

# Performance Analysis of Wireless Ad Hoc Networks using NetSim

## Contents

Performance Analysis of Wireless Ad Hoc Networks using NetSim .....	1
Case 1: Single Link Networks: Proximity and Spatial Reuse .....	3
Procedure:.....	3
Network Scenario .....	3
Device Coordinates .....	3
Simulation Parameters.....	4
Static Route configuration.....	4
Results .....	5
Case 2: Ad Hoc Chain, Single Hop Communication: Two-Hop CS Range.....	6
Procedure:.....	6
Network Scenario .....	7
Device Coordinates .....	7
Simulation Parameters.....	8
Static Route configuration.....	8
Results .....	9
Plotting Average Throughput vs Number of Transmitting Nodes.....	9
Border effects in line networks: Plotting Throughput vs Transmitting Node ID .....	11
Case 3: Ad Hoc Lattice, Single Hop Communication, Two Hop CS Range: Horizontal Traffic ..	14
Procedure:.....	14
Network Scenario .....	15
Device Coordinates .....	15
Simulation Parameters.....	16
Static Route configuration.....	16
Results .....	17
Plotting Average Throughput vs Number of Transmitting Nodes.....	17
Case 4: 4 Hop Network .....	19
Using NetSim's GUI. Importing the project into NetSim. ....	19
Simulation Parameters.....	21
Static Route Configuration .....	22
Numerical Results .....	23
Automated simulation using multi-parameter sweeper .....	24
Changes done to the config file to run the Multi-Parameter sweeper .....	24
Results .....	27
Case 5: 5 Hop Network .....	30
Additional Setting .....	30
Numerical Results .....	30

Changes done to the config file to run the multi-Parameter sweeper .....	31
Case 06: 6- Hop Network .....	32
Additional Setting .....	33
Numerical Results .....	33
Changes done to the config file to run the multi-Parameter sweeper .....	34
Case 07: 7-Hop Network .....	35
Additional Setting .....	35
Numerical Results .....	36
Changes done to the config file to run the multi-Parameter sweeper .....	36
Appendix .....	38
Source code modifications involving in Case 1, Case 2, Case 3. ....	38
Source code modifications involving in Case 4, Case 5, Case 6, Case 7 .....	40

## Case 1: Single Link Networks: Proximity and Spatial Reuse

NetSim UI displays the following Network Scenarios

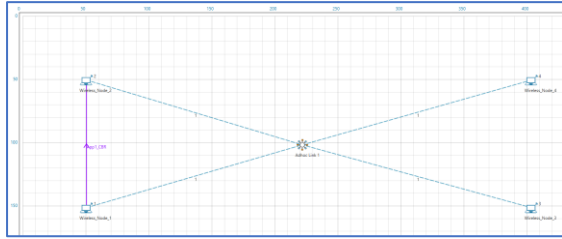


Fig 4: Network Scenario for Case 1-a

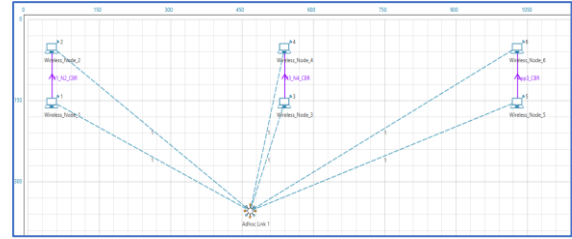


Fig 7: Network Scenario for Case 1-d

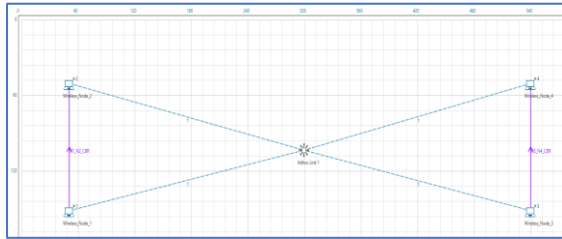


Fig 5: Network Scenario for Case 1-b

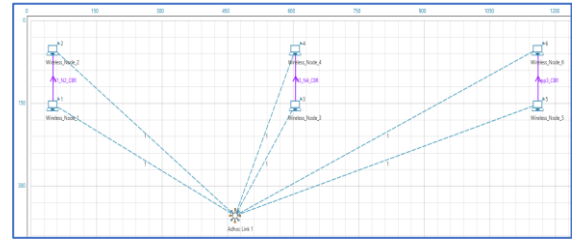


Fig 8: Network Scenario for Case 1-e

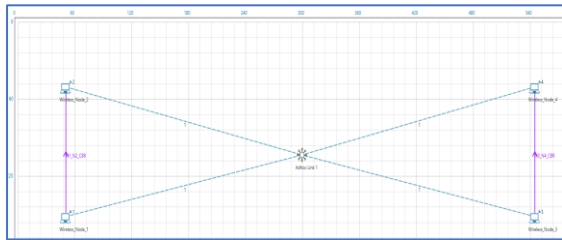


Fig 6: Network Scenario for Case 1-c

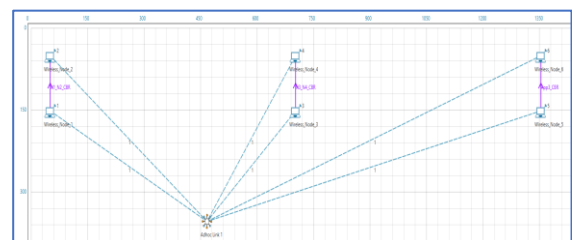


Fig 9: Network Scenario for Case 1-f

## Device Coordinates

The devices were placed in the following coordinates for each of the cases shown above.

Case #	Description	Node 1	Node 2	Node 3	Node 4	Node 5	Node 6
Case 1 a	2 Nodes: N1, N2. Src-Dest separation 100m	(50,150)	(50,50)	-	-	-	-
Case 1 b	4 Nodes: N1, N2, N3, N4 Src-Dest separation 100m Src-Src separation 485m	(50,150)	(50,50)	(535,150)	(535,50)	-	-
Case 1 c	4 Nodes: N1, N2, N3, N4 Src-Dest separation 100m Src-Src separation 490m	(50,150)	(50,50)	(540,150)	(540,50)	-	-
Case 1 d	6 Nodes: N1, N2, N3, N4, N5, N6 Src-Dest separation 100m Src-Src separation 485m	(50,150)	(50,50)	(535,150)	(535,50)	(1020,150)	(1020,50)
Case 1 e	6 Nodes: N1, N2, N3, N4, N5, N6 Src-Dest separation 100m Src-Src separation 550m	(50,150)	(50,50)	(600,150)	(600,50)	(1150,150)	(1150,50)
Case 1 f	6 Nodes: N1, N2, N3, N4, N5, N6 Src-Dest separation 100m Src-Src separation 646m	(50,150)	(50,50)	(696,150)	(696,50)	(1342,150)	(1342,50)

Table 1: Description of the 5 cases under study and the device X, Y co-ordinates in each case

## Simulation Parameters

1. The Grid length was set to  $500 \times 500$  for Case 1-a,  $600 \times 600$  for Case 1-b and Case 1-c,  $1500 \times 1500$  for Case 1-d, Case 1-e and Case 1-f.
2. The simulation Parameters were configured as follows:

802.11 Parameters	
Standard	IEEE802.11g
Operating Frequency	2.4GHz
Rate Adaptation	False
Dot11_RTS Threshold	3000bytes
Frequency Band	2.4GHz
Bandwidth	20MHz
Transmitter Power	100mW
Medium Access Control	DCF

Table 2: Values set for different parameters in simulation

Wireless Link Parameters	
Channel Characteristics	Path Loss Only
Path Loss Model	Log Distance
Pathloss Exponent ( $\eta$ )	2.6
Simulation Duration	
Simulation Time	10 seconds

Table 3: Wireless Link parameter and Simulation time

Application Parameters	
Application Type	CBR
Source ID	Node_1, 3 and 5
Destination ID	Node_2, 4 and 6
Packet Size	1460 bytes
Packet Size Distribution	Constant
Inter-Arrival Time	467.2 microseconds (25 Mbps Application Rate. Saturated queues)
Transport Protocol	UDP

Table 4: Application properties set in this experiment

Protocol Parameters	Value
CS Threshold	-85 dBm
SIFS	$10 \mu s$
Slot Time	$9 \mu s$
DIFS (SIFS + $2 \times$ Slot Time)	$28 \mu s$
Preamble Time	$20 \mu s$
MPDU Size	1528 B

Table 5: 802.11 g protocol parameters. These values are set in the code

## Static Route configuration

Static routes were configured in each source node such that the data gets transmitted directly from source to destination without any dynamic route formation by the routing protocols.

Static routes (whereby  $N_i$  always transmits to  $N_{(i+1)}$ ) are set to ensure single hop transmission. Thereby each node transmits data to the next-hop node according to the topology.

To set the static routes, go to Wireless\_Node properties > Network Layer > Enable Static Route IP. The Static route IP were configured in Wireless\_Node\_1, Wireless\_Node\_3 and Wireless\_Node\_5 as shown below:

Device	Network Destination	Gateway	Subnet Mask	Metrics	Interface ID
Wireless_Node_1	11.1.1.0	11.1.1.2	255.255.255.255	1	1
Wireless_Node_3	11.1.1.0	11.1.1.4	255.255.255.255	1	1
Wireless_Node_5	11.1.1.0	11.1.1.6	255.255.255.255	1	1

Table 6: Static route configured in devices for Cases 1- a, b, c, d, e and f

## Results

The per application throughput is noted down from the NetSim Results window and tabulated in Table 8 of the main document.

## Case 2: Ad Hoc Chain, Single Hop Communication: Two-Hop CS Range

### Procedure:

In this section we explain how users can run the simulation and obtain results that match those provided in the main document

1. Use the following download Link to download a compressed zip folder which contains the workspace: [https://github.com/NetSim-TETCOS/Performance-Analysis-Wireless\\_Adhoc\\_Network\\_v13.1/archive/refs/heads/main.zip](https://github.com/NetSim-TETCOS/Performance-Analysis-Wireless_Adhoc_Network_v13.1/archive/refs/heads/main.zip)
2. Extract the zip folder.
3. The extracted project folder consists of a NetSim workspace file (WiFi\_Adhoc\_Performance\_v13\_1.netsimexp).
4. Go to NetSim Home window, go to Your Work and click on Import.

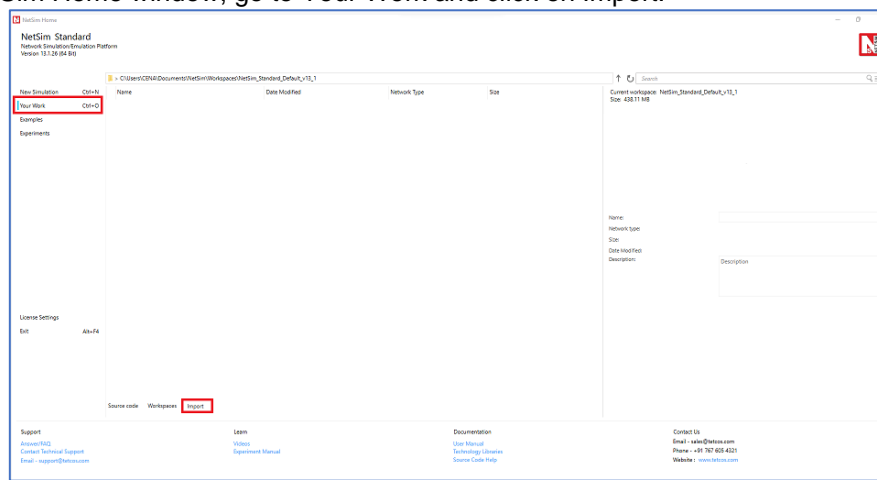


Fig 10: NetSim Home Window

5. In the Import Workspace Window, browse and select the WiFi\_Adhoc\_Performance\_v13\_1.netsimexp file from the extracted directory. Click on create a new workspace option and browse to select a path in your system where you want to set up the workspace folder.
6. Choose a suitable name for the workspace of your choice. Click Import.

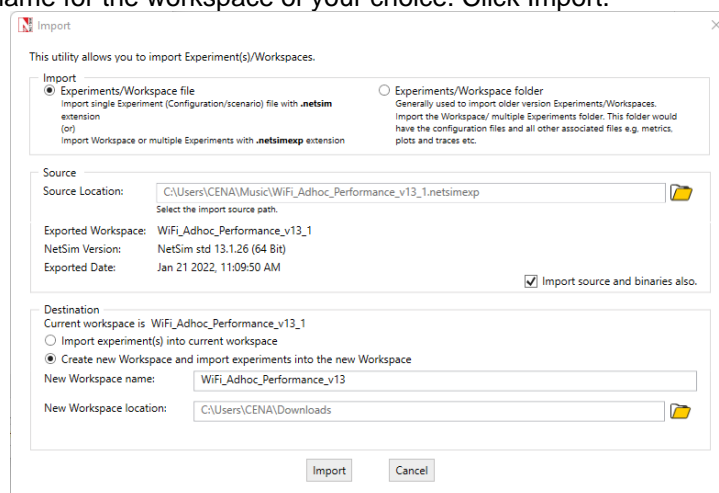


Fig 11: NetSim Import workspace window

7. The Imported Project workspace will automatically be set as the current workspace.
8. The list of experiments is now loaded onto the selected workspace.
9. The Experiments/ configuration files associated with the case mentioned above are available in the folder "Case\_2\_Adhoc\_Chain\_Single\_Hop\_Communication\_Two\_Hop\_CS\_Range" inside the workspace" inside the workspace.

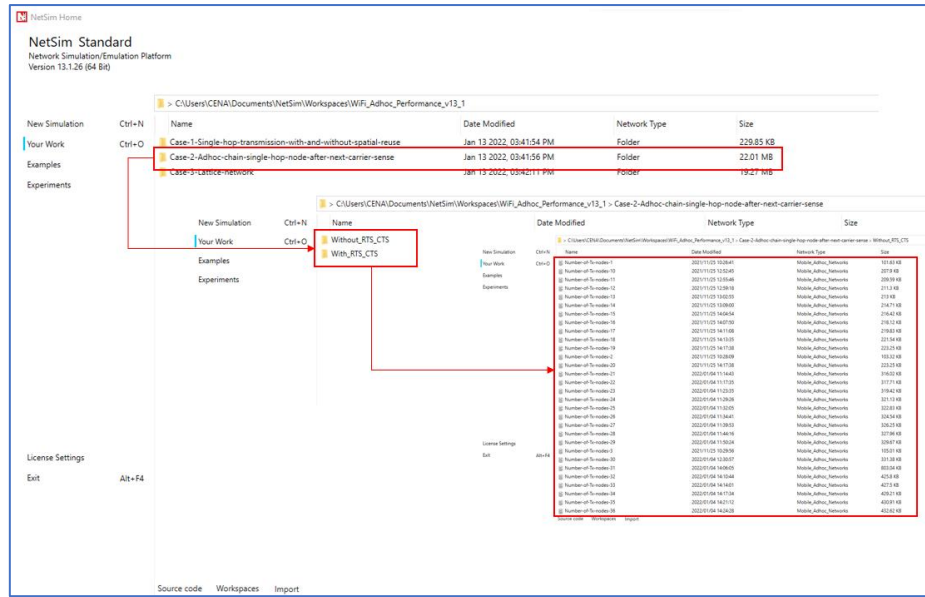


Fig 12: NetSim Your work window with experiments for Case 2- Adhoc chain, single hop communication two hop CS Range- With RTS/CTS and Without RTS/CTS

## Network Scenario

NetSim UI displays the following Network Scenarios.

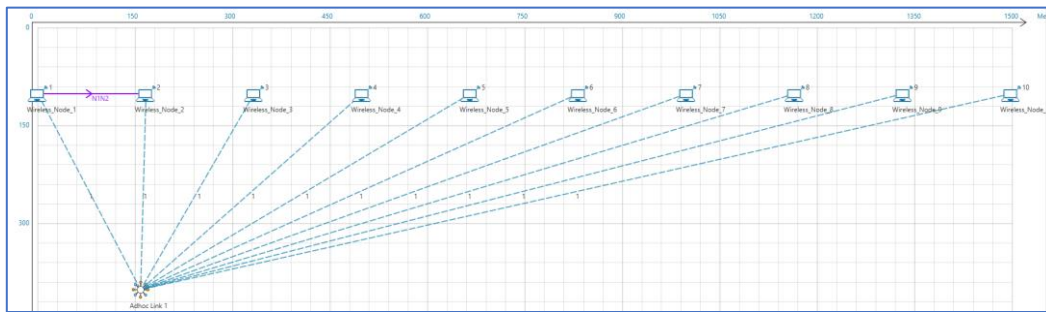


Fig 13: Network Scenario for Case with one node transmitting. The number of transmitting nodes are increased from 1 to 49.

## Device Coordinates

The distance setting should be such that (i) N1-N2 are in transmission range, (ii) N1-N3 are in CS range but not in transmission range, and (iii) N1-N4 are beyond in CS range. These conditions are met when the distance to the immediate neighbour is 165m. Thus, the placement would be  $N1(0,100)$ ,  $N2(165,100)$ ,  $N3(330,100)$  ... and so on

The devices were placed in the following coordinates for each of the cases shown above

Case #	Description	Node 1	Node 2	Node 3	Node 4	Node 5	Node <sub>i+1</sub>	Node 50
1	Number of Transmitting Nodes=1	(0,100)	(165,100)	-	-	-	-	-
2	Number of Transmitting Nodes=2	(0,100)	(165,100)	(330,100)	-	-	-	-
3	Number of Transmitting Nodes=3	(0,100)	(165,100)	(330,100)	(495,100)	-	-	-
4	Number of Transmitting Nodes=4	(0,100)	(165,100)	(330,100)	(495,100)	(660,100)	-	-
49	Number of Transmitting Nodes=49	(0,100)	(165,100)	(330,100)	(495,100)	(660,100)	(x+165, 100)	(7920,100)

Table 7: X and Y coordinates of devices in the network scenario. The devices are placed 165m away from each other.

## Simulation Parameters

1. The Grid length was set to  $8500 \times 8500$ .
2. The simulation Parameters were configured as follows:

802.11 Parameters	
Standard	IEEE802.11g
Operating Frequency	2.4GHz
Rate Adaptation	False
Dot11_RTS Threshold	3000B and 1000B
Frequency Band	2.4GHz
Bandwidth	20MHz
Transmitter Power	100mW
Medium Access Control	DCF

Table 8: Values set for different parameters in simulation

Wireless Link Parameters	
Channel Characteristics	Path Loss Only
Path Loss Model	Log Distance
Pathloss Exponent ( $\eta$ )	2.6
Simulation Duration	
Simulation Time	100 seconds

Table 9: Wireless Link parameter and Simulation time

Application Parameters	
Application Type	CBR
Source ID	$Node_i$ where $i = 1, 2, \dots$ etc 49
Destination ID	$Node_{i+1}$ where $i = 1, 2, \dots$ etc 49
Packet Size	1460 bytes
Packet Size Distribution	Constant
Inter-Arrival Time	973.33 microseconds (12 Mbps Application Rate. Saturated queues)
Transport Protocol	UDP

Table 10: Application properties set in this experiment

Protocol Parameters	Value
CS Threshold	-85 dBm
SIFS	$10 \mu s$
Slot Time	$9 \mu s$
DIFS (SIFS + $2 \times$ Slot Time)	$28 \mu s$
Preamble Time	$20 \mu s$
MPDU Size	1528 B

Table 11: 802.11 g protocol parameters. These values are set in the code

## Static Route configuration

Static routes were configured in each source node such that the data gets transmitted directly from source to destination without any dynamic route formation by the routing protocols.

Static routes (whereby  $N_i$  always transmits to  $N_{(i+1)}$ ) are set to ensure single hop transmission. Thereby each node transmits data to the next-hop node according to the topology.

To set the static routes, go to Wireless\_Node properties > Network Layer > Enable Static Route IP.

Device	Network Destination	Gateway	Subnet Mask	Metrics	Interface ID
Wireless_Node_1	11.1.1.0	11.1.1.2	255.255.255.255	1	1
Wireless_Node_2	11.1.1.0	11.1.1.3	255.255.255.255	1	1
Wireless_Node_3	11.1.1.0	11.1.1.4	255.255.255.255	1	1
Wireless_Node_ $i + 1$	11.1.1.0	11.1.1.( $i+1$ )	255.255.255.255	1	1
Wireless_Node_49	11.1.1.0	11.1.1.50	255.255.255.255	1	1

Table 12: Static route configured in Case 2. Here I vary from 3 to 4



## Results

The average and aggregate application throughputs are noted down from the NetSim Results window and tabulated in Table 9 of the main document.

Steps to calculate the average and aggregate throughput:

1. Run the simulation for cases With RTS- CTS and Without RTS-CTS.
2. After the simulation note down the application throughputs (for each application), copy the values into an Excel sheet.
3. Calculate the Average and Aggregate throughputs for each case and tabulate the data.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	9.9	4.72	4.47	8.92	5.75	4.61	4.47	4.73	5.60	5.40	5.04	4.95	5.50	5.37	5.08	5.09	5.09	5.15	5.11
		5.37																	
			3.46	0.34	0.97	2.41	2.11	1.85	1.41	1.26	1.66	1.77	1.46	1.45	1.65	1.61	1.63	1.58	1.62
				8.98	2.85	1.65	2.28	1.90	1.29	1.01	0.78	1.12	1.00	0.89	0.92	0.98	1.00	1.02	0.96
					5.96	2.59	1.66	2.56	3.83	2.82	2.16	1.86	3.14	2.80	2.38	2.32	2.26	2.34	2.34
						5.52	2.62	1.69	2.13	3.47	2.93	2.23	1.94	2.73	2.56	2.39	2.38	2.35	2.36
							4.65	2.28	1.17	1.72	3.06	2.54	1.74	1.71	2.44	2.36	2.20	2.13	2.16
								4.46	1.83	0.87	1.24	2.45	1.83	1.37	1.38	1.88	1.86	1.75	1.75
									5.47	2.09	0.93	1.46	3.32	2.34	1.68	1.62	1.83	1.91	1.78
										5.74	2.38	1.08	1.51	3.19	2.33	1.79	1.83	2.03	2.06
											5.73	2.31	0.94	1.44	3.05	2.42	1.91	1.89	2.16
												5.57	2.11	0.95	1.47	2.94	2.33	1.82	1.81
													5.84	2.19	1.01	1.39	2.60	2.20	1.68
														5.76	2.19	0.97	1.49	2.65	2.25
															5.67	2.27	1.05	1.55	2.76
																5.66	2.32	1.07	1.50
																	5.42	2.27	1.01
																		5.42	2.28
																			5.49
Average	9.90	5.05	3.42	4.71	3.73	3.28	2.89	2.71	2.76	2.70	2.60	2.49	2.51	2.48	2.43	2.39	2.33	2.31	2.29
Aggregate	9.90	10.10	10.27	18.86	18.67	19.65	20.21	21.72	24.82	26.97	28.64	29.87	32.64	34.72	36.42	38.25	39.69	41.58	43.58

Fig 14: Average and aggregate throughput calculation for each case with number of transmitting nodes increased from 1 to 49

## Plotting Average Throughput vs Number of Transmitting Nodes

The following steps were done to get the average throughput vs Number of transmitting nodes plot:

1. Go to Insert tab and click on chart option, select XY(Scatter) and select Scatter Plot from the options provided. Click on ok.

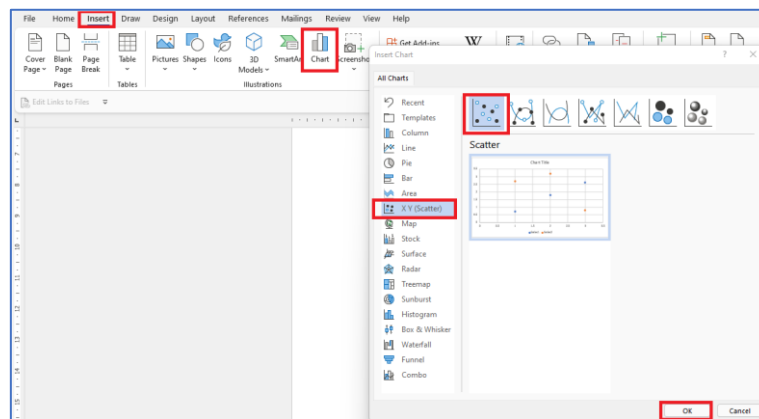


Fig 15: XY Scatter plot selection

2. In the data sheet, enter the Number of Transmitting nodes in the first column and average throughput obtained in the second column

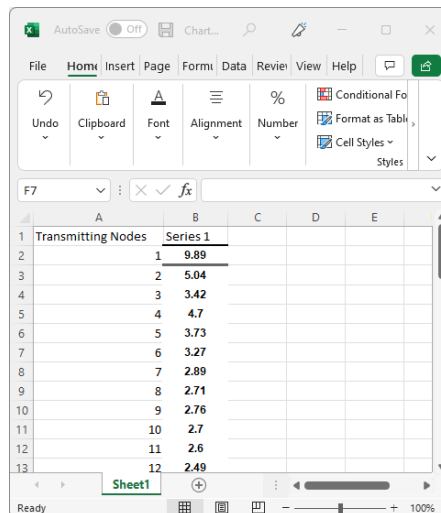


Fig 16: Plot data

- This will provide a scatter plot. Now edit the Axis Titles to Average Throughput in the Y axis and Number of Transmitting Nodes in the X axis
- Right click on the x axis and select format axis option. In the Bound section, set the Minimum field to 1.0 and the Maximum field to 50.0. In the Units section, set the Major field to 8.0 and Minor field to 2.0.

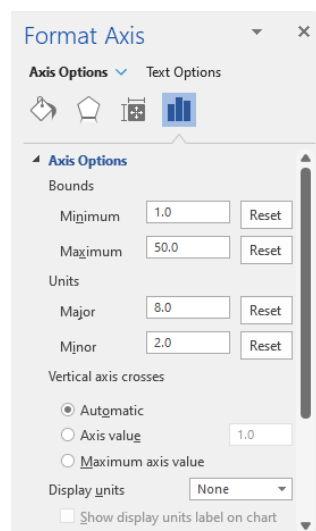


Fig 17: Format axis option

- Enable Trendline in the chart properties. This will provide an Average throughput vs Number of transmitting nodes plot as shown below

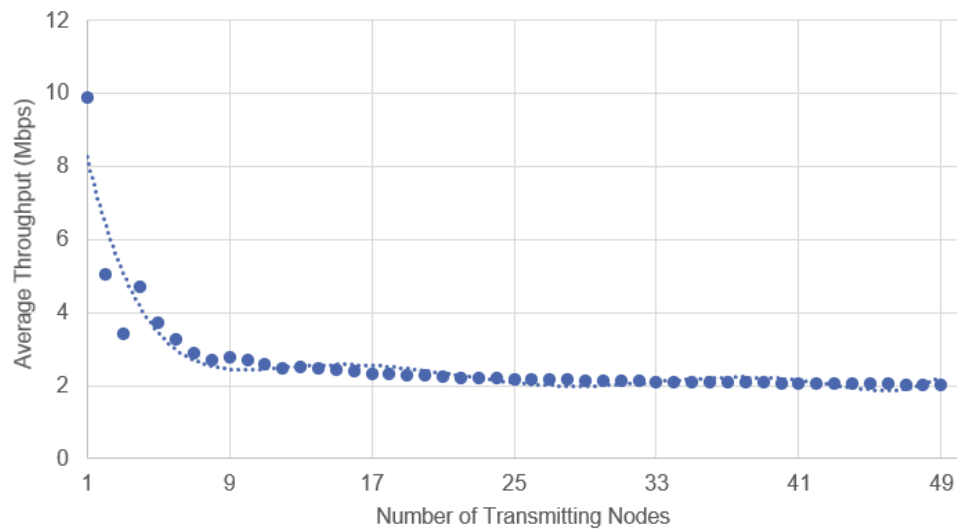


Fig 18: Average throughput vs transmitting nodes plot

## Border effects in line networks: Plotting Throughput vs Transmitting Node ID

The following steps were done to get the throughput vs transmitting node ID plot:

1. Go to Insert tab and click on chart option, select XY(Scatter) and select Scatter Plot from the options provided. Click on ok.

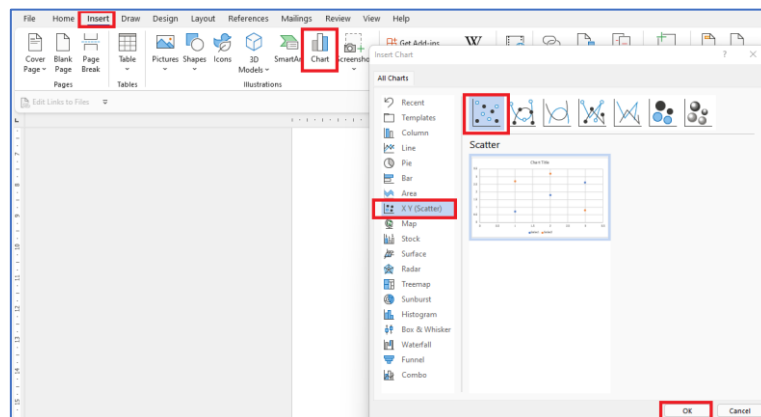


Fig 19: XY Scatter plot selection

2. In the data sheet, enter the transmitting node ID in the first column and the throughput per application obtained after simulation in the second column. Consider the 5 Transmitting Node scenario.

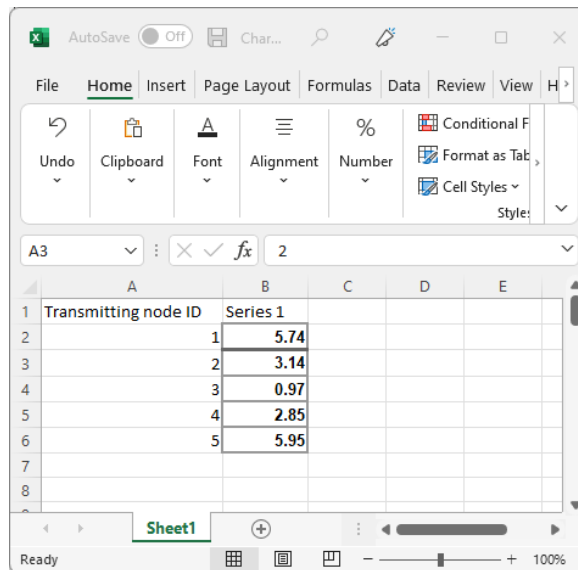


Fig 20: Plot data

3. This will provide a scatter plot. Now edit the Axis Titles to Throughput in the Y axis and Transmitting Node ID in the X axis
4. Right click on the x axis and select format axis option. In the Bound section, set the Minimum field to 1.0 and the Maximum field to 5.0. In the Units section, set the Major field to 1.0 and Minor field to 0.2.

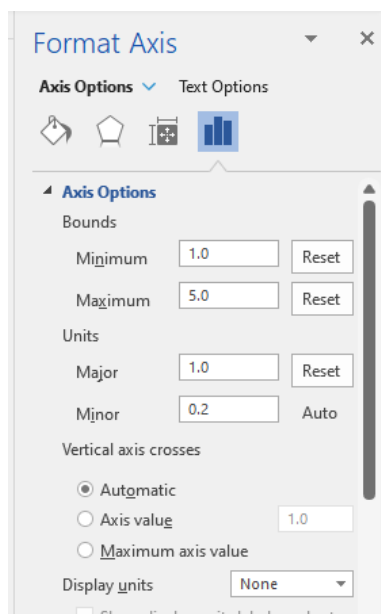


Fig 21: Format axis option

5. This will provide throughput vs Number of transmitting node ID plot as shown below.
6. Similarly plot for 24 Transmitting Nodes and 49 Transmitting Nodes scenarios.

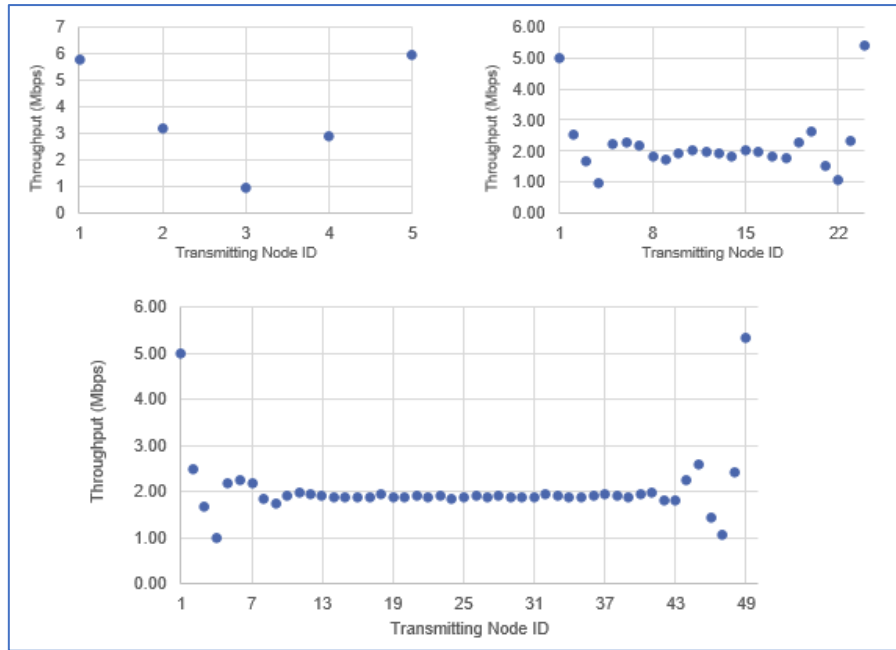


Fig 22: Throughput vs Transmitting Node ID plot for 5 transmitting nodes, 24 transmitting nodes and 49 transmitting nodes scenarios.

## Case 3: Ad Hoc Lattice, Single Hop Communication, Two Hop CS Range: Horizontal Traffic

### Procedure:

In this section we explain how users can run the simulation and obtain results that match those provided in the main document

1. Use the following download Link to download a compressed zip folder which contains the workspace:[https://github.com/NetSim-TETCOS/Performance-Analysis-Wireless-Adhoc-Network\\_v13.1/archive/refs/heads/main.zip](https://github.com/NetSim-TETCOS/Performance-Analysis-Wireless-Adhoc-Network_v13.1/archive/refs/heads/main.zip)
2. Extract the zip folder.
3. The extracted project folder consists of a NetSim workspace file (WiFi\_Adhoc\_Performance\_v13\_1.netsimexp).
4. Go to NetSim Home window, go to Your Work and click on Import.

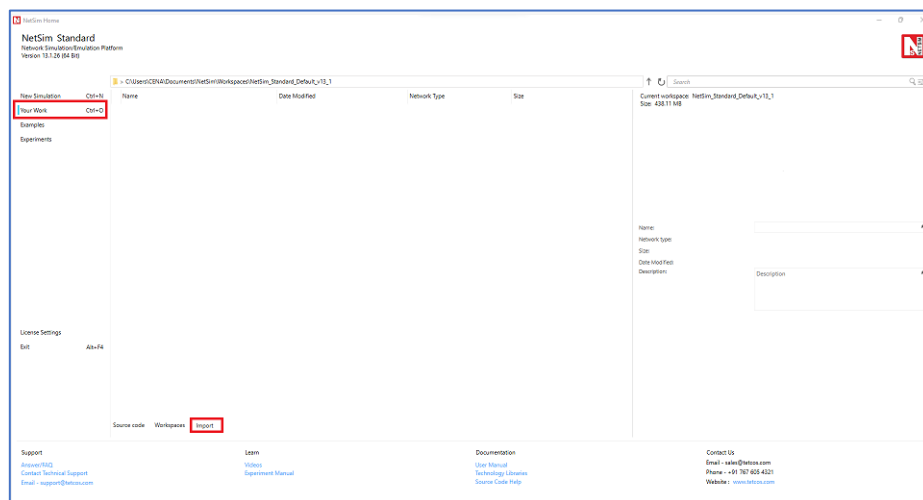


Fig 23: NetSim Home Window

5. In the Import Workspace Window, browse and select the WiFi\_Adhoc\_Performance\_v13\_1.netsimexp file from the extracted directory. Click on create a new workspace option and browse to select a path in your system where you want to set up the workspace folder.
6. Choose a suitable name for the workspace of your choice. Click Import.

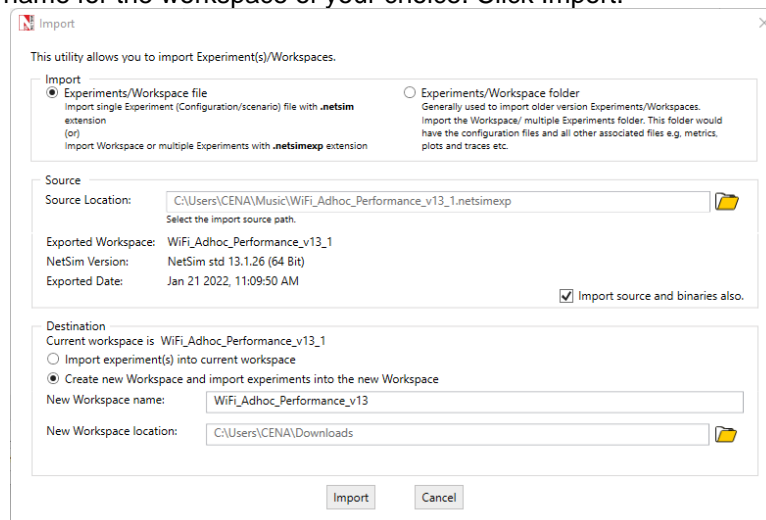


Fig 24: NetSim Import workspace window

7. The Imported Project workspace will automatically be set as the current workspace.

8. The list of experiments is now loaded onto the selected workspace.
9. The Experiments/ configuration files associated with the case mentioned above are available in the folder“Case\_3\_Adhoc\_Lattice\_Single\_Hop\_Communication\_Two\_Hop\_CS\_Range\_Horizontal\_Traffic” inside the workspace” inside the workspace.

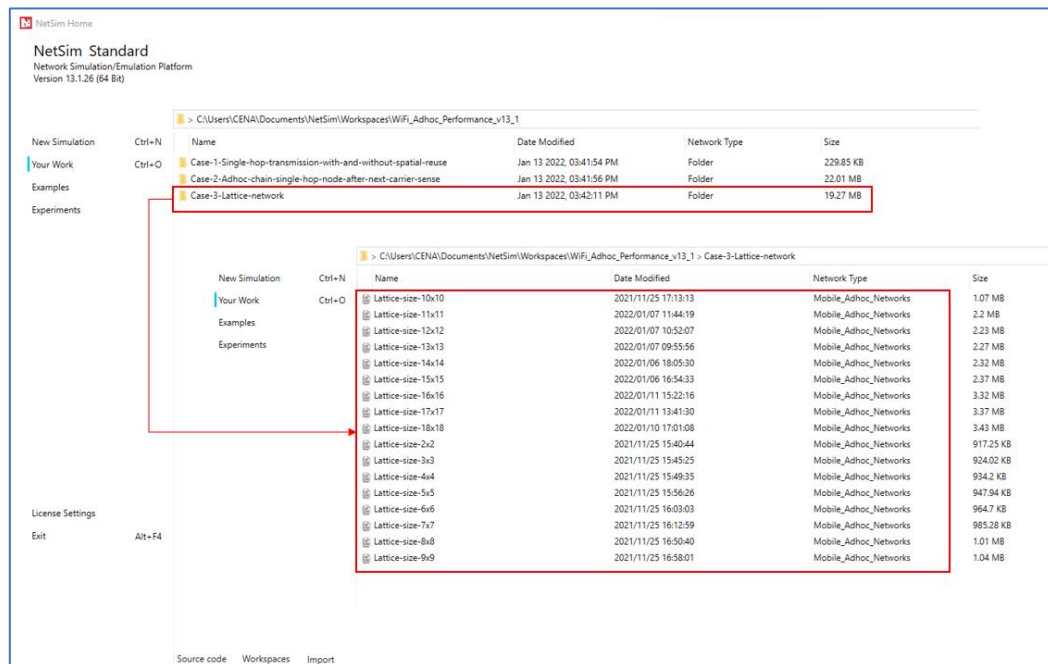


Fig 25: NetSim Your work window with experiments for Case 3- Adhoc Lattice, single hop communication two hop CS Range- Horizontal Traffic

## Network Scenario

NetSim UI displays the following Network Scenarios.



Fig 26: Network scenario with  $10 \times 10$  MANET nodes. The nearest neighbour distance is 165m. The data transmissions are purely horizontal.

## Device Coordinates

The distance setting should be such that the neighbour node is 165m away.  
The devices were placed in the following coordinates for each of the cases shown above

Node 1	Node 2	Node 3	Node 4	Node 5	Node <sub>i+1</sub>	Node 324
(0,0)	(165,0)	(330,0)	(495,0)	(660,0)	(x+165, y+165)	(2805,2805 )

Table 13: X and Y coordinates of devices in the network scenario. The devices are placed 165m away from each other.

## Simulation Parameters

10. The Grid length was set to 8500 × 8500.

11. The simulation Parameters were configured as follows:

802.11 Parameters	
Standard	IEEE802.11g
Operating Frequency	2.4GHz
Rate Adaptation	False
Dot11_RTS Threshold	3000B and 1000B
Frequency Band	2.4GHz
Bandwidth	20MHz
Transmitter Power	100mW
Medium Access Control	DCF

Table 14: Values set for different parameters in simulation

Wireless Link Parameters	
Channel Characteristics	Path Loss Only
Path Loss Model	Log Distance
Pathloss Exponent ( $\eta$ )	2.6
Simulation Duration	
Simulation Time	100 seconds

Table 15: Wireless Link parameter and Simulation time

Application Parameters	
Application Type	CBR
Source ID	Node <sub>i</sub> where i= 1, 2,...etc 49
Destination ID	Node <sub>i+1</sub> where i= 1, 2..etc 49
Packet Size	1460 bytes
Packet Size Distribution	Constant
Inter-Arrival Time	973.33 microseconds (12 Mbps Application Rate. Saturated queues) and 1460 microseconds (for larger lattices)
Transport Protocol	UDP

Table 16: Application properties set in this experiment

Protocol Parameters	Value
CS Threshold	-85 dBm
SIFS	10 $\mu$ s
Slot Time	9 $\mu$ s
DIFS (SIFS + 2 × Slot Time)	28 $\mu$ s
Preamble Time	20 $\mu$ s
MPDU Size	1528 B

Table 17: 802.11 g protocol parameters. These values are set in the code

## Static Route configuration

Static routes were configured in each device so that the flow of data packets will be transmitted from Node i to Node i+1

To set the static routes, go to Wireless\_Node properties > Network Layer > Enable Static Route IP.

Device	Network Destination	Gateway	Subnet Mask	Metrics	Interface ID
Wireless_Node_1	11.1.1.2	11.1.1.2	255.255.255.255	1	1
Wireless_Node_2	11.1.1.3	11.1.1.3	255.255.255.255	1	1
Wireless_Node_3	11.1.1.4	11.1.1.4	255.255.255.255	1	1
Wireless_Node <sub>i</sub> + 1	11.1.1.(i+1)	11.1.1.(i+1)	255.255.255.255	1	1

Table 18: Static route configured for Case 3.



## Results

The average and aggregate application throughput is noted down from the NetSim Results window and tabulated in Table 10 of the main document.

Steps to calculate the average and aggregate throughput are as follows:

1. Run the simulation for cases with different lattice sizes.
2. After the simulation note down the application throughputs (for each application), copy the values into an Excel sheet.
3. Calculate the Average and Aggregate throughputs for each case and tabulate the data.

	2X2	3X3	4X4	5X5
	4.70	1.72	2.35	2.72
	4.69	2.32	2.12	0.82
		0.79	0.57	0.89
		1.60	0.15	0.38
		1.72	0.56	0.88
		2.25	0.17	0.19
			3.10	1.00
			0.37	0.40
			0.38	0.10
			2.39	0.92
			2.10	0.37
			3.08	0.43
				2.22
				1.15
				0.69
				1.13
				2.73
				0.81
				0.98
				2.20
Average	4.70	1.73	1.44	1.05
Aggregate	9.40	10.40	17.33	20.99

Fig 27: Average and aggregate throughput calculation for each case with lattice size varying from  $2 \times 2$  to  $18 \times 18$

## Plotting Average Throughput vs Number of Transmitting Nodes

The following steps were done to get the average throughput vs Number of transmitting nodes plot:

6. Go to Insert tab and click on chart option, select XY(Scatter) and select Scatter Plot from the options provided. Click on ok

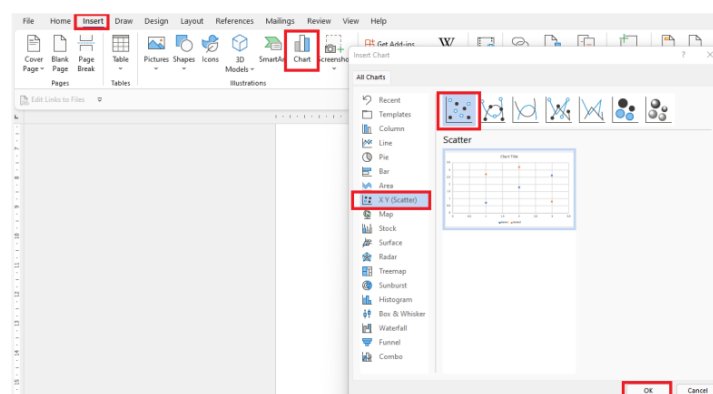


Fig 28: XY Scatter plot selection

7. In the data sheet, enter the Number of Transmitting nodes in the first column (x values) and average throughput obtained in the second column (Y values).

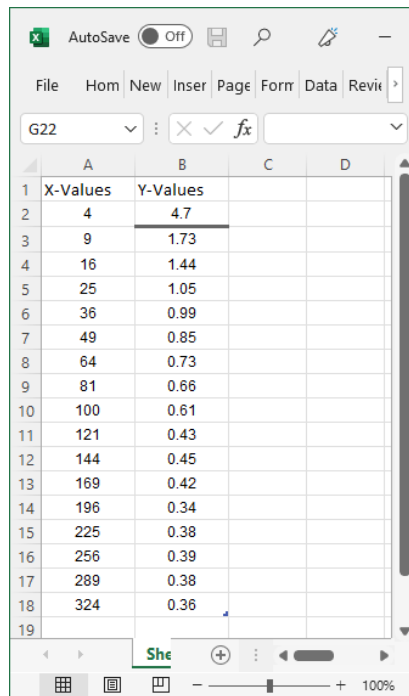


Fig 29: Plot data

8. This will provide a scatter plot. Now edit the Axis Titles to Average Throughput in the Y axis and Number of Transmitting Nodes in the X axis
9. Right click on the x axis and select format axis option. In the Bound section, set the Minimum field to 0 and the Maximum field to 350.0. In the Units section, set the Major field to 50.0 and Minor field to 10.0.

**Format Axis**

Axis Options Text Options

**Axis Options**

Bounds

Minimum 0.0 Auto

Maximum 350.0 Auto

Units

Major 50.0 Auto

Minor 10.0 Auto

Vertical axis crosses

☒ Automatic

☐ Axis value 0.0

☐ Maximum axis value

Display units None

☐ Show display units label on chart

Fig 30: Format axis option

10. Enable Trendline in the chart properties. This will provide an Average throughput vs Number of transmitting nodes plot as shown below

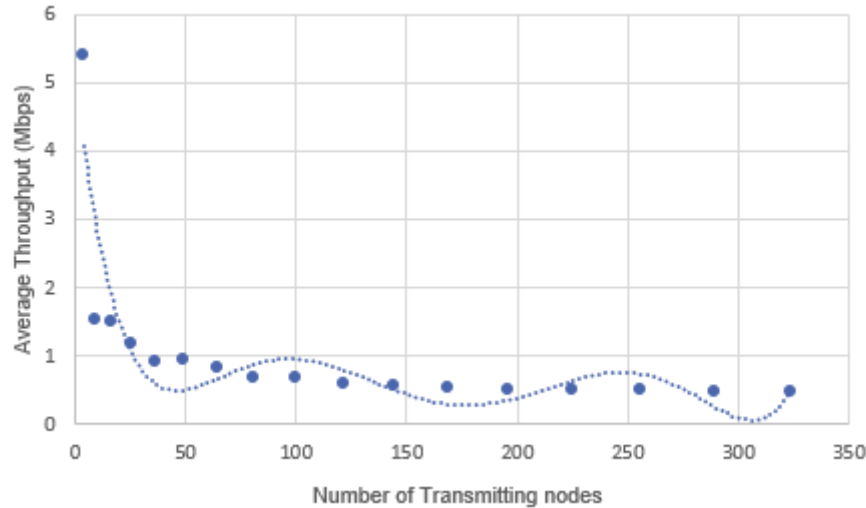


Fig 31: Average throughput vs transmitting nodes plot

### Case 4: 4 Hop Network

In this section we explain how users can run the simulation and obtain results that match those provided in the main document.

Users have two options for running these simulations.

- Using the NetSim GUI. In this case the simulation will need to open one-by-one in NetSim and then run. The process is time consuming but is useful initially to get an understanding of the network the parameters configured and the results
- Using the Multi-parameter sweeper utility. With this user can sweep one or more parameters, change their values between simulation runs, and compare and analyse the performance metrics from each run. The entire process changing the values, running simulations, and collating results is done automatically by this utility.

The first sub section below explains how users can run simulations via the GUI while the next explains how to use the multi-parameter sweeper.

### Using NetSim's GUI. Importing the project into NetSim.

1. Download a compressed zip folder which contains the NetSim workspace from [https://github.com/NetSim-TETCOS/Performance-Analysis-Wireless-Adhoc-Network\\_v13.1/archive/refs/heads/main.zip](https://github.com/NetSim-TETCOS/Performance-Analysis-Wireless-Adhoc-Network_v13.1/archive/refs/heads/main.zip)
2. The extracted project folder would have a NetSim workspace file Performance-Analysis-Wireless-Adhoc-network.netsimexp.
3. Go to NetSim Home window, go to Your Work and click on Import.

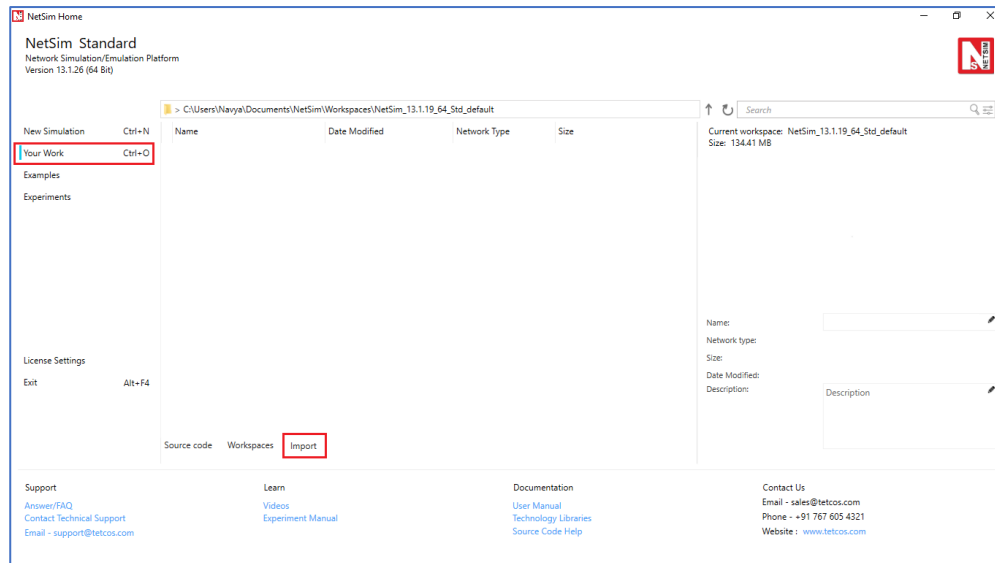


Fig 32:NetSim Home page window

4. In the Import Workspace Window, browse and select the Performance-Analysis-Wireless-Adhoc-network.netsimexp file from the extracted directory. Click on create a new workspace option. Browse to select a path where you want to have the workspace folder.
5. Choose a suitable name for the workspace. Click Import

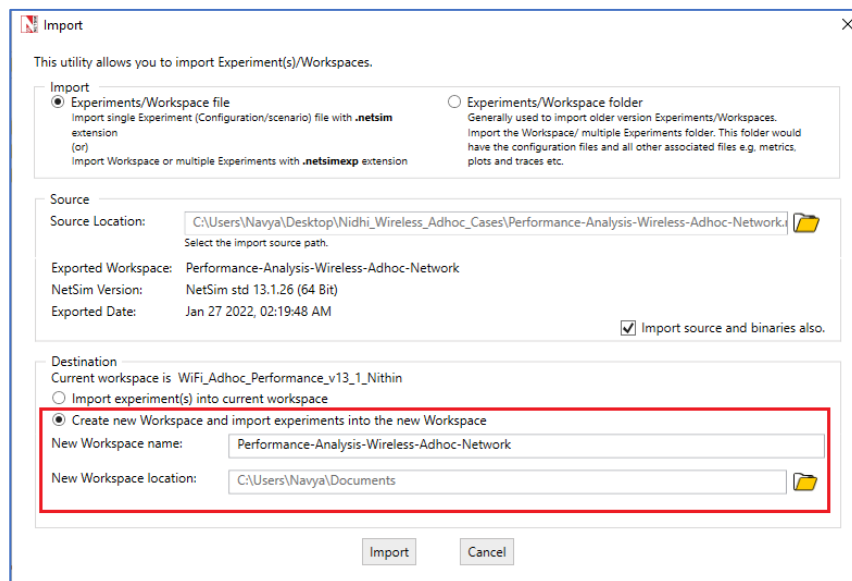


Fig 33: Importing workspace into new workspace

6. The Imported Project workspace will automatically be set as the current workspace. The list of experiments is now loaded onto the selected workspace.
7. The screen shot below shows the folder/file organization for 4-hop, 5-hop, 6-hop and 7-hop network scenarios.

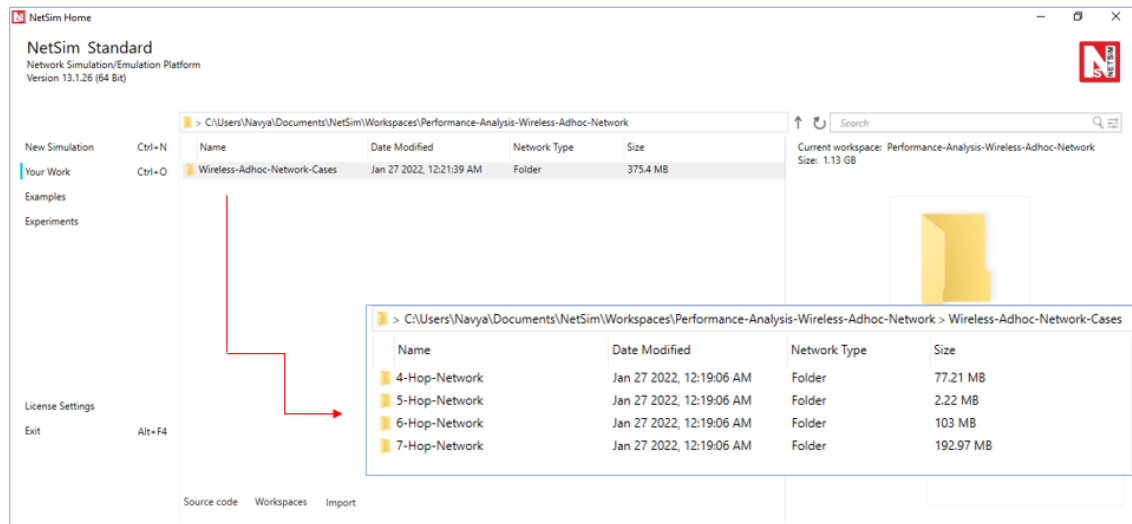


Fig 34: Folder with 4-Hop, 5- Hop, 6-Hop and 7-Hop (With RTS-CTS) examples

## Simulation Parameters

1. Grid length of 1500m x 1500m

Devices	X	Y
Wireless_Node_1	0	300
Wireless_Node_2	300	300
Wireless_Node_3	600	300
Wireless_Node_4	900	300
Wireless_Node_5	1200	300

Table 19: X and Y coordinates of devices in the network scenario

2. The designed in NetSim's MANET library. It comprises of Wireless nodes connected via an Adhoc link.

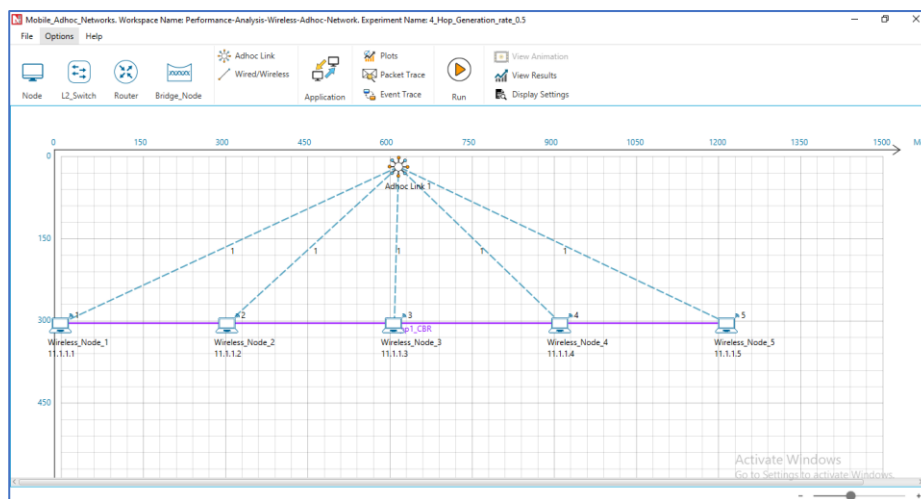


Fig 35: Network Scenario for 4-Hop Network designed with 5 wireless nodes

3. Device properties are set as shown below

802.11 properties	
Physical layer	
Standard	802.11b
Operating frequency	2.4GHz
Transmitter Power	100mW
Transmission type	DSSS
Power Source	Main
Datalink layer	
Rate Adaptation	False
Media Access protocol	DCF
Dot11_RTS Threshold	3000bytes

Table 20: Wireless Node 802.11 parameters

General Properties	
Mobility Model	No Mobility

Table 21: Wireless general properties

Wireless link Properties	
Channel Characteristics	Path Loss Only
Path Loss Model	Log Distance
Pathloss Exponent ( $\eta$ )	2.6

Table 22: Wireless Link Parameters

Application properties	
Application type	CBR
Source ID	Wireless_Node_1
Destination ID	Wireless_Node_5
Transport layer protocol	UDP
Packet Size	1000 bytes
Generation Rate/ Inter arrival time	Varied as shown in Table 24

Table 23: Application properties

The interpacket arrival time is computed as shown below

$$Inter(packet)ArrivalTime (\mu s) = \frac{PacketSize \times 8}{GenerationRate (Mbps)}$$

## Static Route Configuration

Static routes (whereby  $N_i$  always transmits to  $N_{(i+1)}$ ) are set to ensure single hop transmission. Thereby each node transmits data to the next-hop node according to the topology

To set Static routes for wireless nodes in order to transmit the data from Wireless\_Node\_1>Wireless\_Node\_2>Wireless\_Node\_3>Wireless\_Node\_4>Wireless\_Node\_5

Go to Wireless Nodes > Network layer >Enable Static IP Route and configure as below

**Wireless\_Node\_1:**

Fig 36: Static Route Configuration for Source node

Click on Add and then OK to set static routes in wireless node

Similarly configure for static routes for the remaining wireless nodes

Device	Network Destination	Gateway	Subnet Mask	Metrics	Interface ID
Wireless_node_2	11.1.1.5	11.1.1.3	255.255.255.255	1	1
Wireless_node_3	11.1.1.5	11.1.1.4	255.255.255.255	1	1
Wireless_node_4	11.1.1.5	11.1.1.5	255.255.255.255	1	1

Run the simulation for 100 sec and tabulate the throughput

## Numerical Results

The following data is used to generate the plot.

Generation Rate (Mbps)	IAT ( $\mu$ s)	Throughput (Mbps)
0.05	160000	0.050021
0.1	80000	0.100042
0.15	53333	0.149979
0.2	40000	0.2
0.25	32000	0.250021
0.3	26666	0.299958
0.35	22857	0.349979
0.4	20000	0.399916
0.45	17777	0.449937
0.5	16000	0.499958
0.51	15686	0.509979
0.52	15384	0.519916
0.53	15094	0.492968
0.54	14814	0.471832
0.55	14545	0.460042
0.56	14285	0.447326
0.57	14035	0.434863
0.58	13793	0.420126
0.59	13559	0.420968
0.6	13333	0.416337
0.65	12307	0.410189
0.7	11428	0.404379

0.75	10666	0.3952
0.8	10000	0.3872
0.85	9411	0.379453
0.9	8888	0.376168
0.95	8421	0.372968
1	8000	0.378442

Table 24:Numerical results for 4-Hop Network

## Automated simulation using multi-parameter sweeper

1. Here we run the samples for the same example mentioned above using the multi parameter sweeper
2. We use a base (first) network configuration file with packet size as 1000 bytes and Inter arrival time as 160000  $\mu$ s. This gives a traffic generation rate of 0.05 Mbps.
3. The Multi parameter has been configured to run the multiple scenarios (one after the other sequentially) with Inter-arrival times varying in the steps of 0.05 Mbps.

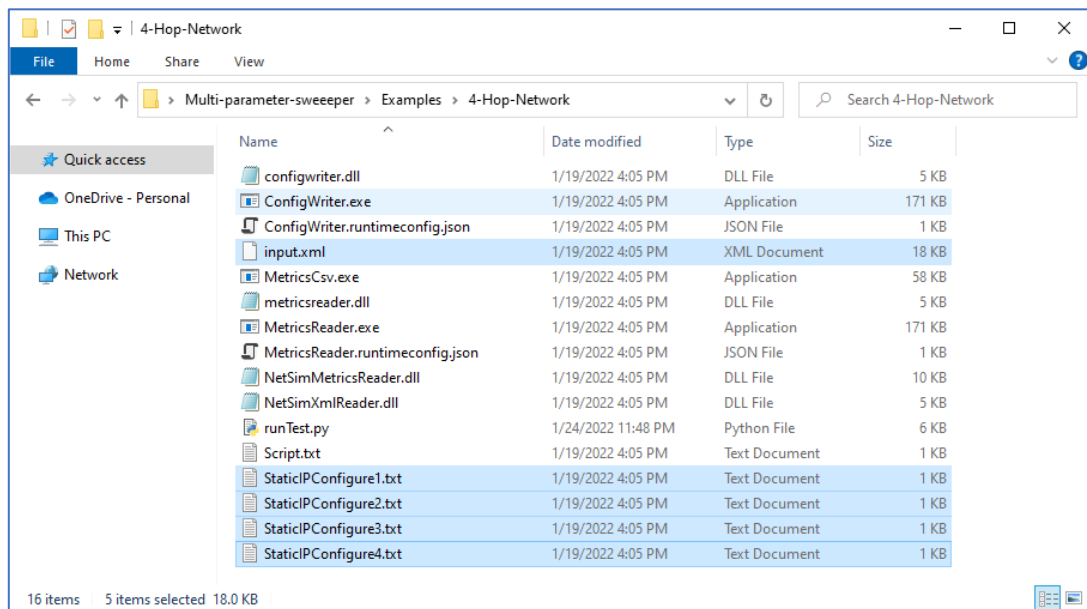


Fig 37:Multi Parameter sweeper folder consists of input.xml and several binaries

## Changes done to the config file to run the Multi-Parameter sweeper

1. Users can skip this sub section if they only wish to run simulations. In this sub section we explain the changes done to run the sweeper. It would be beneficial to users who wish to run the sweeper for other scenarios.
2. The Configuration.netsim and Static Route Configuration text files are copied from the saved experiment folder to Multi-parameter sweeper folder. Configuration.netsim is renamed as input.xml
3. The value of Inter\_Arrival\_time has been modified during each simulation is updated to {0} in the Configuration.netsim as shown below



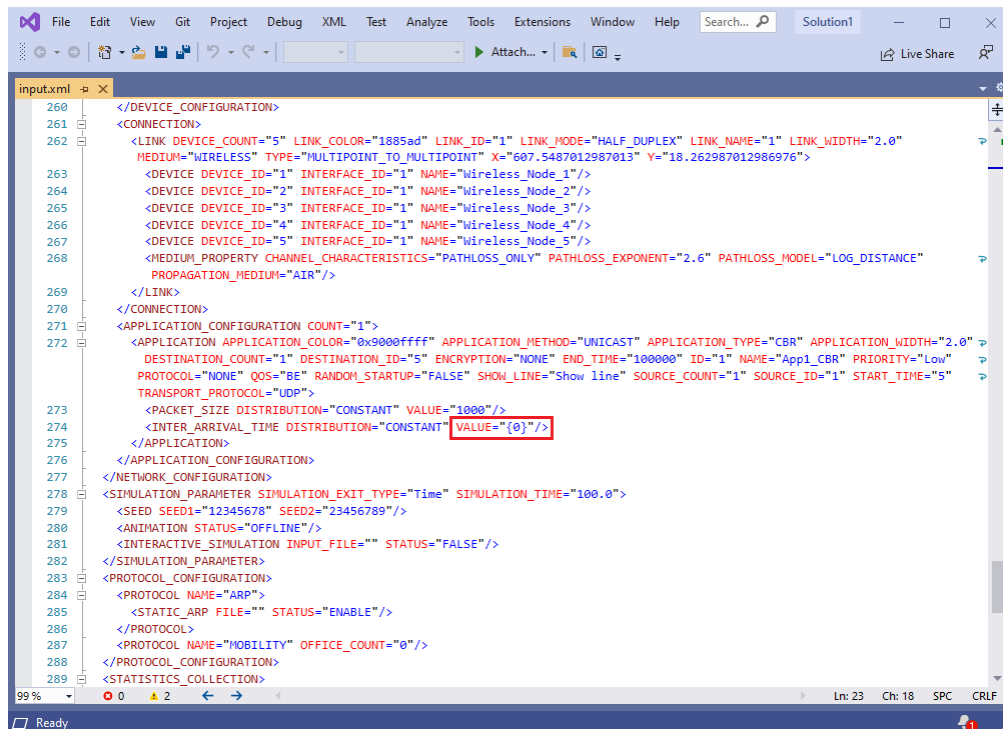


Fig 38: Modified Inter arrival time value in input.xml

4. The MP sweeper collates the results of the different simulation runs into a single XL file. How is this done? The Script.txt file (available in MP sweeper folder) is updated with the details of the output parameter to be read from the Metrics.xml file. After each run it is this output parameter's value that is added to the result csv log file.
5. In this example it the Application Throughput that we wish to log for each simulation run.
6. runTest.py is updated to (i) pass different inter-arrival values during each iteration (ii) generate configuration files (iii) run simulation and (iv) update the result csv log.

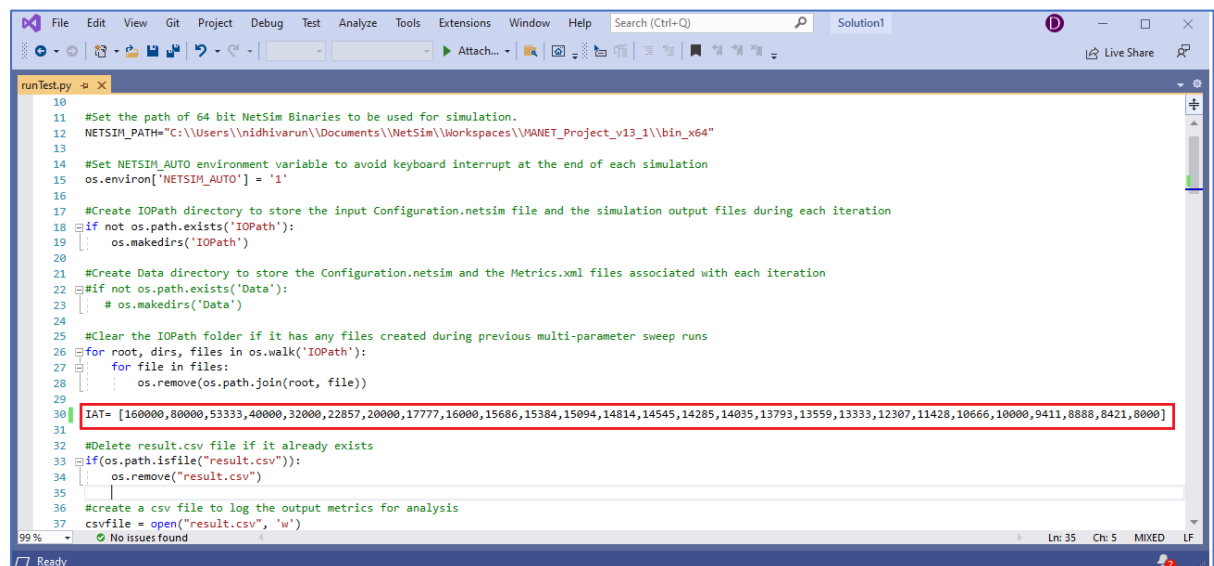


Fig 39: Varied Inter arrival time in runTest.py

7. NetSim\_Path variable is set to the workspace directory and to this bin\_x64 is appended. An example is shown in the screen shot below.



```

82 strIOPATH=os.getcwd()+"\\IOPATH"
83
84 #Run NetSim via CLI mode by passing the apppath iopath and license information to the NetSimCore.exe
85 cmd=NETSIM_PATH+ "\" + "NetSimcore.exe -appath \"\"+NETSIM_PATH+\"\" -iopath \"\"+strIOPATH+\"
86 \"\" -license \"\"+\"C:\\Program Files\\NetSim\\Standard_v13_1\\bin\\\"
87
88 #print(cmd)
89 os.system(cmd)
90
91
92 #Create a copy of the output Metrics.xml file for writing the result log
93 if(os.path.isfile("IOPATH\\Metrics.xml")):
94     shutil.copy("IOPATH\\Metrics.xml", "Metrics.xml")
95
96 cmd="MetricsCsv.exe IOPATH"
97 os.system(cmd)
98
99 #Number of Script files i.e Number of Output parameters to be read from Metrics.xml
100 #If only one output parameter is to be read only one Script text file with name Script.txt to be provided
101 #If more than one output parameter is to be read, multiple Script text file with name Script1.txt, Script2.txt,...
102 #...,Scriptn.txt to be provided
103 OUTPUT_PARAM_COUNT=1;
104
105 if(os.path.isfile("Metrics.xml")):
106     #Write the value of the variable parameters in the current iteration to the result log
107     csvfile = open("result.csv", 'a')
108     csvfile.write('\\n'+str(i)+'\\n')
109     csvfile.close()
110
111

```

Fig 42: License path is set based on node locked evaluation license / cloud license/ server node locked/dongle based floating license

- The sweeper process is started by opening command prompt in the directory of the Multi-Parameter-Sweeper project and starting the python script as shown below

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19043.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Navya\Desktop\Multi-Parameter-Sweeper>python runTest.py

```

Fig 43: Running python script using cmd prompt

- Then the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Inter Arrival time.

## Results

Upon completing all the simulations, the Multi-Parameter-Sweeping folder will have the following file and folders created as shown below

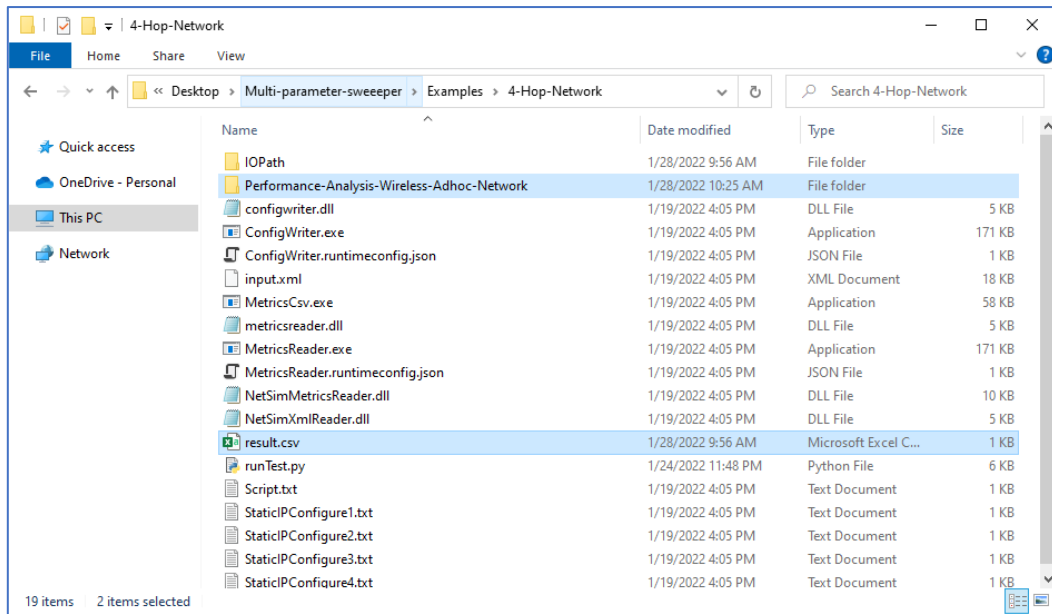


Fig 44: After Simulation Multi-Parameter-Sweeping folder contains output files like result.scv, Folders consists of related network cases

**Performance-Analysis-Wireless-Adhoc-Network:** Contains multiple folders corresponding to each simulation run. The file name has the value of the parameter used in that iteration

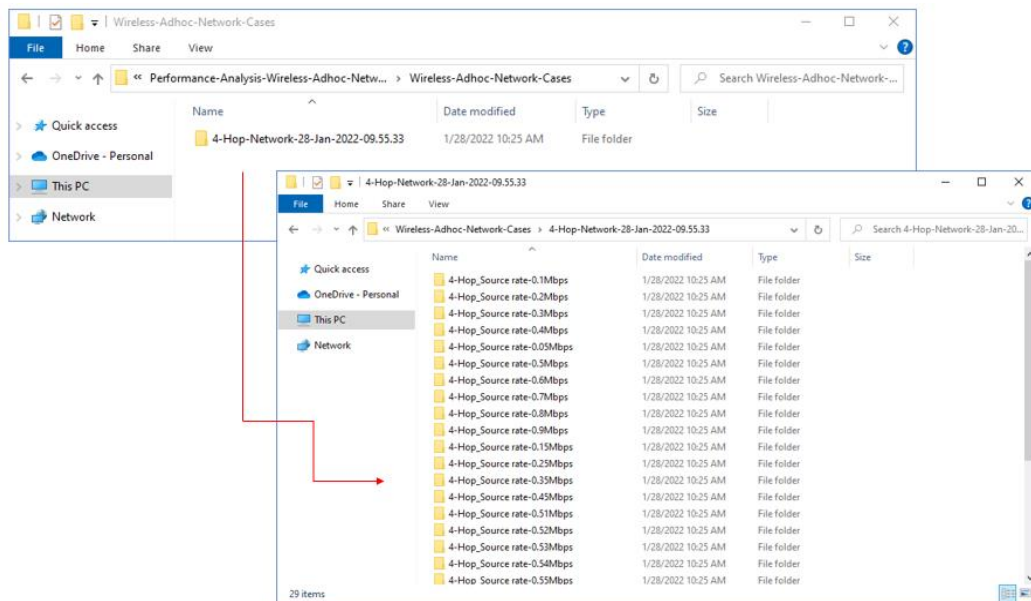


Fig 45:Based on Source rate/Different Inter arrival time Configuration files are created  
Each folder contains the all the output files associated with each simulation run

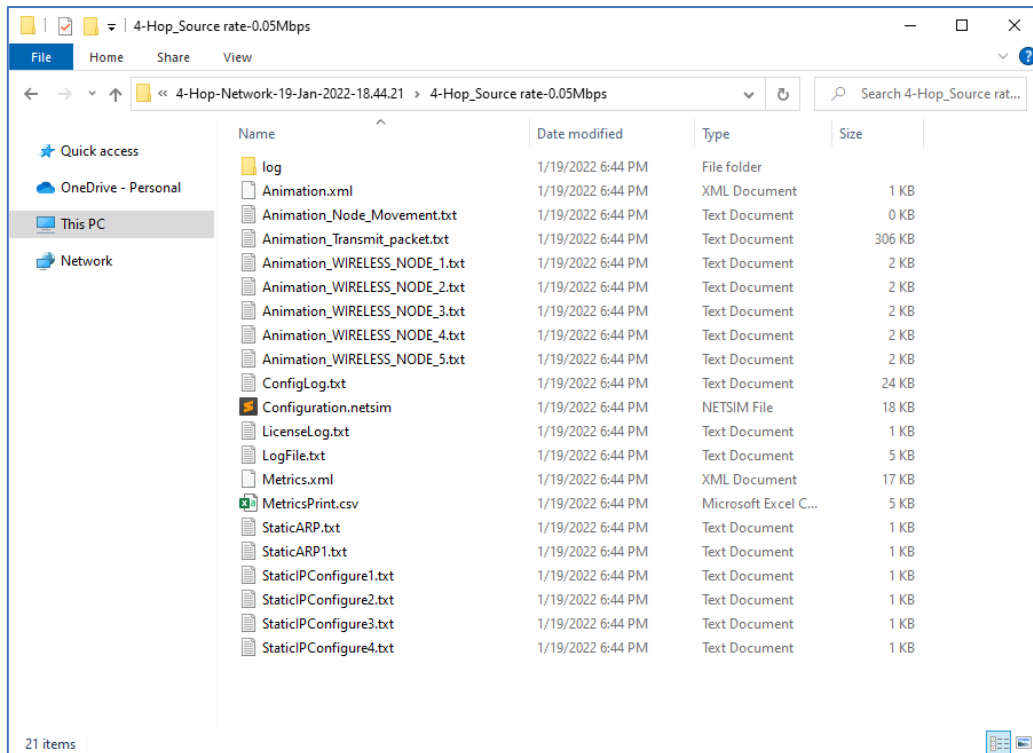


Fig 46: Each folder contains the all the output files

**Result.csv:** This is the output log which contains the Inter Arrival time varied during each simulation run and the corresponding throughput after each run.

	A	B	C
	INTER_ARRIVAL_TIME(micro sec)	THROUGHPUT(Mbps)	
1			
2	160000	0.050021	
3	80000	0.100042	
4	53333	0.149979	
5	40000	0.2	
6	32000	0.250021	
7	26666	0.299958	
8	22857	0.349979	
9	20000	0.399916	
10	17777	0.449937	
11	16000	0.499958	
12	15686	0.509979	
13	15384	0.519916	
14	15094	0.492968	
15	14814	0.471832	
16	14545	0.460042	
17	14285	0.447326	
18	14035	0.434863	
19	13793	0.420126	
20	13559	0.420968	
21	13333	0.416337	

Fig 47:Throughput (Mbps) obtained for different value of IAT in result.csv

## Case 5: 5 Hop Network

1. In the imported workspace, go to 5-Hop Network folder to display the related 5-Hop cases.
2. This designed in NetSim's MANET library. It comprises of 6 Wireless nodes connected via an Adhoc link and the Application is created between Source ID Wireless\_Node\_1 and destination ID Wireless\_Node\_6.

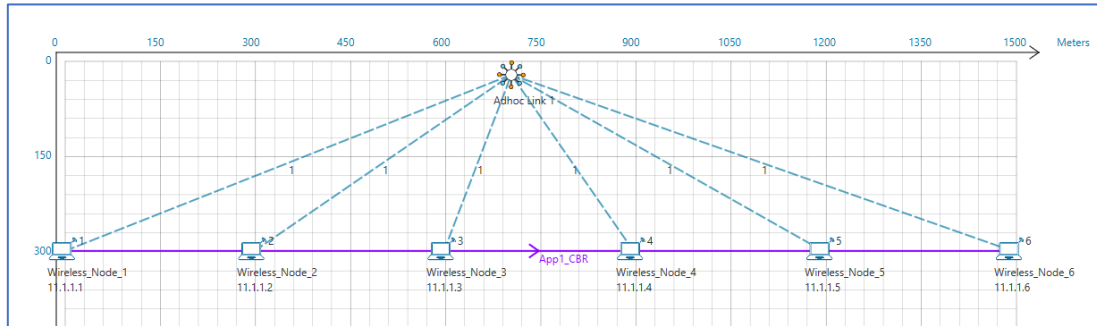


Fig 48: Network Scenario for 5-Hop Network designed with 6 wireless nodes

3. The Simulation parameters setting remains the same as previous case

## Additional Setting

4. Grid length of 1500m x 1500m

Devices	X	Y
Wireless_Node_1	0	300
Wireless_Node_2	300	300
Wireless_Node_3	600	300
Wireless_Node_4	900	300
Wireless_Node_5	1200	300
Wireless_Node_6	1500	300

Table 25: X and Y Co- ordinates of devices for 5-Hop Network

5. Static route should be carefully configured to ensure the data flow from Wireless\_Node\_1>Wireless\_Node\_2> Wireless\_Node\_3> Wireless\_Node\_4> Wireless\_Node\_5> Wireless\_Node\_6

## Numerical Results

The following data is used to generate the plot.

Generation Rate (Mbps)	IAT (μs)	Throughput (Mbps)
0.05	160000	0.050021
0.1	80000	0.100042
0.15	53333	0.149979
0.2	40000	0.2
0.25	32000	0.249937
0.3	26666	0.299958
0.35	22857	0.349979
0.4	20000	0.399916
0.45	17777	0.449937
0.5	16000	0.499874

0.51	15686	0.509895
0.52	15384	0.519916
0.53	15094	0.483705
0.54	14814	0.463411
0.55	14545	0.450442
0.56	14285	0.433179
0.57	14035	0.421305
0.58	13793	0.406063
0.59	13559	0.392084
0.6	13333	0.376168
0.65	12307	0.357811
0.7	11428	0.346779
0.75	10666	0.345684
0.8	10000	0.332379
0.85	9411	0.334905
0.9	8888	0.327663
0.95	8421	0.329937
1	8000	0.331705

Table 26: Numerical results for 5-Hop Network

## Changes done to the config file to run the multi-Parameter sweeper

1. The Configuration file saved with above setting is replaced in the Multi parameter folder and Inter arrival time value mentioned as {0}
2. Number of nodes used in Network scenario is given to copy the Static Route Configuration text files to IO path same as Configuration.netsim

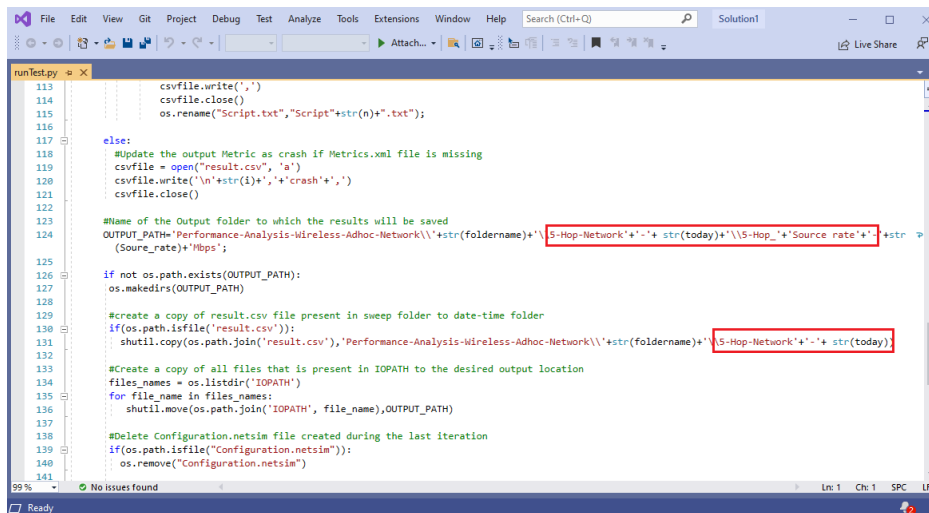
```

1 import subprocess
2 import shlex
3 import random
4 import shutil
5 import math
6 import sys
7 import datetime
8 import time
9 import os
10
11 #Set the path of 64 bit NetSim Binaries to be used for simulation.
12 NETSIM_PATH="C:\\Users\\Navya\\Documents\\NetSim\\Workspaces\\Performance-Analysis-Wireless-Adhoc-Network\\bin_x64"
13 NUMBER_OF_NODES=6
14
15 #Set NETSIM_AUTO environment variable to avoid keyboard interrupt at the end of each simulation
16 os.environ["NETSIM_AUTO"] = '1'
17
18 #Create IOPATH directory to store the input Configuration.netsim file and the simulation output files during each iteration
19 if not os.path.exists('IOPATH'):
20     os.makedirs('IOPATH')
21
22 #Create Data directory to store the Configuration.netsim and the Metrics.xml files associated with each iteration
23 if not os.path.exists('Data'):
24     os.makedirs('Data')
25
26 #Clear the IOPATH folder if it has any files created during previous multi-parameter sweep runs
27 for root, dirs, files in os.walk('IOPATH'):
28     for file in files:
29         os.remove(os.path.join(root, file))
30

```

Fig 49: Number of nodes in scenario is given as input to copy the Static route configuration files to IO path

3. If the user wishes to change the output folder name, it can be renamed as 5-Hop as in shown in



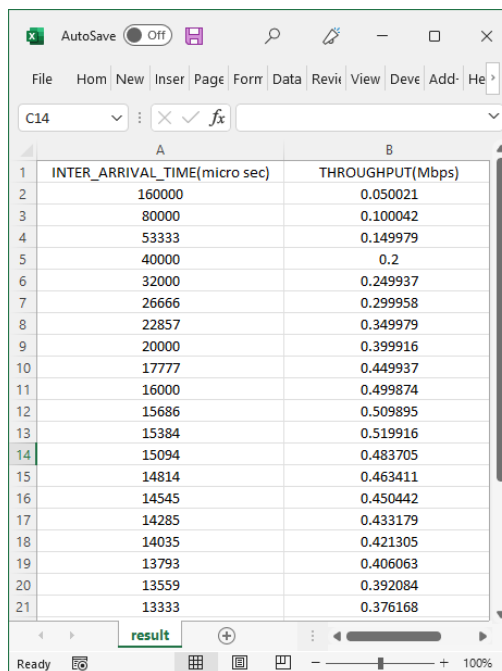
```

113     csvfile.write(',')
114     csvfile.close()
115     os.rename("Script.txt","Script"+str(n)+".txt");
116
117 else:
118     #Update the output Metric as crash if Metrics.xml file is missing
119     csvfile = open("result.csv", "a")
120     csvfile.write('\n'+str(i)+' '+'crash'+',')
121     csvfile.close()
122
123 #Name of the Output folder to which the results will be saved
124 OUTPUT_PATH="Performance-Analysis-Wireless-Adhoc-Network\\"+str(foldername)+"5-Hop-Network"+"-" + str(today)+"\5-Hop-"+"Source rate"+"-" + str
125 (Source_rate)+"\Mbps";
126
127 if not os.path.exists(OUTPUT_PATH):
128     os.makedirs(OUTPUT_PATH)
129
130 #create a copy of result.csv file present in sweep folder to date-time folder
131 if(os.path.isfile('result.csv')):
132     shutil.copy(os.path.join('result.csv'),'Performance-Analysis-Wireless-Adhoc-Network\\"+str(foldername)+"5-Hop-Network"+"-" + str(today)
133
134 #Create a copy of all files that is present in IOPATH to the desired output location
135 files_names = os.listdir('IOPATH')
136 for file_name in files_names:
137     shutil.move(os.path.join('IOPATH', file_name),OUTPUT_PATH)
138
139 #Delete Configuration.netsim file created during the last iteration
140 if(os.path.isfile("Configuration.netsim")):
141     os.remove("Configuration.netsim")

```

Fig 50:Renaming the output folder by Number of Hops

4. The sweeper process is started by opening command prompt in the directory of the Multi-Parameter-Sweeper project.
5. Then the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Inter Arrival time and provides the results in results.csv file as shown in below



	A	B
1	INTER_ARRIVAL_TIME(micro sec)	THROUGHPUT(Mbps)
2	160000	0.050021
3	80000	0.100042
4	53333	0.149979
5	40000	0.2
6	32000	0.249937
7	26666	0.299958
8	22857	0.349979
9	20000	0.399916
10	17777	0.449937
11	16000	0.499874
12	15686	0.509895
13	15384	0.519916
14	15094	0.483705
15	14814	0.463411
16	14545	0.450442
17	14285	0.433179
18	14035	0.421305
19	13793	0.406063
20	13559	0.392084
21	13333	0.376168

Fig 51: 5-Hop Network results

## Case 06: 6- Hop Network

1. In the imported workspace, go to 6-Hop Network folder to display the related 6-Hop cases.
2. This designed in NetSim's MANET library. It comprises of 7 Wireless nodes connected via an Adhoc link and the Application is created between Source ID Wireless\_Node\_1 and destination ID Wireless\_Node\_7.



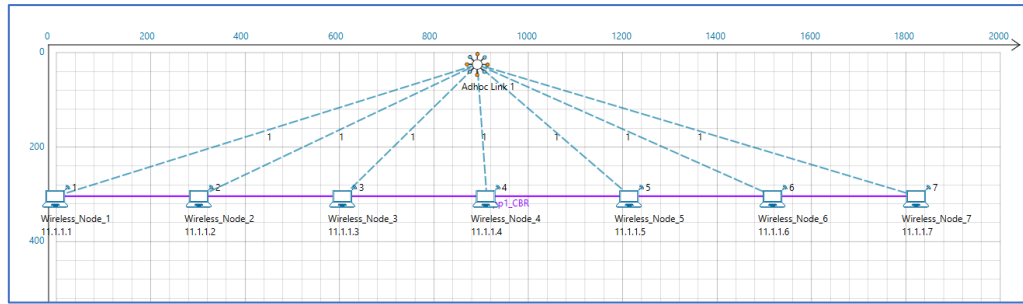


Fig 52: Network Scenario for 6-Hop Network designed with 7 wireless nodes

3. The Simulation parameters setting remains the same as previous case

## Additional Setting

4. Grid length is 2000m × 2000m, Place the devices 300m apart from one another.

Devices	X	Y
Wireless_Node_1	0	300
Wireless_Node_2	300	300
Wireless_Node_3	600	300
Wireless_Node_4	900	300
Wireless_Node_5	1200	300
Wireless_Node_6	1500	300
Wireless_Node_7	1800	300

Table 27: X and Y Co- ordinates of devices for 6-Hop Network

5. Static route should be carefully configured to ensure the data flow from Wireless\_Node\_1>Wireless\_Node\_2>Wireless\_Node\_3> Wireless\_Node\_4> Wireless\_Node\_5> Wireless\_Node\_6>Wireless\_Node\_7.
6. Run the simulation for 100 sec and tabulate the throughput.

## Numerical Results

The following data is used to generate the plot.

Generation Rate (Mbps)	IAT (μs)	Throughput (Mbps)
0.05	160000	0.050021
0.1	80000	0.100042
0.15	53333	0.149979
0.2	40000	0.2
0.25	32000	0.249937
0.3	26666	0.299958
0.35	22857	0.349895
0.4	20000	0.399916
0.45	17777	0.449937
0.5	16000	0.499874
0.51	15686	0.453642
0.52	15384	0.438232
0.53	15094	0.426442
0.54	14814	0.415579

0.55	14545	0.407411
0.56	14285	0.393347
0.57	14035	0.383242
0.58	13793	0.366905
0.59	13559	0.354442
0.6	13333	0.340884
0.65	12307	0.336926
0.7	11428	0.332884
0.75	10666	0.320084
0.8	10000	0.310063
0.85	9411	0.312084
0.9	8888	0.309389
0.95	8421	0.308716
1	8000	0.310484

Table 28: Numerical results for 6-Hop Network

## Changes done to the config file to run the multi-Parameter sweeper

1. The Configuration file saved with above setting is replaced in the Multi parameter folder and Inter arrival time value mentioned as {0}
2. Number of nodes used in Network scenario is given to copy the Static Route Configuration text files to IO path same as Configuration.netsim

```

1 import subprocess
2 import shlex
3 import random
4 import shutil
5 import math
6 import sys
7 import datetime
8 import time
9 import os
10
11 #Set the path of 64 bit NetSim Binaries to be used for simulation.
12 NETSIM_PATH="C:\\Users\\Navya\\Documents\\NetSim\\Workspaces\\Performance-Analysis-Wireless-Adhoc-Network\\bin_x64"
13 NUMBER_OF_NODES=7
14
15 #Set NETSIM_AUTO environment variable to avoid keyboard interrupt at the end of each simulation
16 os.environ["NETSIM_AUTO"] = '1'
17
18 #Create IOPATH directory to store the input Configuration.netsim file and the simulation output files during each iteration
19 if not os.path.exists('IOPATH'):
20     os.makedirs('IOPATH')
21
22 #Create Data directory to store the Configuration.netsim and the Metrics.xml files associated with each iteration
23 if not os.path.exists('Data'):
24     os.makedirs('Data')
25
26 #Clear the IOPATH folder if it has any files created during previous multi-parameter sweep runs
27 for root, dirs, files in os.walk('IOPATH'):
28     for file in files:
29         os.remove(os.path.join(root, file))
30
31

```

Fig 53: Number of nodes in scenario is given as input to copy the Static route configuration files to IO path

3. If the user wishes to change the output folder name, it can be renamed as 6-Hop.

```

113 csvfile.write(',')
114 csvfile.close()
115 os.rename("Script.txt", "Script"+str(n)+".txt");
116
117 else:
118     #Update the output Metric as crash if Metrics.xml file is missing
119     csvfile = open("result.csv", 'a')
120     csvfile.write('\n'+str(i)+'-'+str('crash')+',')
121     csvfile.close()
122
123 #Name of the Output folder to which the results will be saved
124 OUTPUT_PATH="Performance-Analysis-Wireless-Adhoc-Network\\"+str(foldername)+"6-Hop-Network"+str(today)+"6-Hop_"+str(Source_rate)+"Mbps";
125
126 if not os.path.exists(OUTPUT_PATH):
127     os.makedirs(OUTPUT_PATH)
128
129 #create a copy of result.csv file present in sweep folder to date-time folder
130 if os.path.isfile('result.csv'):
131     shutil.copy(os.path.join('result.csv'), "Performance-Analysis-Wireless-Adhoc-Network\\"+str(foldername)+"6-Hop-Network"+str(today))
132
133 #Create a copy of all files that is present in IOPATH to the desired output location
134 files_names = os.listdir('IOPATH')
135 for file_name in files_names:
136     shutil.move(os.path.join('IOPATH', file_name), OUTPUT_PATH)
137
138 #Delete Configuration.netsim file created during the last iteration
139 if os.path.isfile("Configuration.netsim"):
140     os.remove("Configuration.netsim")
141

```

Fig 54:Renaming the Output folder by Number of Hop count

4. The sweeper process is started by opening command prompt in the directory of the Multi-Parameter-Sweeper project.
5. Then the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Inter Arrival time and provides the results in results.csv file as shown in below

	A	B	C
	INTER_ARRIVAL_TIME(micro sec)	THROUGHPUT(Mbps)	
1			
2	160000	0.050021	
3	80000	0.100042	
4	53333	0.149979	
5	40000	0.2	
6	32000	0.249937	
7	26666	0.299958	
8	22857	0.349895	
9	20000	0.399916	
10	17777	0.449937	
11	16000	0.499874	
12	15686	0.453642	
13	15384	0.438232	
14	15094	0.426442	
15	14814	0.415579	
16	14545	0.407411	
17	14285	0.393347	
18	14035	0.383242	
19	13793	0.366905	
20	13550	0.354443	

Fig 55: 6-Hop Network results

## Case 07: 7-Hop Network

1. In the imported workspace, go to 7-Hop Network folder to display the related 7-Hop cases.
2. This designed in NetSim's MANET library. It comprises of 8 Wireless nodes connected via an Adhoc link and the Application is created between Source ID Wireless\_Node\_1 and destination ID Wireless\_Node\_8.

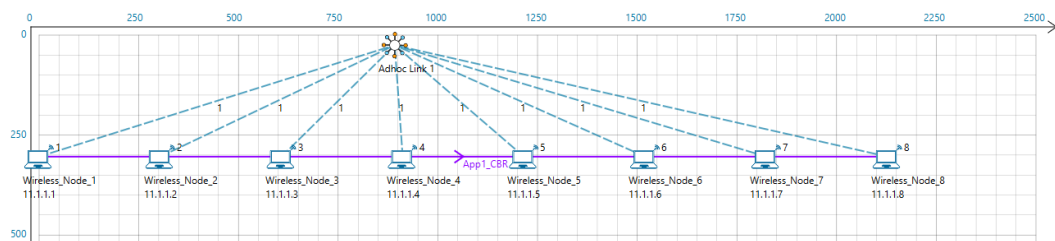


Fig 56: Network Scenario for 7-Hop Network designed with 8 wireless nodes

3. The Simulation parameters setting remains the same as previous case

## Additional Setting

4. Grid length 2500m x2500m

Devices	X	Y
Wireless_Node_1	0	300
Wireless_Node_2	300	300

Wireless_Node_3	600	300
Wireless_Node_4	900	300
Wireless_Node_5	1200	300
Wireless_Node_6	1500	300
Wireless_Node_7	1800	300
Wireless_Node_8	2100	300

Table 29: X and Y Co- ordinates of devices for 7-Hop Network

- Static route should be carefully configured to ensure the data flow from Wireless\_Node\_1>Wireless\_Node\_2>Wireless\_Node\_3> Wireless\_Node\_4> Wireless\_Node\_5> Wireless\_Node\_6>Wireless\_Node\_7>Wireless\_Node\_8
- Dot11\_RTS Threshold value is changed to 750 bytes in all wireless nodes.
- Run the simulation for 100 sec and tabulate the throughput.

## Numerical Results

The following data is used to generate the plot.

Generation Rate (Mbps)	IAT ( $\mu$ s)	Throughput (Mbps)
0.05	160000	0.050021
0.1	80000	0.100042
0.15	53333	0.149979
0.2	40000	0.2
0.25	32000	0.250189
0.3	26666	0.300042
0.35	22857	0.327074
0.4	20000	0.313432
0.45	17777	0.310989
0.5	16000	0.274611
0.51	15686	0.265768
0.52	15384	0.255411
0.53	15094	0.244632
0.54	14814	0.244042
0.55	14545	0.244126
0.56	14285	0.235368
0.57	14035	0.235621
0.58	13793	0.235705
0.59	13559	0.236716
0.6	13333	0.226947
0.65	12307	0.217768
0.7	11428	0.221558
0.75	10666	0.227958
0.8	10000	0.215832
0.85	9411	0.2224
0.9	8888	0.223495
0.95	8421	0.234442
1	8000	0.220042

Table 30: Numerical results for 7-Hop Network

## Changes done to the config file to run the multi-Parameter sweeper

- The Configuration file saved with above setting is replaced in the Multi parameter folder and Inter arrival time value mentioned as {0}

2. Number of nodes used in Network scenario is given to copy the Static Route Configuration text files to IO path same as Configuration.netsim

```

1 import subprocess
2 import shlex
3 import random
4 import shutil
5 import math
6 import sys
7 import datetime
8 import time
9 import os
10
11 #Set the path of 64 bit NetSim Binaries to be used for simulation.
12 NETSIM_PATH="C:\\Users\\Navya\\Documents\\NetSim\\Workspaces\\Performance-Analysis-Wireless-Adhoc-Network\\bin_x64"
13 NUMBER_OF_NODES=8
14
15 #Set NETSIM_AUTO environment variable to avoid keyboard interrupt at the end of each simulation
16 os.environ["NETSIM_AUTO"] = '1'
17
18 #Create IOPATH directory to store the input Configuration.netsim file and the simulation output files during each iteration
19 if not os.path.exists('IOPATH'):
20     os.makedirs('IOPATH')
21
22 #Create Data directory to store the Configuration.netsim and the Metrics.xml files associated with each iteration
23 if not os.path.exists('Data'):
24     os.makedirs('Data')
25
26 #Clean the IOPATH folder if it has any files created during previous multi-parameter sweep runs
27 for root, dirs, files in os.walk('IOPATH'):
28     for file in files:
29         os.remove(os.path.join(root, file))
30

```

Fig 57: Number of nodes in scenario is given as input to copy the Static route configuration files to IO path

3. If the user wishes to change the output folder name, it can be renamed as 7-Hop.

```

110 os.rename("Script"+str(n)+".txt","Script.txt");
111 os.system("MetricsReader.exe result.csv")
112 csvfile = open("result.csv", 'a')
113 csvfile.write(',')
114 csvfile.close()
115 os.rename("Script.txt","Script"+str(n)+".txt");
116
117 else:
118     #Update the output Metric as crash if Metrics.xml file is missing
119     csvfile = open("result.csv", 'a')
120     csvfile.write('\n'+str(i)+','+str('crash')+','+str(''))
121     csvfile.close()
122
123 #Name of the Output folder to which the results will be saved
124 OUTPUT_PATH="Performance-Analysis-Wireless-Adhoc-Network\\"+str(foldername)+"7-Hop-Network"+str(today)+"\\7-Hop_"+str(Source_rate)+"Mbps"
125
126 if not os.path.exists(OUTPUT_PATH):
127     os.makedirs(OUTPUT_PATH)
128
129 #Create a copy of result.csv file present in sweep folder to date-time folder
130 if os.path.isfile('result.csv'):
131     shutil.copy(os.path.join('result.csv'),'Performance-Analysis-Wireless-Adhoc-Network\\"+str(foldername)+"7-Hop-Network"+str(today))
132
133 #Create a copy of all files that is present in IOPATH to the desired output location
134 files_names = os.listdir('IOPATH')
135 for file_name in files_names:
136     shutil.move(os.path.join('IOPATH', file_name),OUTPUT_PATH)
137
138 #Delete Configuration.netsim file created during the last iteration

```

Fig 58: Renaming the output folder by Number of Hop count

4. The sweeper process is started by opening command prompt in the directory of the Multi-Parameter-Sweeper project.
5. Then the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Inter Arrival time and provides the results in results.csv file as shown in below

	A	B
	INTER_ARRIVAL_TIME(micro sec)	THROUGHPUT(Mbps)
1		
2	160000	0.050021
3	80000	0.100042
4	53333	0.149979
5	40000	0.2
6	32000	0.250189
7	26666	0.300042
8	22857	0.327074
9	20000	0.313432
10	17777	0.310989
11	16000	0.274611
12	15686	0.265768
13	15384	0.255411
14	15094	0.244632
15	14814	0.244042
16	14545	0.244126
17	14285	0.235368
18	14035	0.235621
19	13793	0.235705
20	13559	0.236716

Fig 59: 7-Hop network results

## Appendix

### Source code modifications involving in Case 1, Case 2, Case 3.

The workspace was modified with the following code changes:

1. The **CSRANGEDIFF** variable was set to -8 dB in the IEEE802\_11\_Phy.h file present in the IEEE802\_11 project.
2. The following lines of code were commented in the IEEE802\_11.h file present in the IEEE802\_11 project.  
**#define \_RECALCULATE\_RX\_SENSITIVITY\_BASED\_ON\_PEP\_**

To modify the source code, the steps given below are to be followed:

1. Open NetSim source codes via Your Work > Source Code > Open Code option

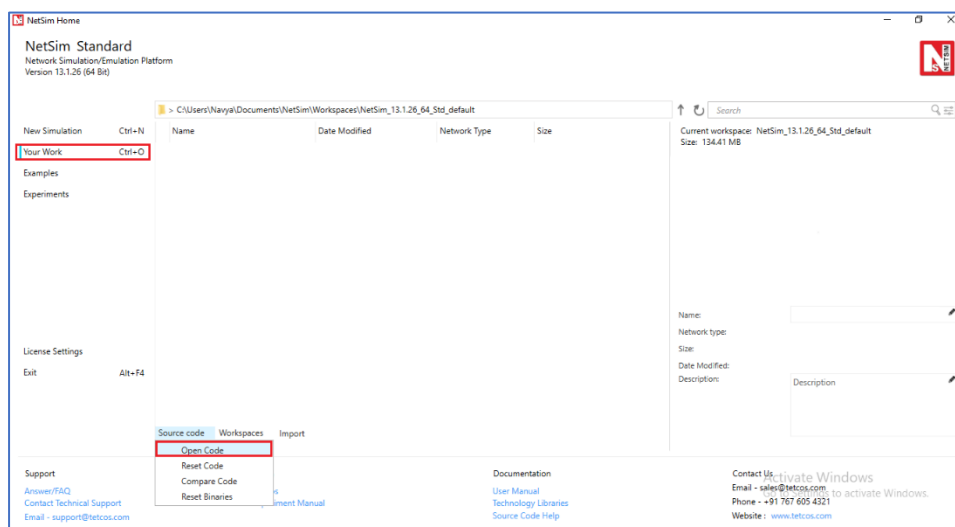


Fig 60: NetSim Open Code option

- This will open the NetSim Source codes in Visual Studio as shown below:

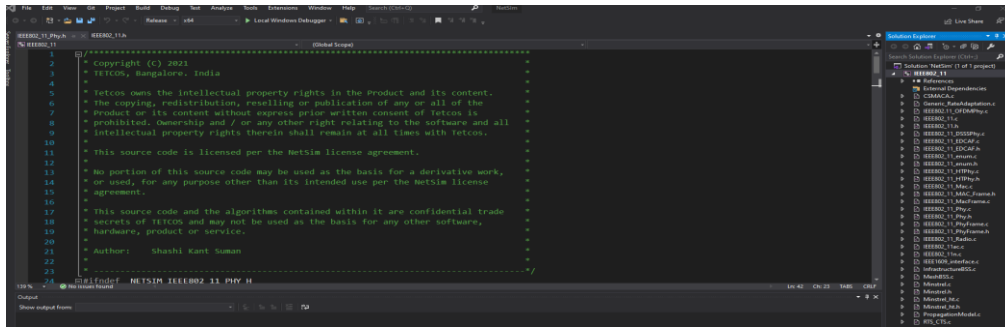


Fig 61: NetSim source codes in Visual Studio

- Go to IEEE802\_11\_Phy.h file in the IEEE802\_11 project. In the line #42, the CSRRANGEDIFF was set to -8 dB.

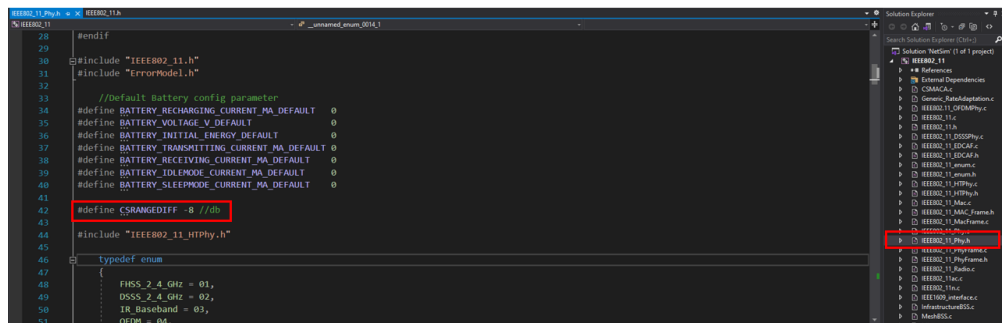


Fig 62: Code modification done in IEEE802\_11\_Phy.h file

- Go to IEEE802\_11.h file in the IEEE802\_11 project and comment the line #38 as shown below

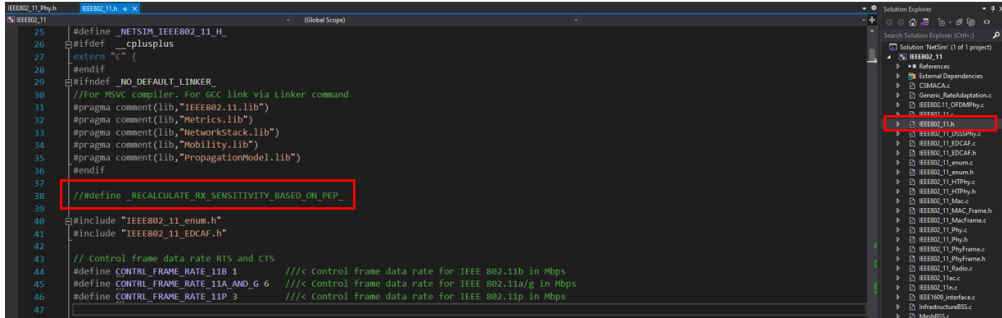


Fig 63: Code modification done in IEEE802\_11.h file

- Right click on the IEEE802\_11 project in the solution Explorer and click on Rebuild.

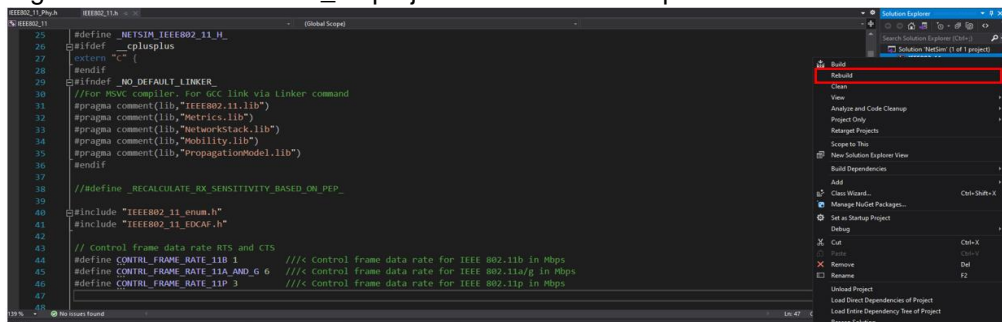


Fig 64: Rebuild code option

6. Upon a successful build, the dll in the binaries folder of the current workspace will automatically get updated.

## Source code modifications involving in Case 4, Case 5, Case 6, Case 7

The workspace was modified with the following code changes:

1. The **CSRANGEDIFF** variable was set to -3 dB in the IEEE802\_11\_Phy.h file present in the IEEE802\_11 project.
2. The following lines of code were commented in the IEEE802\_11.h file present in the IEEE802\_11 project.  
**#Define\_RECALCULATE\_RX\_SENSITIVITY\_BASED\_ON\_PEP\_**

To modify the source code, the steps given below are to be followed:

1. Open NetSim source codes via Your Work > Source Code > Open Code option

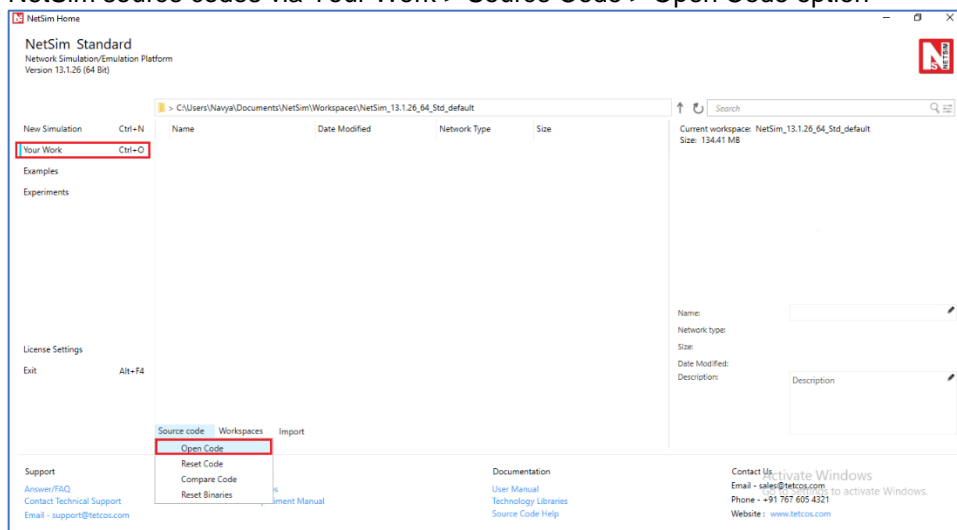


Fig 65: NetSim Open Code option

2. This will open the NetSim Source codes in Visual Studio as shown below:

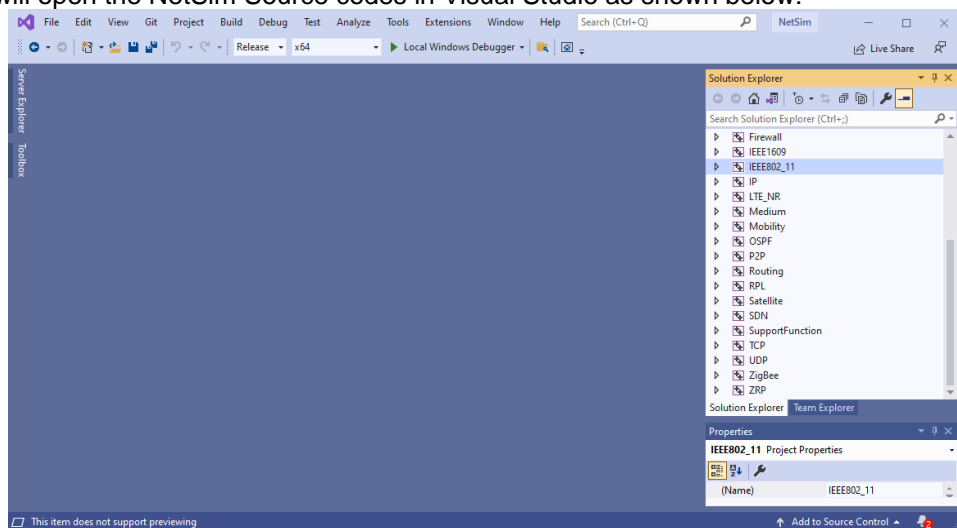


Fig 66: NetSim Source code in Visual Studios



- Go to IEEE802\_11\_Phy.h file inside the IEEE802\_11 project. Modify the CSRANGEDIFF to -3 dB in line #42.

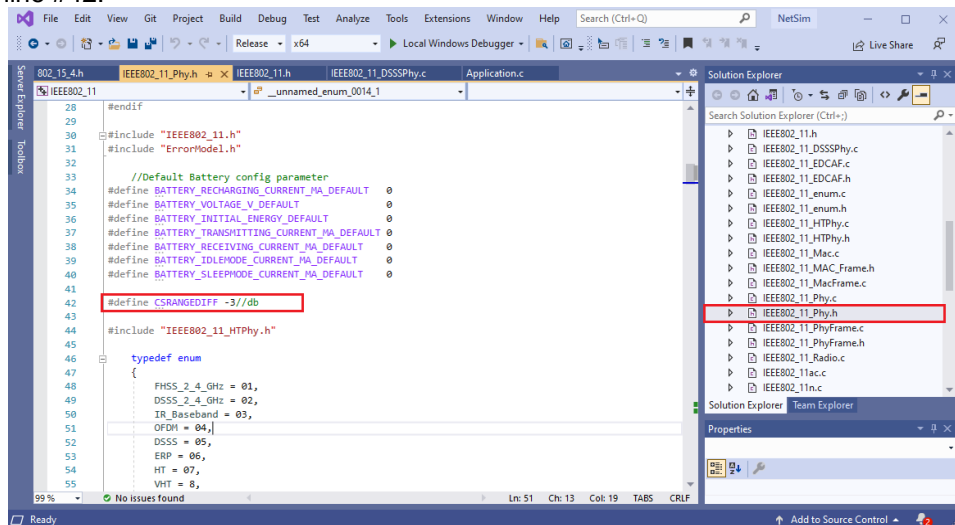


Fig 67: Code modification done in IEEE802\_11\_Phy.h file

- Go to IEEE802\_11.h file in the IEEE802\_11 project and comment the line #38 as shown below

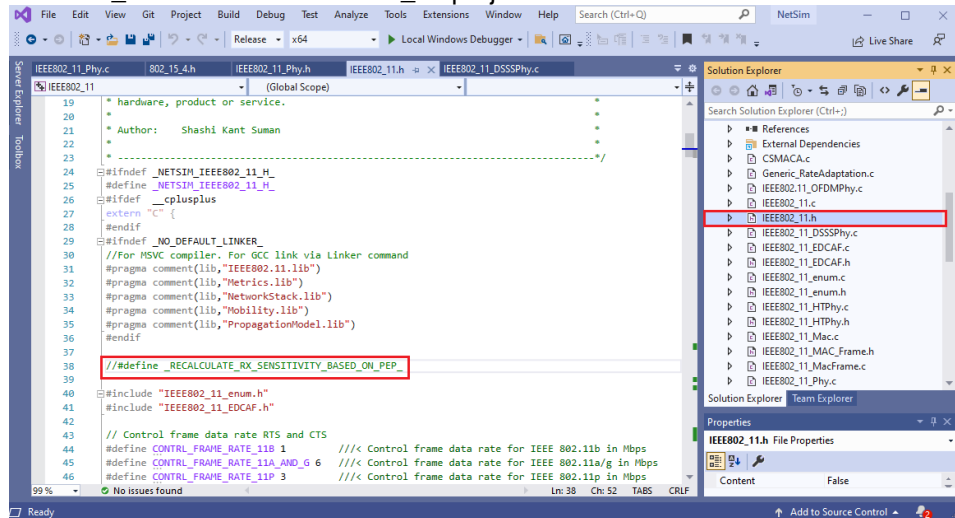


Fig 68: Code modification done in IEEE802\_11.h file

- In IEEE802\_11\_DSSSPhy.c, Receiver sensitivity is set to -87 dBm for 2Mbps in line#44

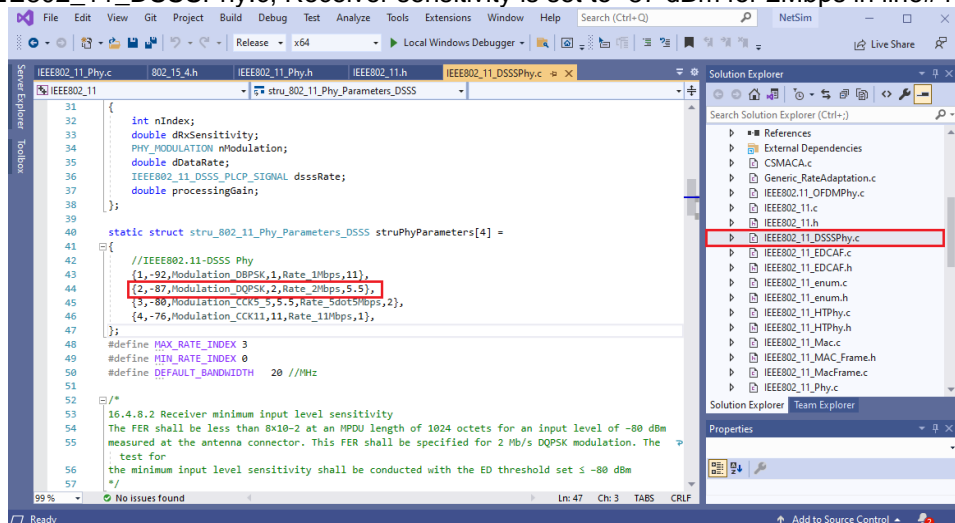


Fig 69: Code modification done in IEEE802\_11\_DSSSPhy.c file

6. Right click on the IEEE802\_11 project in the solution Explorer and click on Rebuild

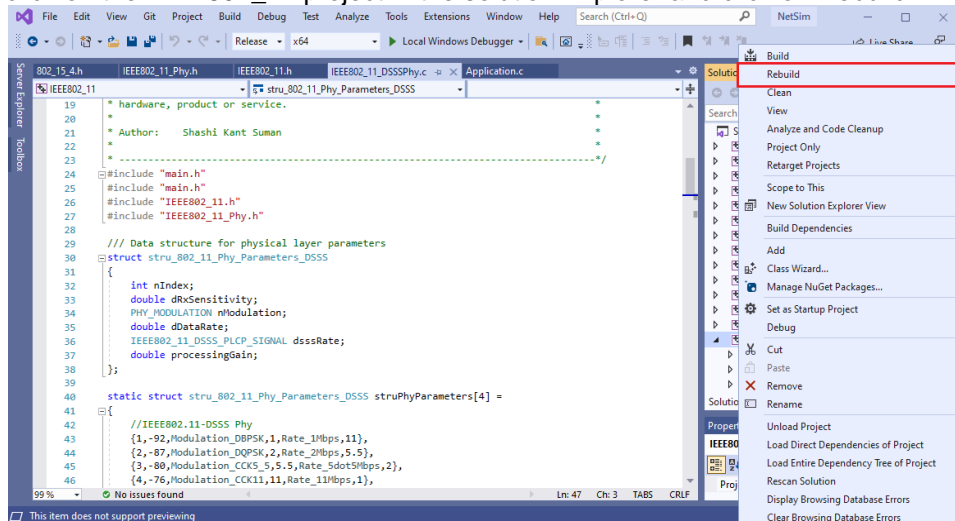


Fig 70: Rebuild 802.11project

7. Upon a successful build, the dll in the binaries folder of the current workspace will automatically get updated.