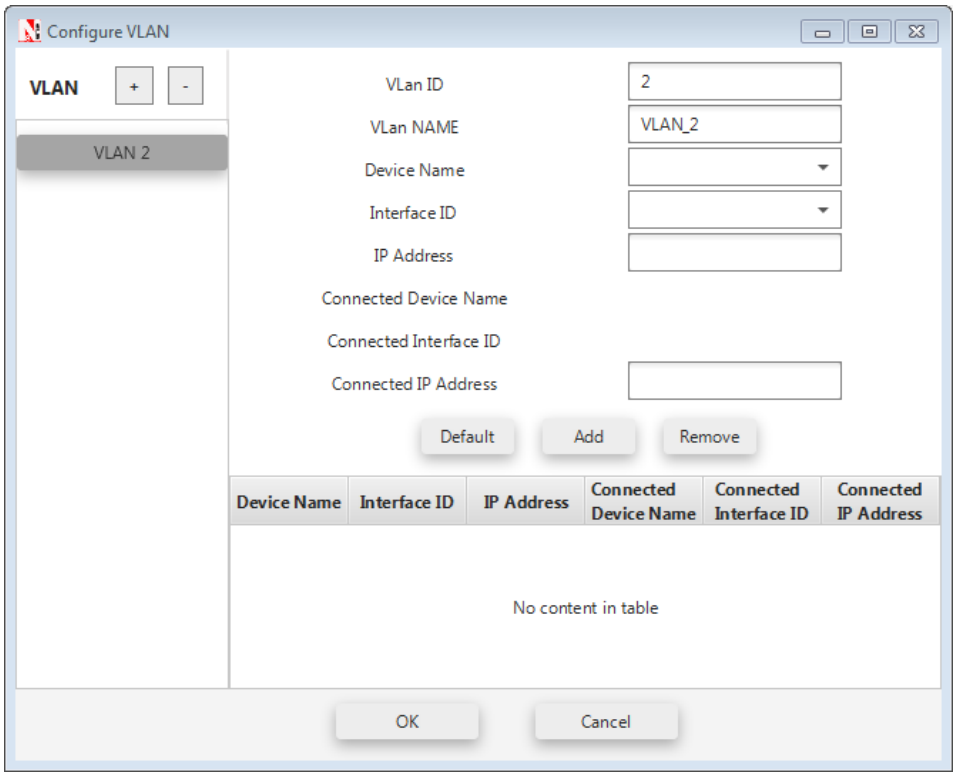
L2 Switch 1			
Interface ID	VLAN Status	VLAN ID	VLAN Port Type

Interface_1	TRUE	2	Access _Port
Interface_2	TRUE	2	Access _Port
Interface_3	TRUE	3	Access _Port

To configure VLAN settings in L2 switch go to VLAN_Status parameter under Interface_1 (Ethernet) and set as TRUE.

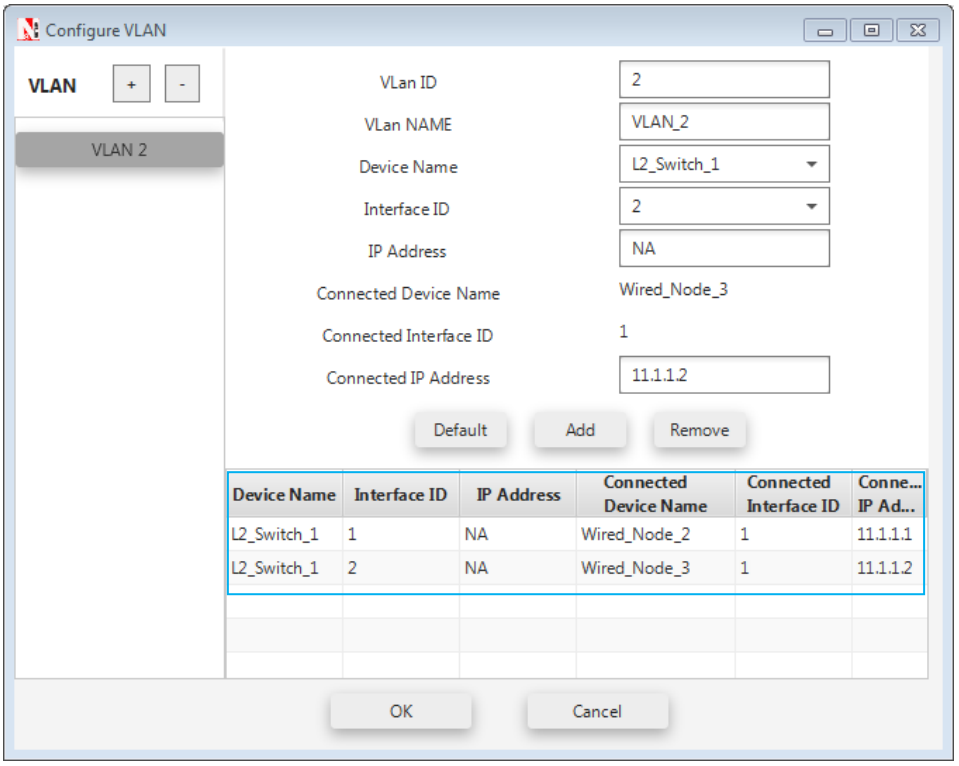


Then click Configure VLAN in VLAN_GUI parameter. The following window will open.

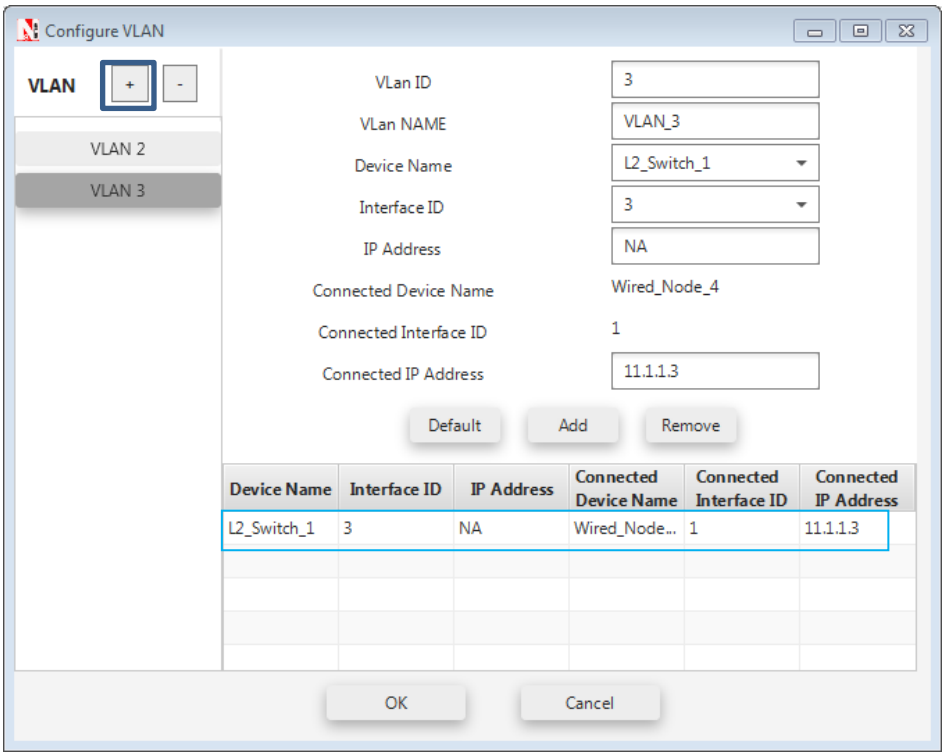


Now set the properties as shown below and after changing the properties click on Add button to add it in the VLAN table

Similarly change the VLAN properties for Interface ID 2 and click on ADD



To add another VLAN click plus icon, after that add the VLAN properties for Interface ID 3



Run simulation for 10 seconds and observe the throughputs.

27.2.2 Output and Inference

Throughput (Mbps)	
Application 1	0.58
Application 2	0

The throughput for 2nd application is zero because the source and destination is in different VLANs, thereby traffic flow or communication between 2 VLANs using Layer2 switch is not possible. To overcome this problem, an L3 switch is used.

27.3 Sample 2 -Inter-VLAN

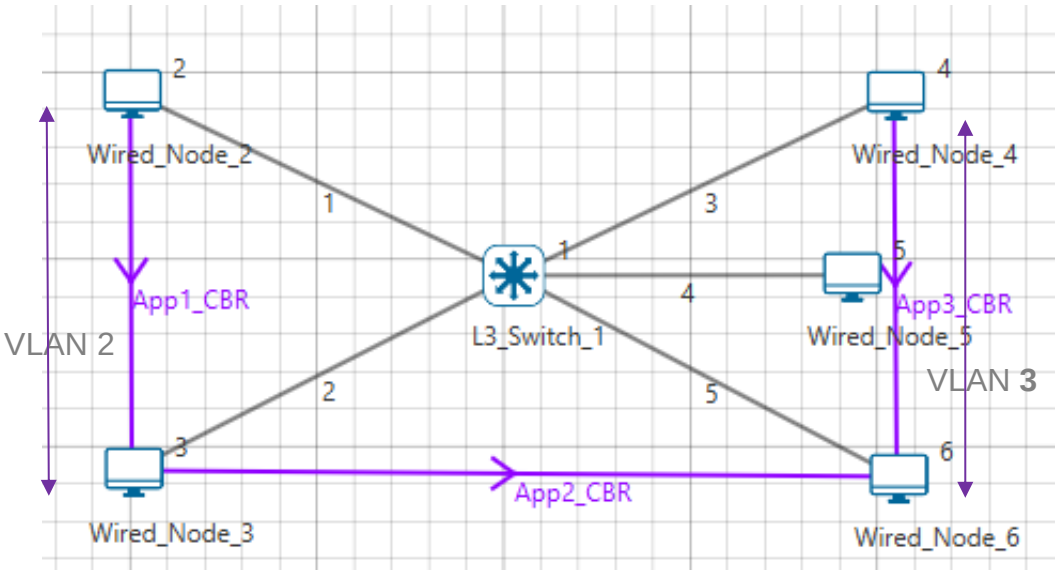
27.3.1 Theory

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

What is a layer 3 switch?

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

27.3.2 Procedure:



Create a network as per the above screenshot. Edit all the wired node properties shown below:

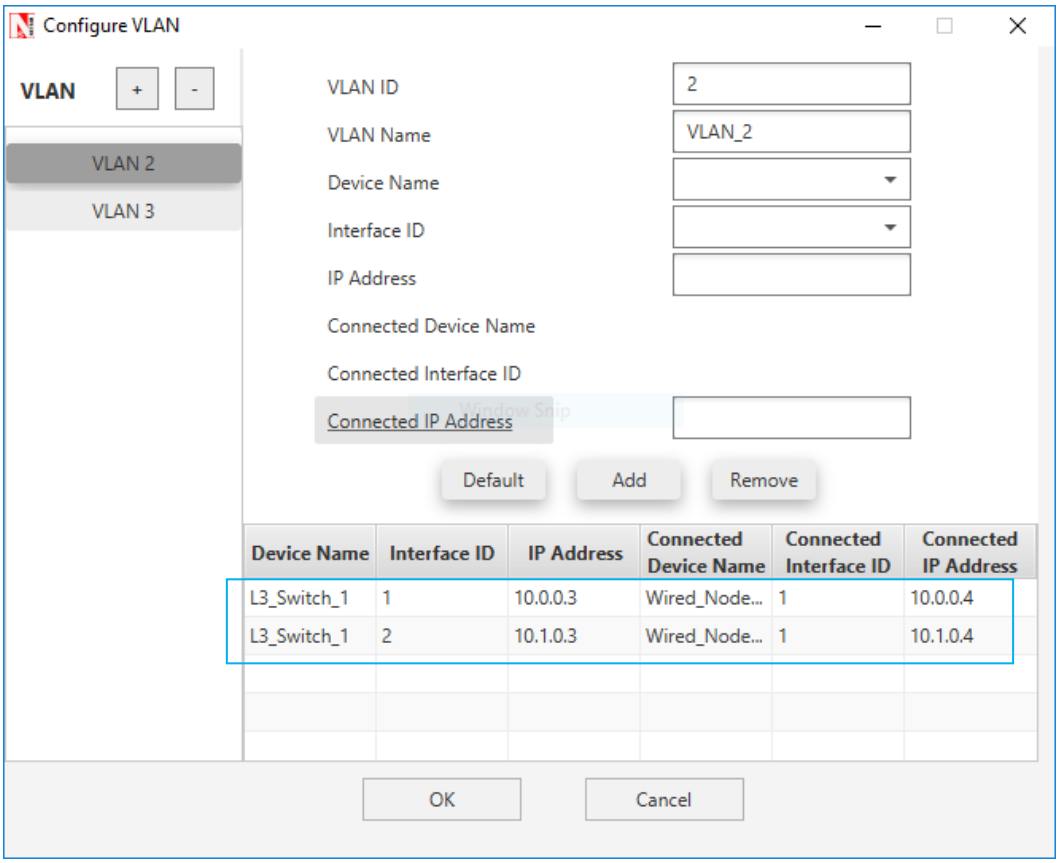
Node	Wired Node2	Wired Node3	Wired Node4	Wired Node5	Wired Node6
	I/f1_Ethernet	I/f1_Ethernet	I/f1_Ethernet	I/f1_Ethernet	I/f1_Ethernet
IP Address	10.0.0.4	10.1.0.4	11.2.0.4	11.3.0.4	11.4.0.4
Default Gateway	10.0.0.3	10.1.0.3	11.2.0.3	11.3.0.3	11.4.0.3

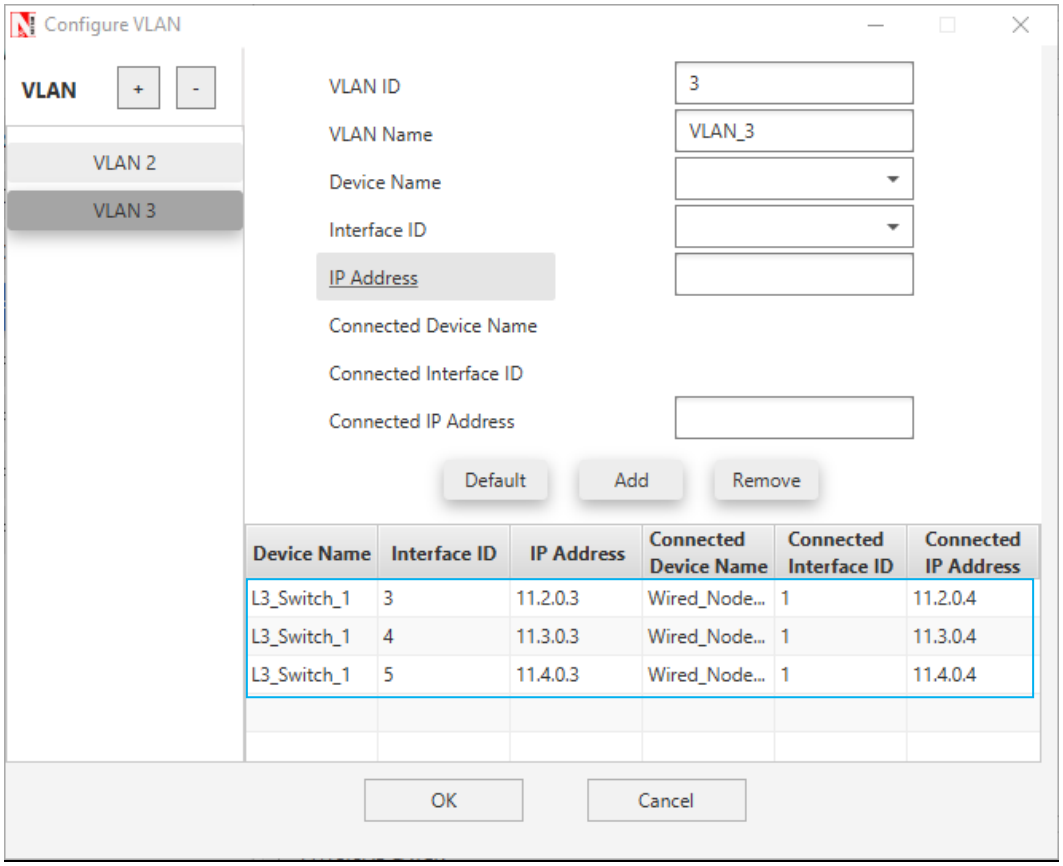
Edit the L3 Switch 1 properties shown below:

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

L3 Switch 1			
Interface ID	VLAN Status	VLAN ID	VLAN Port Type
Interface_1	TRUE	2	Access _Port
Interface_2	TRUE	2	Access _Port
Interface_3	TRUE	3	Access _Port
Interface_4	TRUE	3	Access _Port
Interface_5	TRUE	3	Access _Port

Configure the VLAN properties of L3 Switch 1 as per the below screenshots:





Run simulation for 10 seconds and observe the throughputs.

27.3.3 Output and Inference

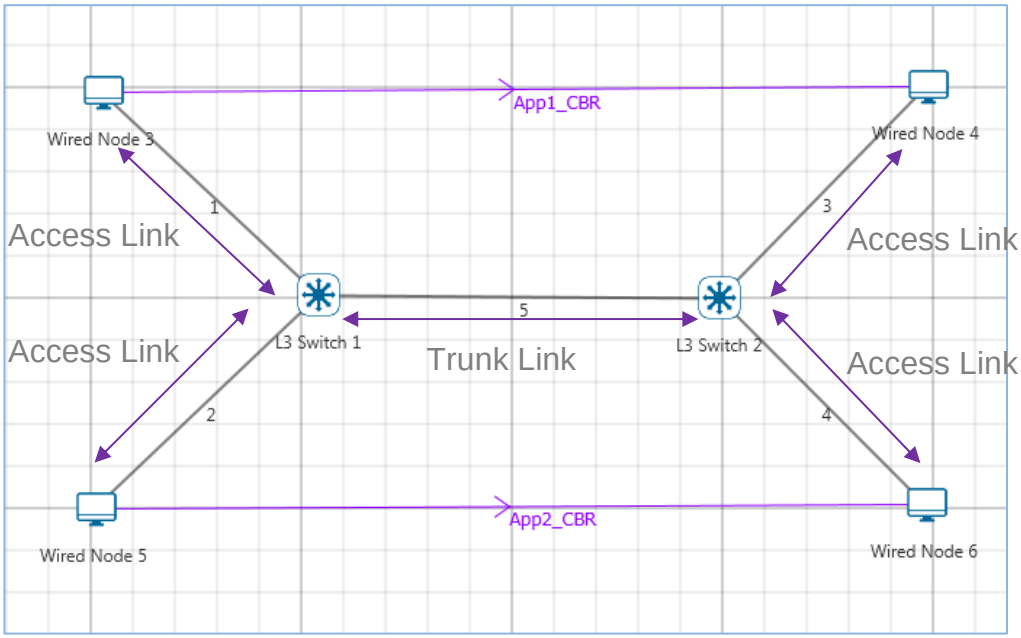
Throughput (Mbps)	
Application 1	**0.58**
Application 2	**0.58**
Application 3	**0.58**

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

28. Understanding Access and Trunk Links in VLANs

28.1 Theory

The links connecting the end devices are called access links. These are the links usually carrying the Data VLAN information. The link between the switches is called trunk link. It carries packets from all the VLANs.



Access link:

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

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

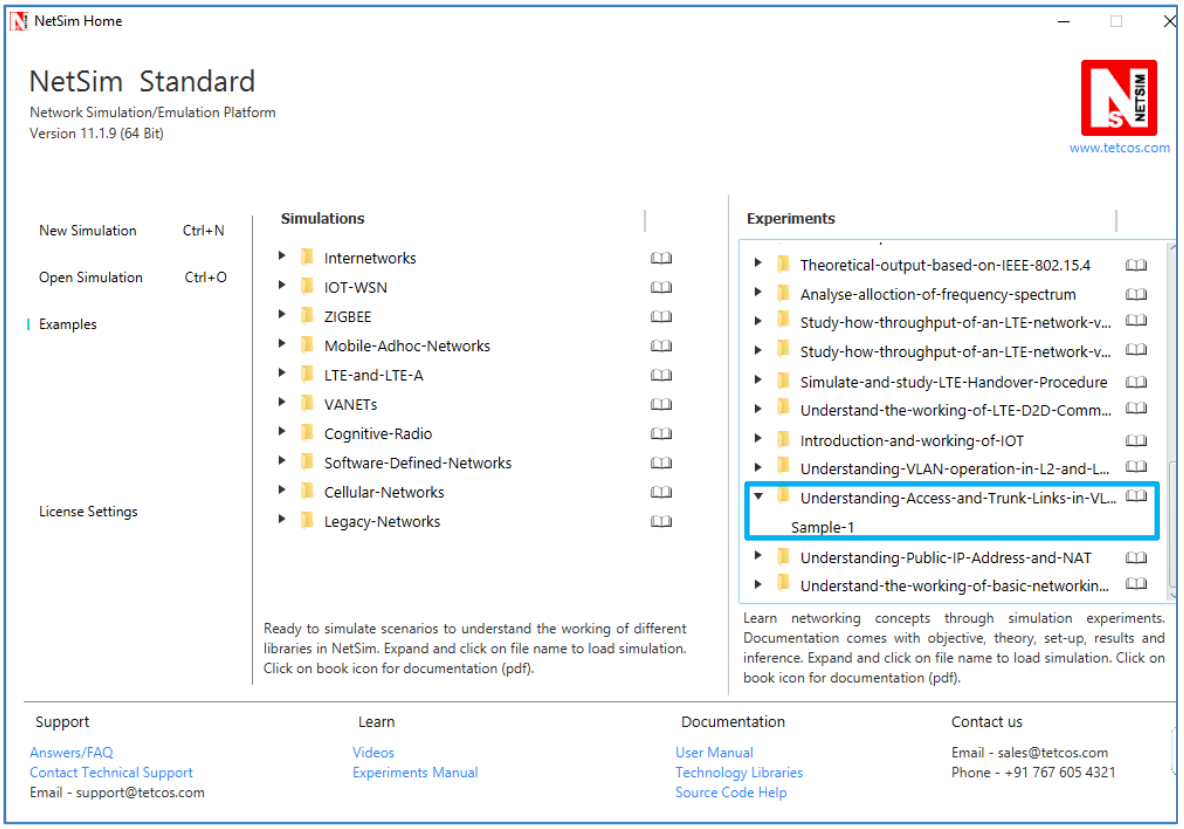
Trunk link:

Trunk link connection is the connection where switch port is connected with a device that is capable to understand multiple VLANs. Usually trunk link connection is used to connect two switches. A

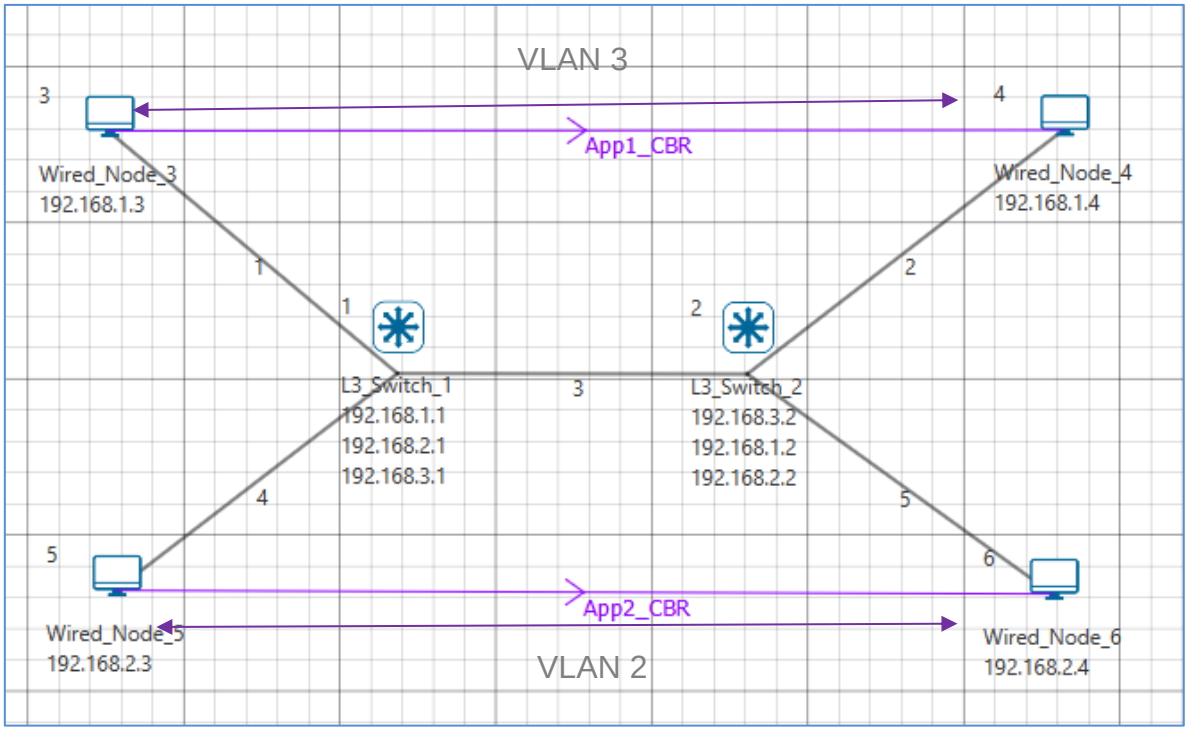
VLAN can span anywhere in network, and that can happen due to trunk link connection. Trunking allows us to send or receive VLAN information across the network. To support trunking, original Ethernet frame is modified to carry VLAN information.

28.2 Procedure

Open Examples → Understanding-Access-and-Trunk-Links-in-VLAN as shown below:



Create a network and edit the properties as per the below screenshot



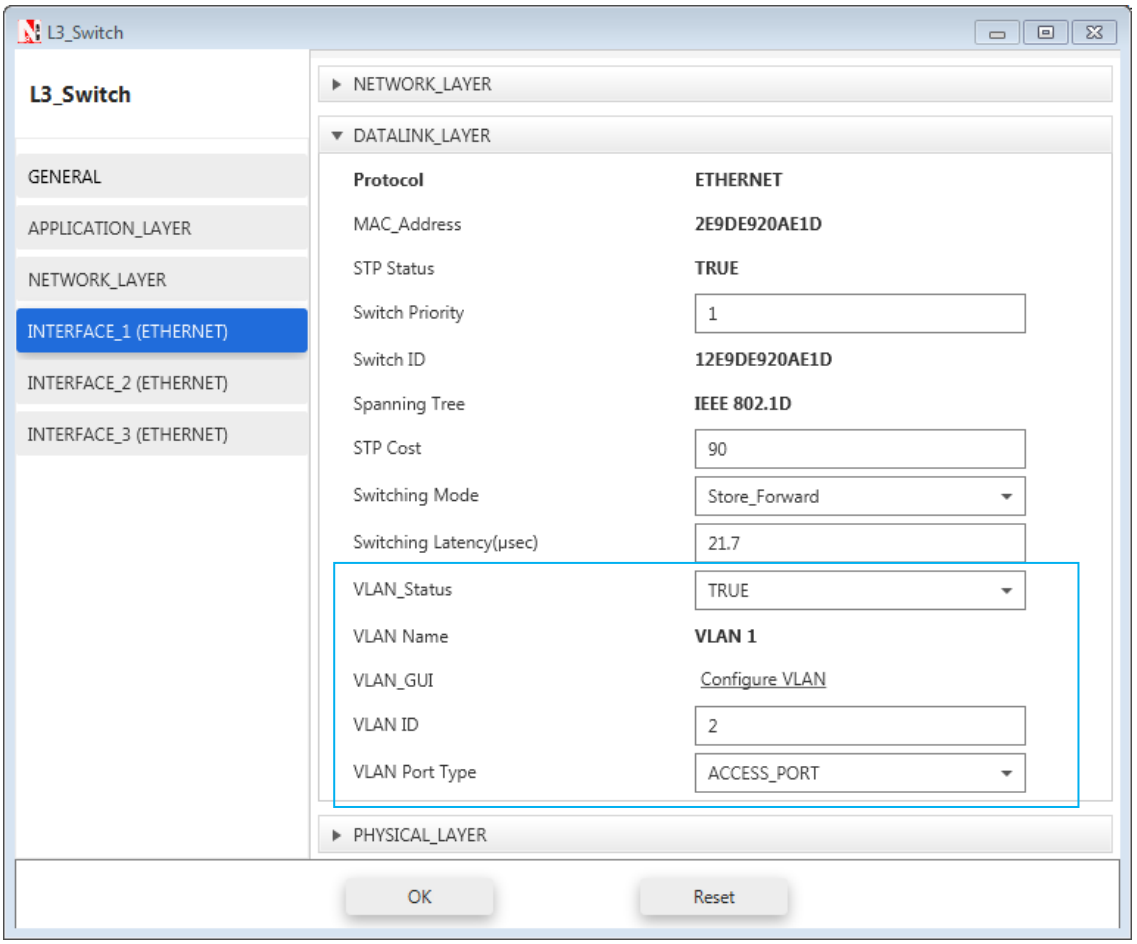
Edit all the wired node properties shown below:

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

Edit the L3 Switch 1 and L3 Switch 2 properties shown below:

Change subnet mask of all L3 Switch interfaces to 255.255.255.0

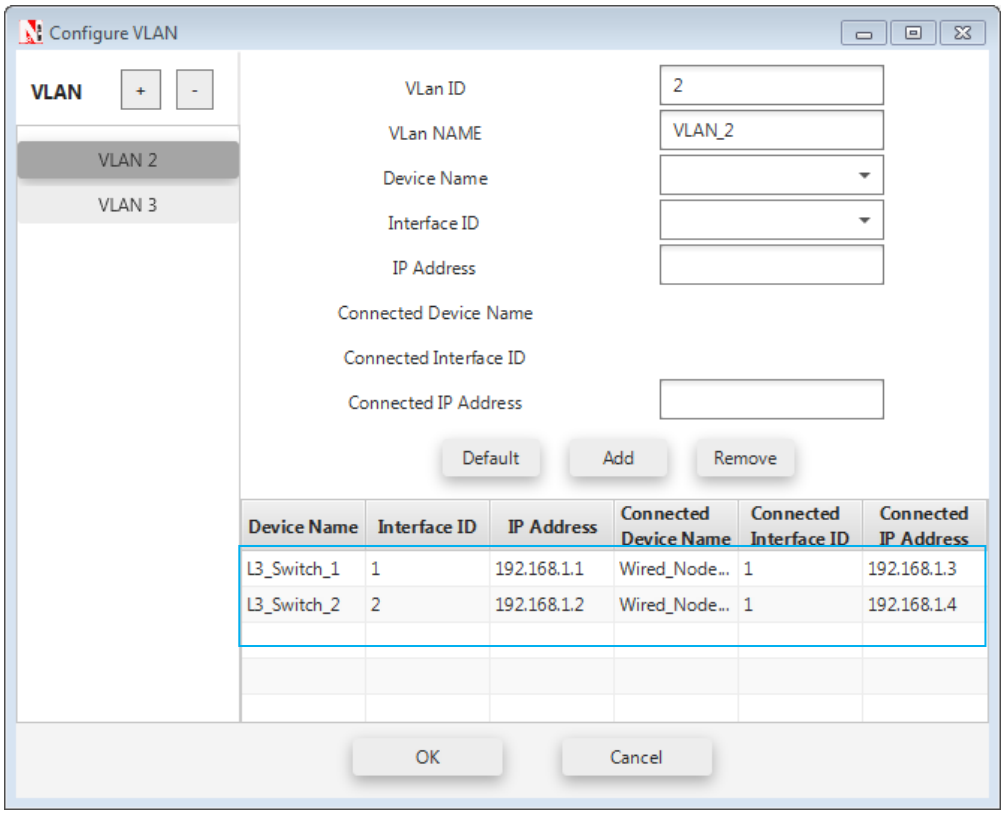
Switch	I/f1_Ethernet	I/f2_Ethernet	I/f3_Ethernet
	IP Address	IP Address	IP Address
L3 Switch 1	192.168.1.1	192.168.2.1	192.168.3.1
L3 Switch 2	192.168.3.2	192.168.1.2	192.168.2.2



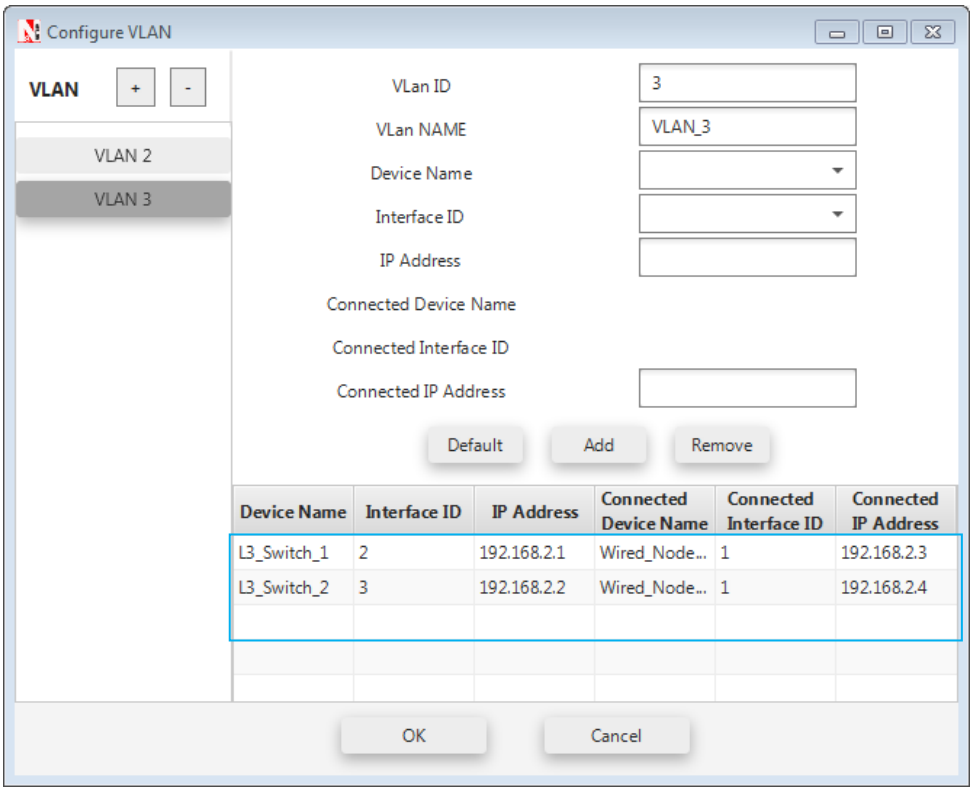
L3 Switch 1			
Interface ID	**VLAN Status**	**VLAN ID**	**VLAN Port Type**
Interface_1	TRUE	2	Access _Port
Interface_2	TRUE	3	Access _Port
Interface_3	TRUE	1	Trunk _Port

L3 Switch 2			
Interface ID	**VLAN Status**	**VLAN ID**	**VLAN Port Type**
Interface_1	TRUE	1	Trunk _Port
Interface_2	TRUE	2	Access _Port
Interface_3	TRUE	3	Access _Port

Click on Configure VLAN in L3_Switch_1 and set the properties for VLAN 2 as per the screenshot shown below

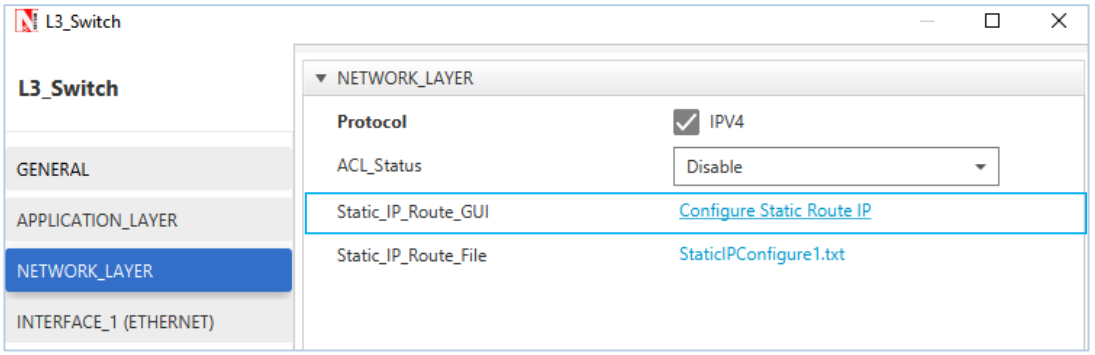


Set the properties for VLAN 3 as per the screenshot



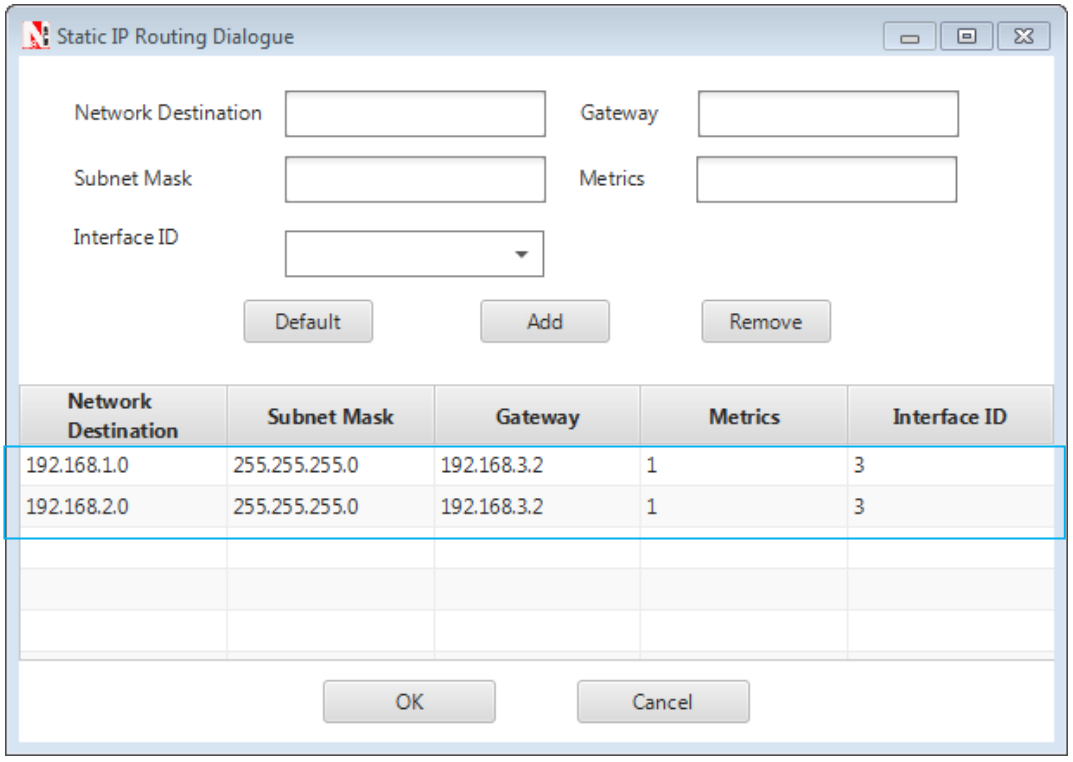
After setting the properties of VLAN2 and VLAN3 click on OK.

Go to L3_Switch_1 properties →Network Layer → Configure Static Route IP



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

Click on **OK**



Note: Disable TCP in Transport Layer in Wired Node 3 and Wired Node 5

Run simulation for 10 seconds and observe the throughput.

28.3 Output:

	Throughput (Mbps)
Application 1	0.57
Application 2	0.57

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

29. Understanding Public IP Address & NAT (Network Address Translation)

29.1 Theory

29.1.1 Public Address:

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

29.1.2 Private Address:

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

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

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

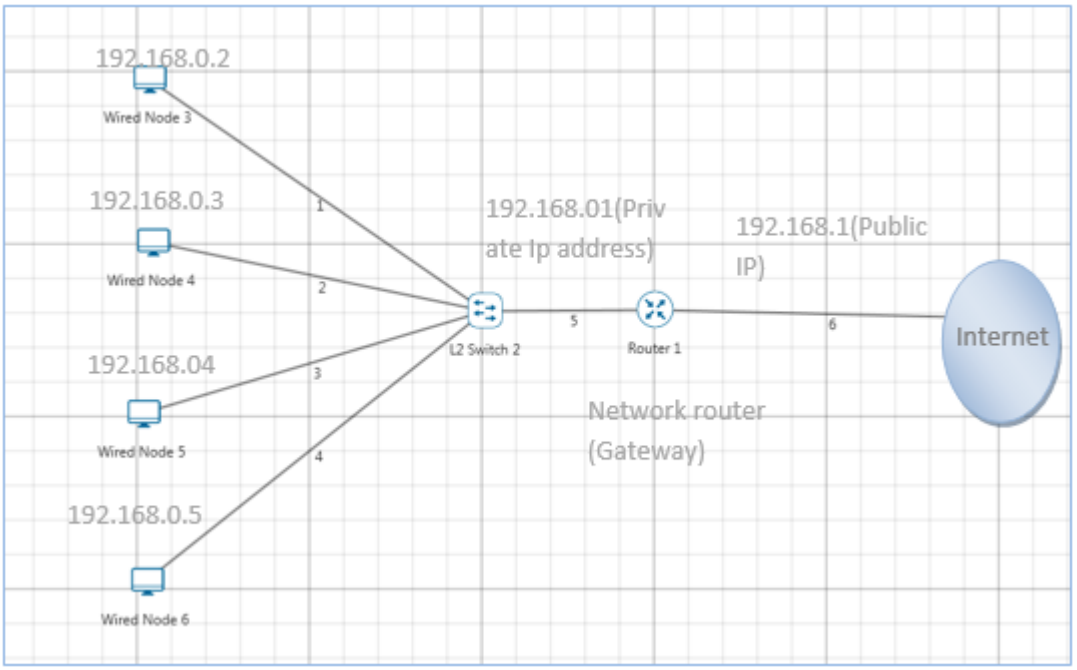
Devices with private IP addresses cannot connect directly to the Internet. Likewise, computers outside the local network cannot connect directly to a device with a private IP. It is possible to interconnect two private networks with the help of a router or a similar device that supports Network Address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

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

29.1.3 Network address translation (NAT)

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

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



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

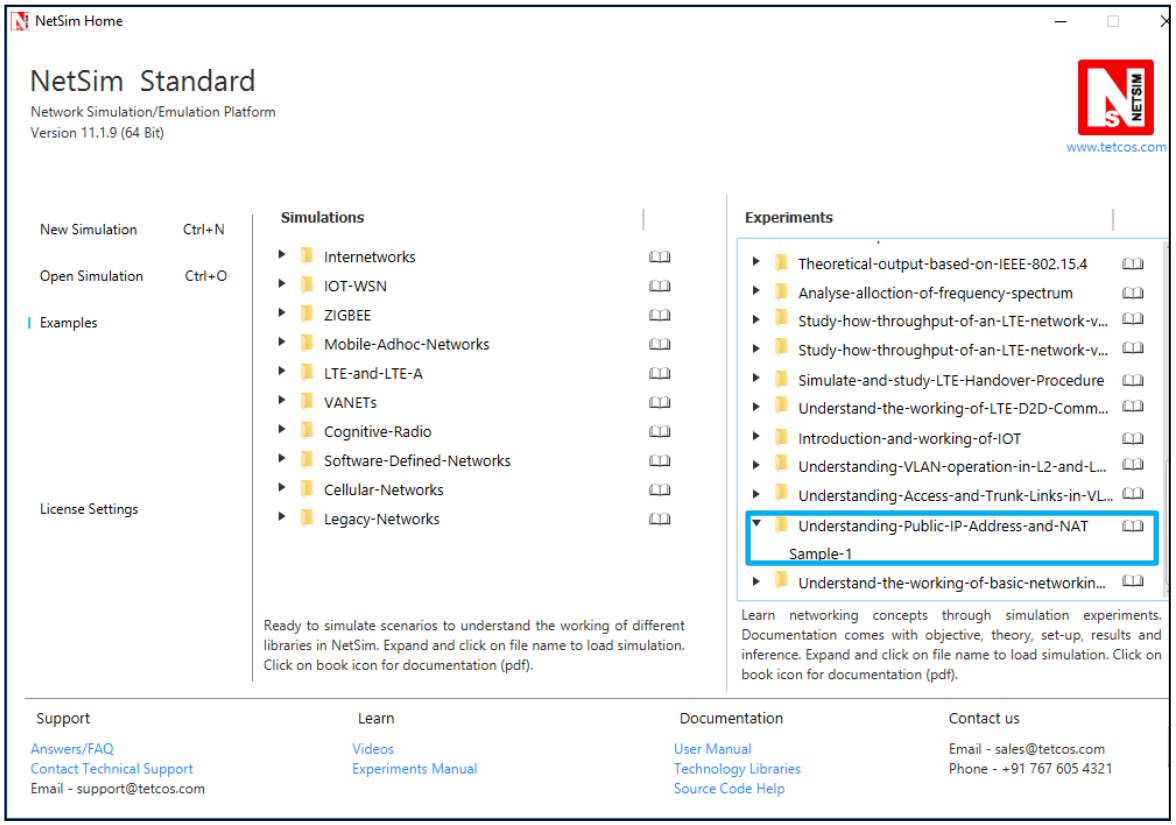
In the above figure, a simple network of 4 hosts and one router that connects this network to the Internet. All hosts in the network have a private Class C IP Address, including the router's private interface (192.168.0.1), while the public interface that's connected to the Internet has a real IP

Address (203.31.220.134). This is the IP address the Internet sees as all internal IP addresses are hidden.

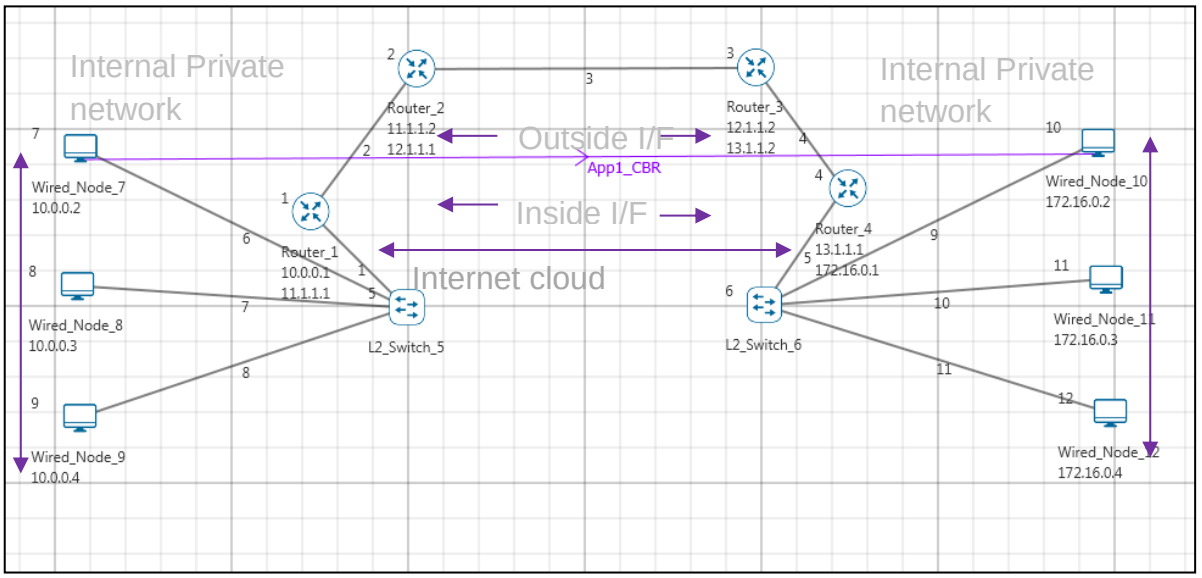
29.2 Network Setup

Working of NAT in NetSim:

Open Examples → Understanding-Public-IP-Address-and-NAT as shown below:



Create a scenario as per the screenshot and set the properties shown below:



Wired node Properties:

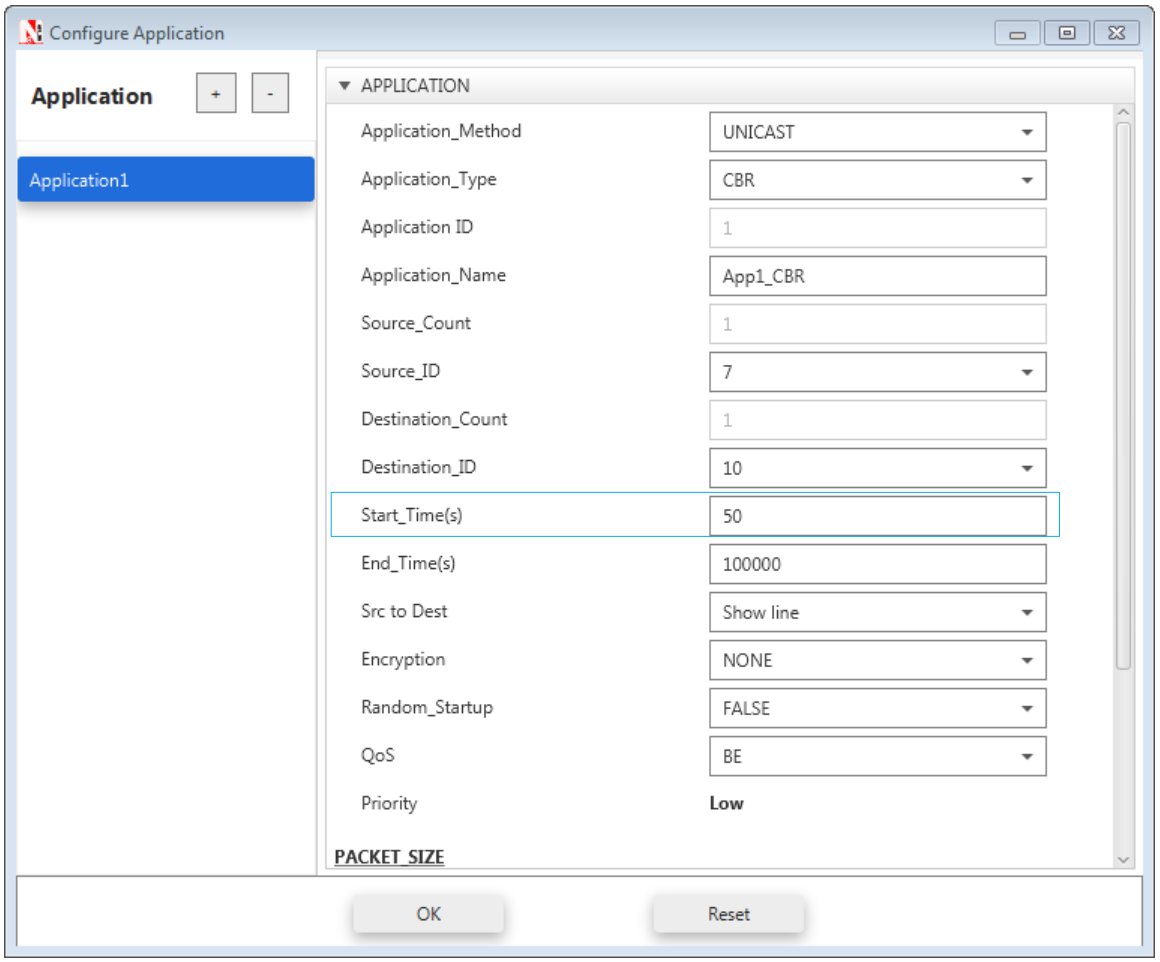
Wired Node	IP address	Subnet mask
7	10.0.0.2	255.0.0.0
8	10.0.0.3	255.0.0.0
9	10.0.0.4	255.0.0.0
10	172.16.0.2	255.255.0.0
11	172.16.0.3	255.255.0.0
12	172.16.0.4	255.255.0.0

Router Properties:

Router	Interface	IP address	Subnet mask
Router 1	**Interface_2(WAN)**	11.1.1.1	255.0.0.0
	Interface_1(Ethernet)	10.0.0.1	255.0.0.0
Router 2	**Interface_1(WAN)**	11.1.1.2	255.0.0.0
	Interface_2(WAN)	12.1.1.1	255.0.0.0
Router 3	**Interface_1(WAN)**	12.1.1.2	255.0.0.0
	Interface_2(WAN)	13.1.1.2	255.0.0.0
Router 4	**Interface_1(WAN)**	13.1.1.1	255.0.0.0
	Interface_2(Ethernet)	172.16.0.1	255.255.0.0

Configure the application with Source_ID as 7 and Destination_ID as 10

Set start time =50



Enable Packet trace and run simulation for 100 seconds. After simulation open packet trace and filter Packet Id to 1

29.3 Inference

PACKET_I	SEGMENT	PACKET_T	CONTROL	SOURCE_I	DESTINATION_I	SOURCE_IP	DESTINATION_IP	GATEWAY_IP	NEXT_HOP_IP
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	10.0.0.1	10.0.0.2	10.0.0.1
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	10.0.0.1	10.0.0.2	10.0.0.1
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	13.1.1.1	11.1.1.1	11.1.1.2
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	13.1.1.1	12.1.1.1	12.1.1.2
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	13.1.1.1	13.1.1.2	13.1.1.1
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	172.16.0.2	172.16.0.1	172.16.0.2
1	0	CBR	App1_CBR	NODE-7	NODE-10	10.0.0.2	172.16.0.2	172.16.0.1	172.16.0.2

SOURCE_IP – source node IP (Node)

DESTINATION_IP – gateway IP (Router/ Node)

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

NEXT_HOP_IP – IP of the next hop (Router/ Node)

Source node 7 (10.0.0.2) wouldn't know how to route to the destination and hence its default gateway is Router 1 with interface IP (10.0.0.1). So, the first line in the above screenshot specifies packet

flow from Source Node 7 to L2 Switch 6 with SOURCE_IP (10.0.0.2), DESTINATION_IP (10.0.0.1), GATEWAY_IP (10.0.0.2) and NEXT_HOP_IP (10.0.0.1). Since Switch is Layer2 device there is no change in the IPs in second line. Third line specifies the packet flow from Router 1 to Router 2 with SOURCE_IP (10.0.0.2), DESTINATION_IP (13.1.1.1- IP of the router connected to destination. Since OSPF is running, the router is looks up the route to its destination from routing table), GATEWAY_IP (11.1.1.1) and NEXT_HOP_IP (11.1.1.2) and so on.

30. Understand the working of basic networking commands (Ping, Route Add/Delete/Print, ACL)

30.1 Theory

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

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, or incorrectly configured

Route Commands

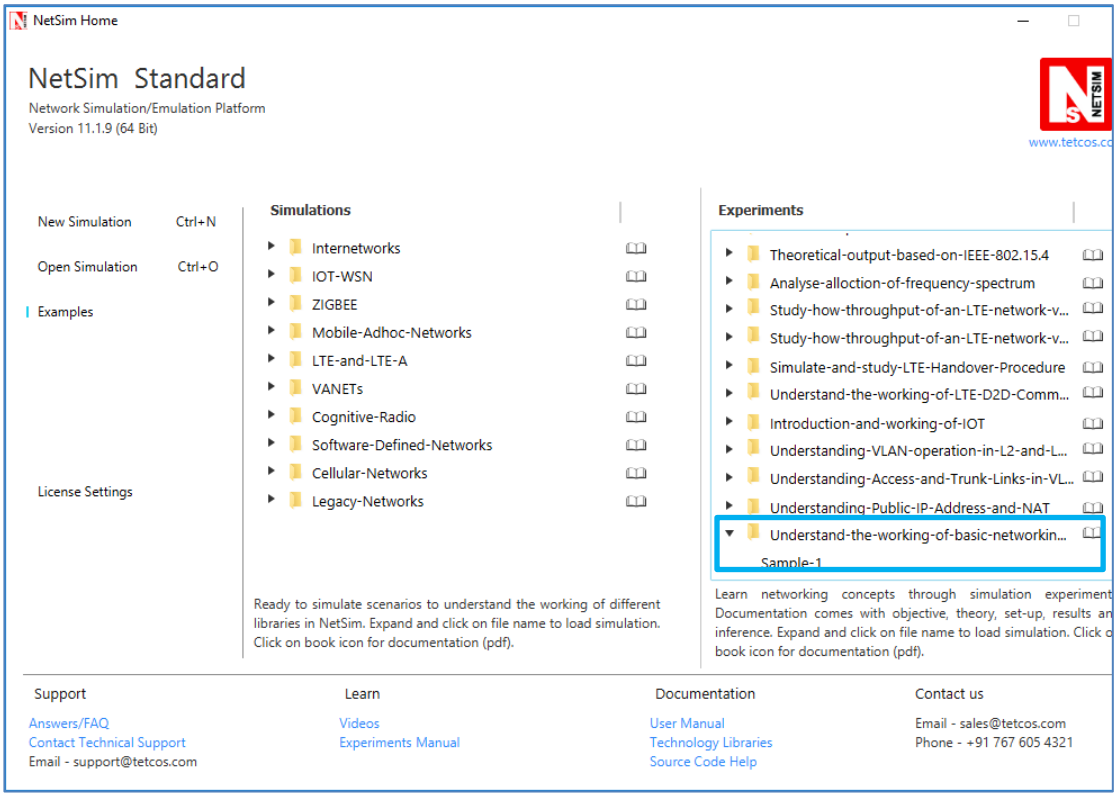
- You can use the route command to view, add and delete routes in IP routing tables
- **route print :** In order to view the entire contents of the IP routing table, issue the route print command
- **route delete:** In order to delete all routes in the IP routing table
- **route add**: To add a static TCP/IP route to the IP routing table

ACL Configuration

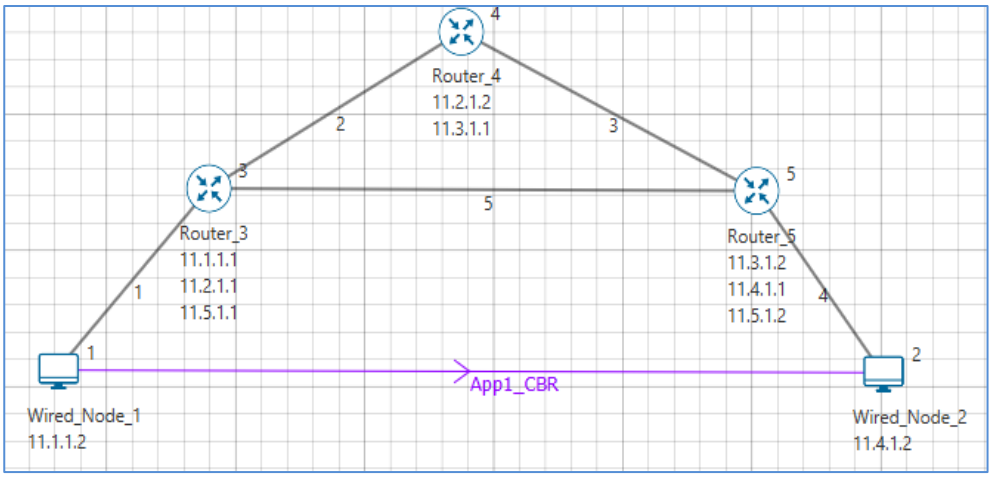
Routers provide basic traffic filtering capabilities, such as blocking Internet traffic, with access control lists (ACLs). An ACL is a sequential list of permit or deny statements that apply to addresses or upper-layer protocols. These lists tell the router what types of packets to: permit or deny. When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

30.2 Network setup

Step 1: Open Examples → Understand-the-working-of-basic-networking-commands as shown below:

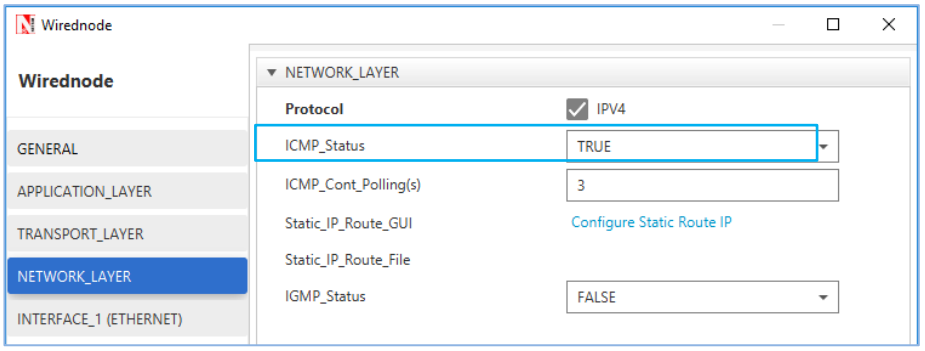


Click & drop Wired Nodes and Router onto the Simulation Environment and link them as shown below.



Step 2:

- ICMP_Status should be set as True in all nodes(Wired_Node and Router)



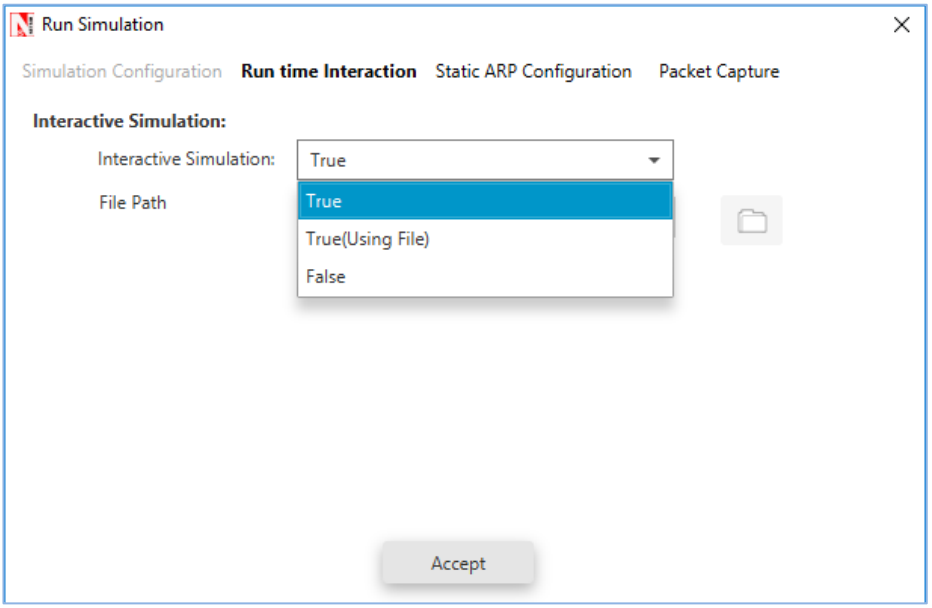
- Right click on Wired_Node_1 and go to properties. Under General properties enable Wireshark Capture option as “Online”

Step 3:

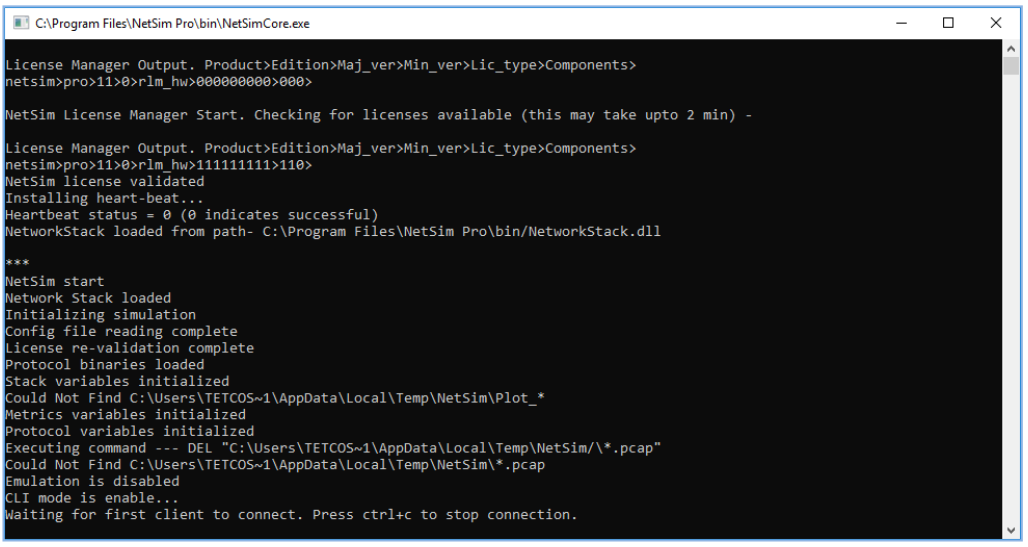
- Click on Application icon present in the top ribbon and set the Application type as CBR. The Source_Id is 1 and Destination_Id is 2
- Set Start Time as 30 Sec
- Enable Plots and Packet trace options

Run Simulation

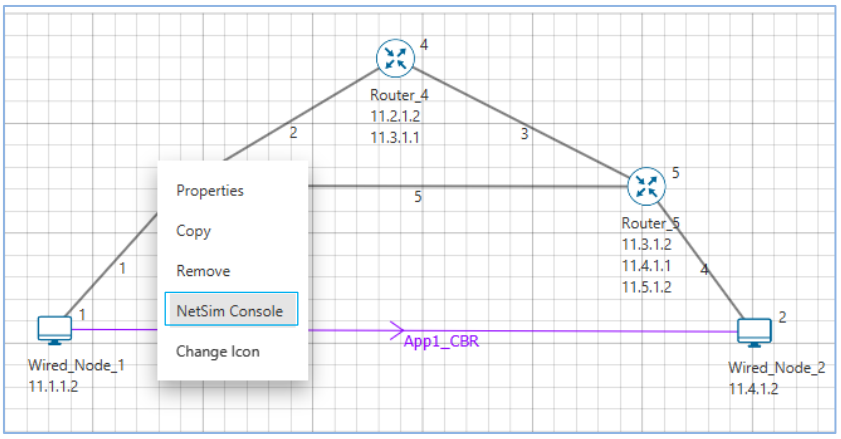
- Click on run simulation option and In the Run time Interaction tab set Interactive Simulation as True and click on Accept
- Set the Simulation Time as 300 sec or more (**It is recommended to specify a longer simulation time to ensure that there is sufficient time for the user to execute the various commands and see the effect of that before Simulation ends)** and click OK



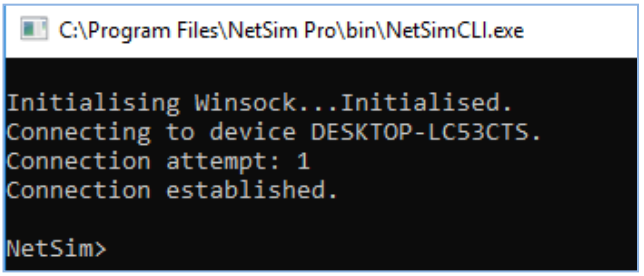
- Simulation (NetSimCore.exe) will start running and will display a message “waiting for first client to connect” as shown below:



- 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 a connection with NetSimCore.exe. After the connection is established the window will look like



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

30.3 Network Commands

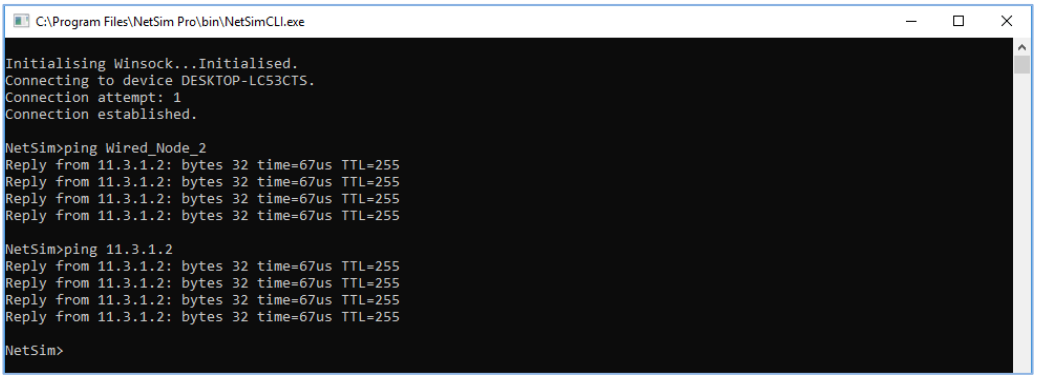
Ping Command:

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

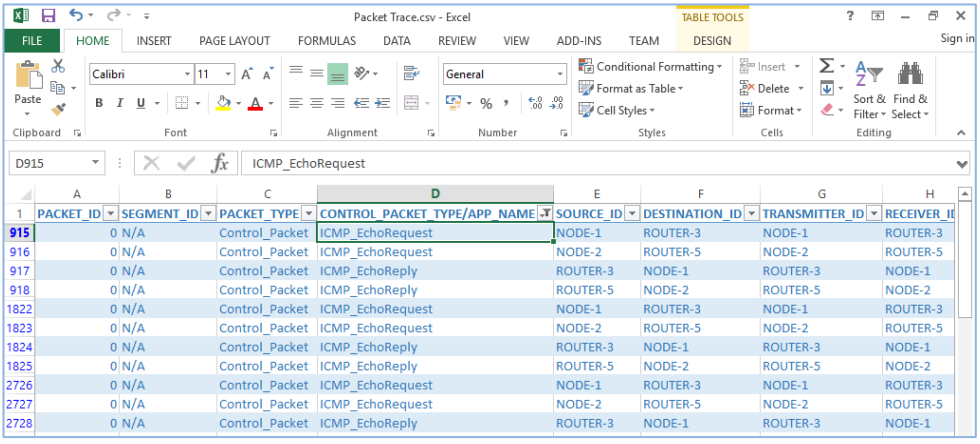
Ping <IP address> e.g. ping 11.4.1.2

Ping <Node Name> e.g. ping Wired_Node_2

Ping Command Results:

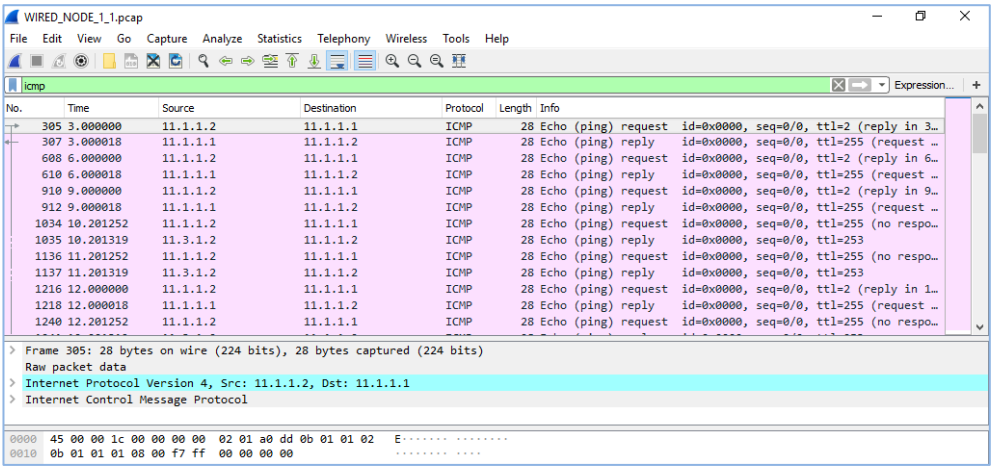


- After simulation open packet trace and filter ICMP_EchoRequest and ICMP_EchoReply from CONTROL_PACKET_TYPE/APP_NAME column



	A	B	C	D	E	F	G	H
1	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

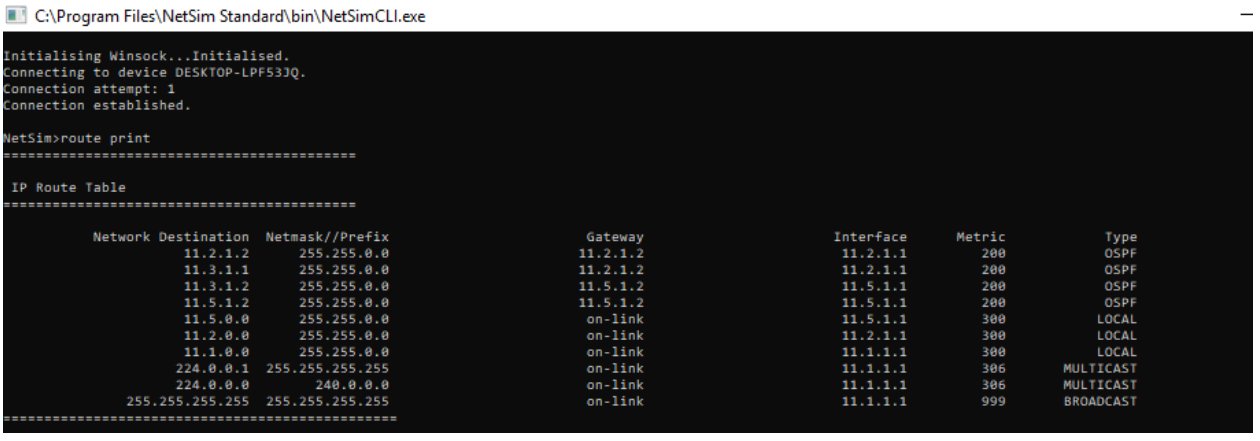


Route Commands:

1. route print
2. route delete
3. route add

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

route print



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

route delete *destination_network*

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

route delete 11.5.1.2

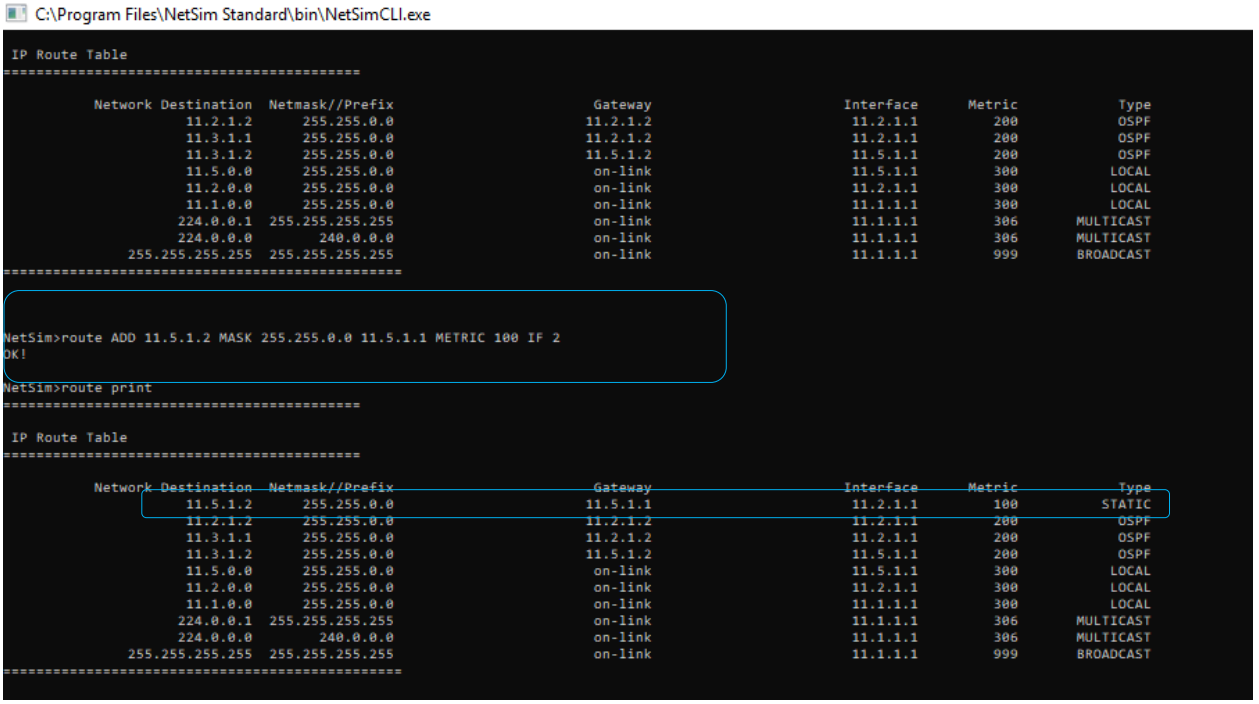
- To check whether route has been deleted or not check again using **route print** command
- To add a static route to the table, you'll type a command using the following syntax

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

- So, for example, if you wanted to add a route specifying that all traffic bound for the 11.5.1.2 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'd see your new static route



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.

ACL Configuration:

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 300 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

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

*ROUTER_3/ACLCONFIG>**exit***

*NetSim>**acl disable***

ACL is disable

NetSim>

```
NetSim>acl enable
ACL is enable

NetSim>aclconfig

ROUTER_3/ACLCONFIG>acl print
Usage: [PERMIT,DENY] [INBOUND,OUTBOUND,BOTH] PROTO SRC DEST SPORT DPORT IFID

ROUTER_3/ACLCONFIG>DENY BOTH UDP ANY ANY 0 0 2
OK!
ROUTER_3/ACLCONFIG>DENY BOTH UDP ANY ANY 0 0 3
OK!
ROUTER_3/ACLCONFIG>print
DENY BOTH UDP ANY/0 ANY/0 0 0 2
DENY BOTH UDP ANY/0 ANY/0 0 0 3

ROUTER_3/ACLCONFIG>exit

NetSim>acl disable
ACL is disable

NetSim>
```

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

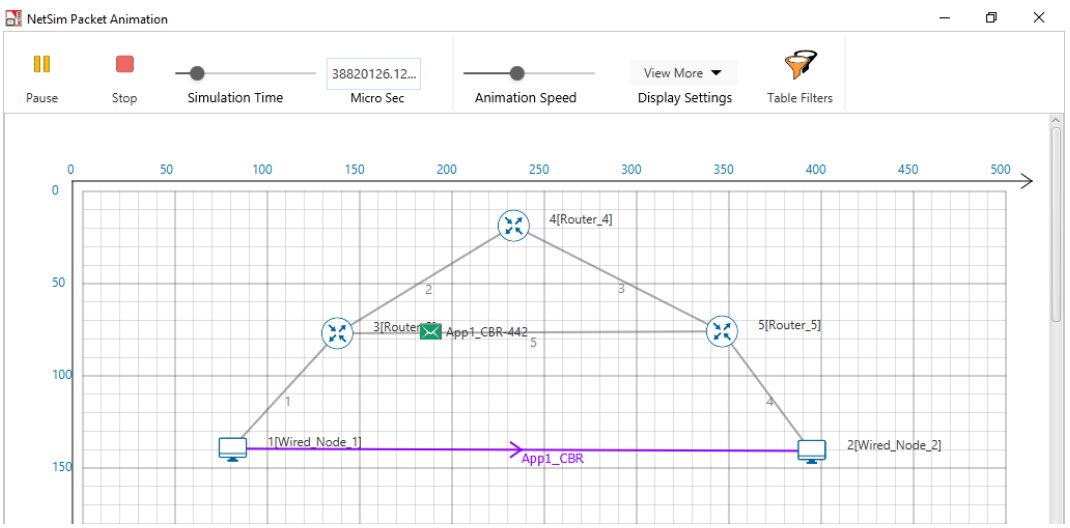
IP_Metrics_Table

IP_Metrics Detailed View

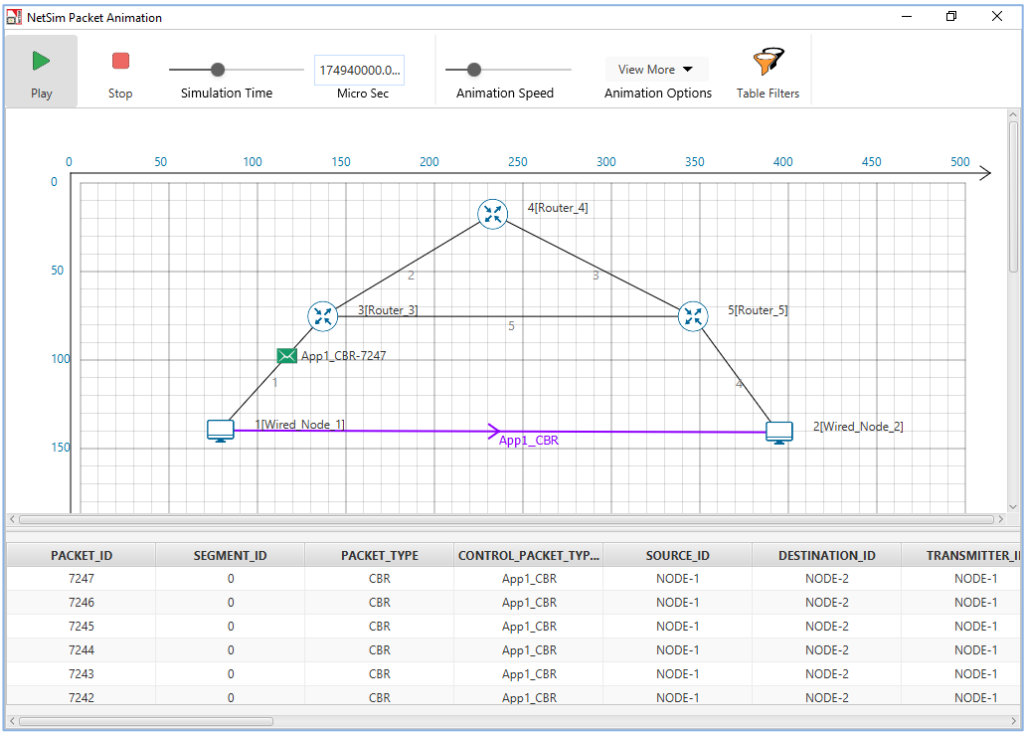
Device Id	Packet sent	Packet forwarded	Packet received	Packet discarded	TTL expired	Firewall blocked
1	13599	0	0	0	0	0
2	99	0	8419	0	0	0
3	8609	13482	72	0	0	5047
4	74	0	74	0	0	0
5	8605	8435	74	0	0	0

Note: Number of packets blocked may vary based on the time at which ACL is configured

- 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



PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYP...	SOURCE_ID	DESTINATION_ID	TRANSMITTER_I
7247	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1
7246	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1
7245	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1
7244	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1
7243	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1
7242	0	CBR	App1_CBR	NODE-1	NODE-2	NODE-1