

NetSim Multi-Parameter Sweeper MATLAB Script

Software: NetSim v14.1 (64 bit), [DOT NET CORE SDK 3.1](#), MATLAB (2022 or higher)

Project Download Link:

[https://github.com/NetSim-TETCOS/Multi-Parameter Sweeper MATLAB v14.1/archive/refs/heads/main.zip](https://github.com/NetSim-TETCOS/Multi-Parameter-Sweeper-MATLAB-v14.1/archive/refs/heads/main.zip)

Introduction

When users want to *sweep* one or more parameters, they change their values between simulation runs, and compare and analyse the performance metrics from each run. NetSim multi-parameter sweeper enables users to automate the sweep process.

Consider an example, where a user wishes to create and simulate a network scenario for all possible values of one or more parameters in combination and analyse a set of performance metrics across the simulation runs. This is extremely time consuming to do manually using the NetSim GUI.

The multi-parameter sweep program enables users to automate the sweep process across multiple input parameters, simulate each run, save each result, and compare specific output metrics via a spreadsheet software like MS Excel.

The sweep program runs NetSim via its CLI interface.

File Organization

The project directory consists of several binaries which are responsible for different tasks during a multi-parameter sweep:

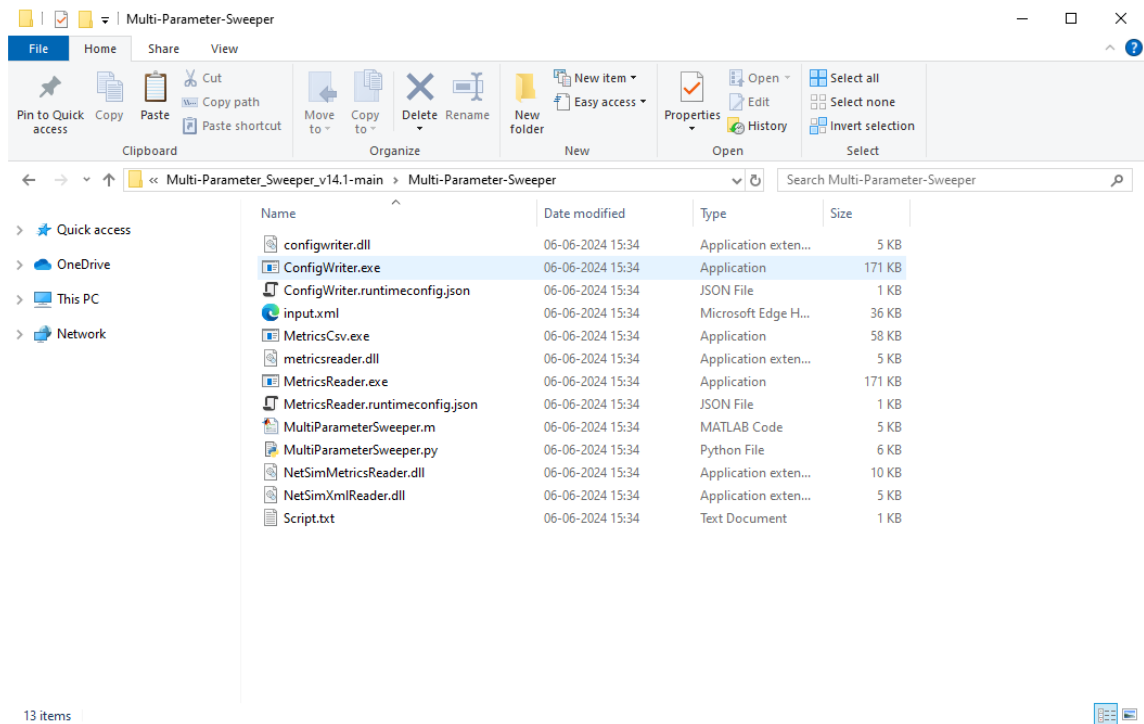


Figure 1: project directory consists of several binaries

1. **input.xml:** This file contains the base NetSim network configuration that is to be simulated. This file is created by copy pasting the Configuration.netsim file that can be obtained by saving a network configuration in NetSim and renaming it to input.xml.

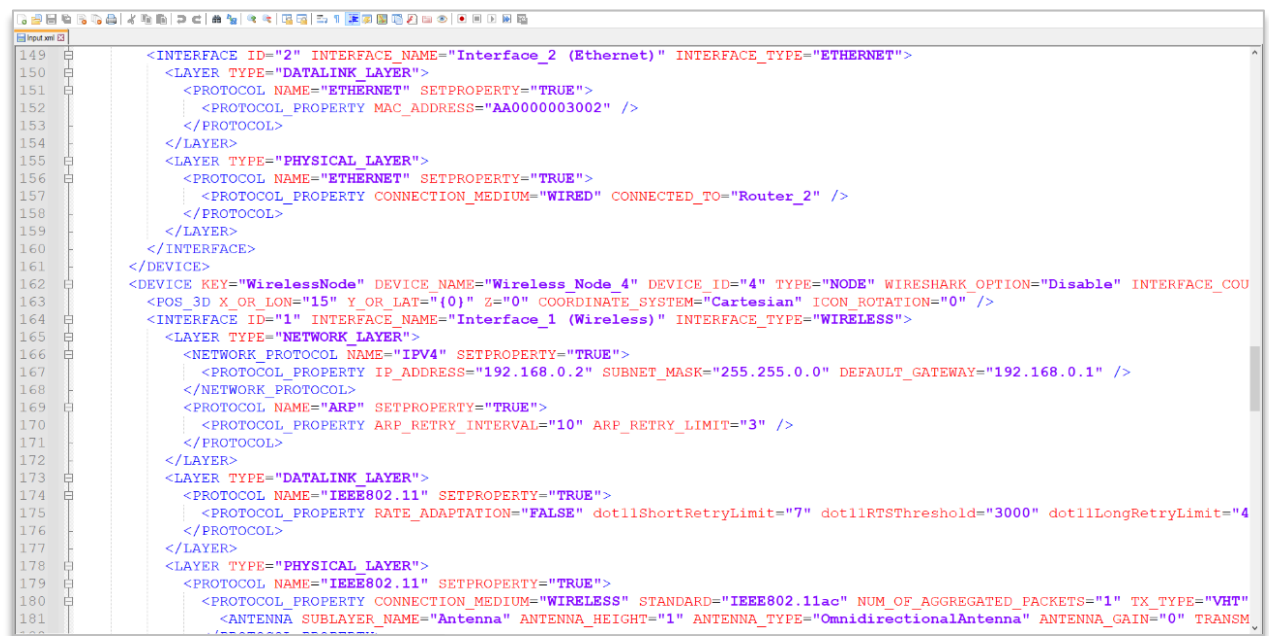
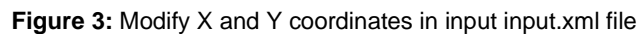


Figure 2: NetSim input Configuration.netsim file

- The values of parameters which are to be varied during each simulation run needs to be specified as {0}, {1}, {2}, etc. respectively.



At the end of every simulation, NetSim generates a Metrics.xml file which contain the performance metrics written in a specific format based on which it is loaded in the results dashboard.

Each Metric is part of a results table which can be accessed using a menu in the results dashboard.

A NetSim Metrics.xml file is shown below:

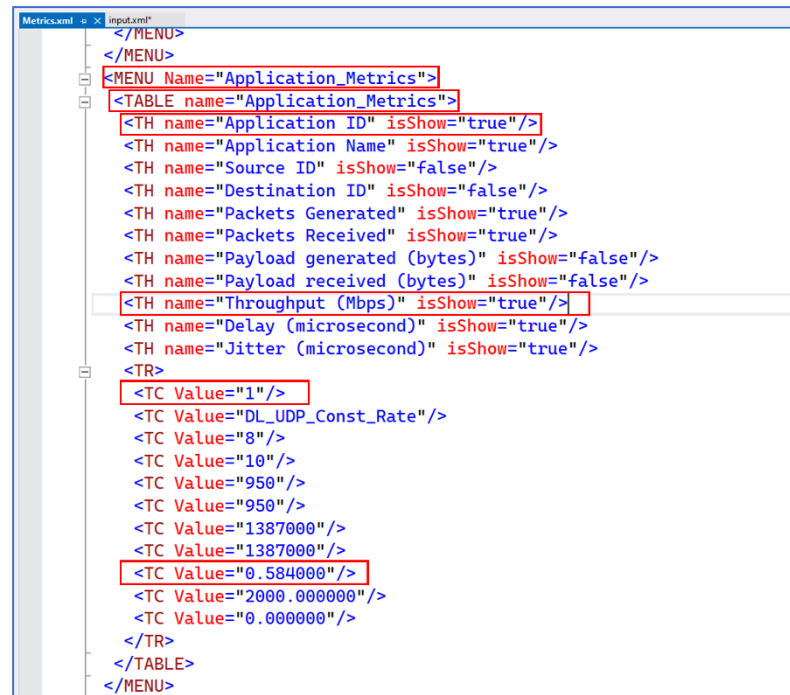


Figure 4: NetSim output Metrics.xml file

For Example, if the application throughput is to be logged for each simulation run then the script file can be updated as shown below:

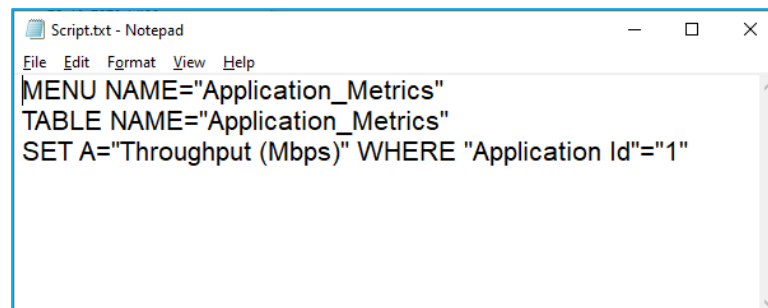


Figure 5: The application throughput is to be logged for each simulation modified in Script.txt

3. **ConfigWriter.exe**: This executable takes one or more command line arguments as input and generated Configuration.netsim file by replacing the arguments in place of the variable parameters specified in the input.xml file.

If there are two variable parameters specified in the input.xml file ({0} and {1}) then two arguments need to be passed while calling ConfigWriter.exe.

4. **MetricsCSV.exe**: This executable is used to convert the Metrics.xml file present in the output folder into a comma separated file, MetricsPrint.csv.

5. **MetricsReader.exe:** This executable is responsible for reading the output parameter from the Metrics.xml file generated after each simulation and logging it to the results file.

It uses the Script.txt file to determine which parameter to read from the Metrics file.

If multiple parameters are to be read and logged, then the MetricsReader.exe can be called multiple times with Script.txt file having information about the parameter to be read each time.

6. **Supporting DLL's:** Some the supporting files such as ConfigWriter.dll, MetricsReader.dll, NetSimMetricsReader.dll, NetSimXmlReader.dll, etc. which are present in the project folder are used by other executable such as ConfigWriter.exe and MetricsReader.exe for various purposes during a multi-parameter sweep.

7. **MultiParameterSweeper.m** uses MATLAB programming language which is less complex and offers more flexibility as the number of input and output parameters increases.

- Users can also write the script to run the multi-parameter sweep process in a preferred programming language as per the convenience.
- The script can be configured to run multiple simulation iterations based on the number of parameters to be varied and the range of values of each parameter.
- NETSIM_PATH variable can be set to the path of NetSim 64-bit binaries (bin_x64) in the install directory or workspace which is to be used to run Simulations.
- LICENSE_ARG variable can be set to License server port and IP details in case of floating on premise licenses or the path of license file in case of node locked or cloud licenses.

```

1 % Set the path of 64 bit NetSim Binaries to be used for simulation.
2 NETSIM_PATH = "C:\Program Files\NetSim\Standard_v14_1\bin\bin_x64";
3 LICENSE_ARG = "5053@192.168.0.9"; %Floating license
4 %LICENSE_ARG = ""C:\Program Files\NetSim\Standard_v13_2\bin""; %Node Locked license
5
6 % Set NETSIM_AUTO environment variable to avoid keyboard interrupt at the end of each simulation
7 setenv('NETSIM_AUTO','1');
8
9 % Create IOPath directory to store the input Configuration.netsim file and
10 % the simulation output files during each iteration
11 if(~isfolder('IOPath'))
12     mkdir IOPath
13 end
14
15 % Create Data directory to store the Configuration.netsim and the
16 % Metrics.xml files associated with each iteration
17 if(~isfolder('Data'))
18     mkdir Data
19 end
20

```

Figure 6: User need to set NetSim Path based on 64-bit

For example,

NETSIM_PATH

=

"C:\Users\HP\OneDrive\Documents\NetSim\Workspaces\Default_Workspace_v14_1\bin_x64";

Floating on-premise license (<port no>@<server ip address>):

LICENSE_ARG = "5053@192.168.0.9";

Node Locked or Cloud licenses (path of license file):

LICENSE_ARG = ""C:\Program Files\NetSim\Standard_v14_1\bin"";

- The MetricsCSV.exe will convert the Metrics.xml file inside the output folder into a csv file.

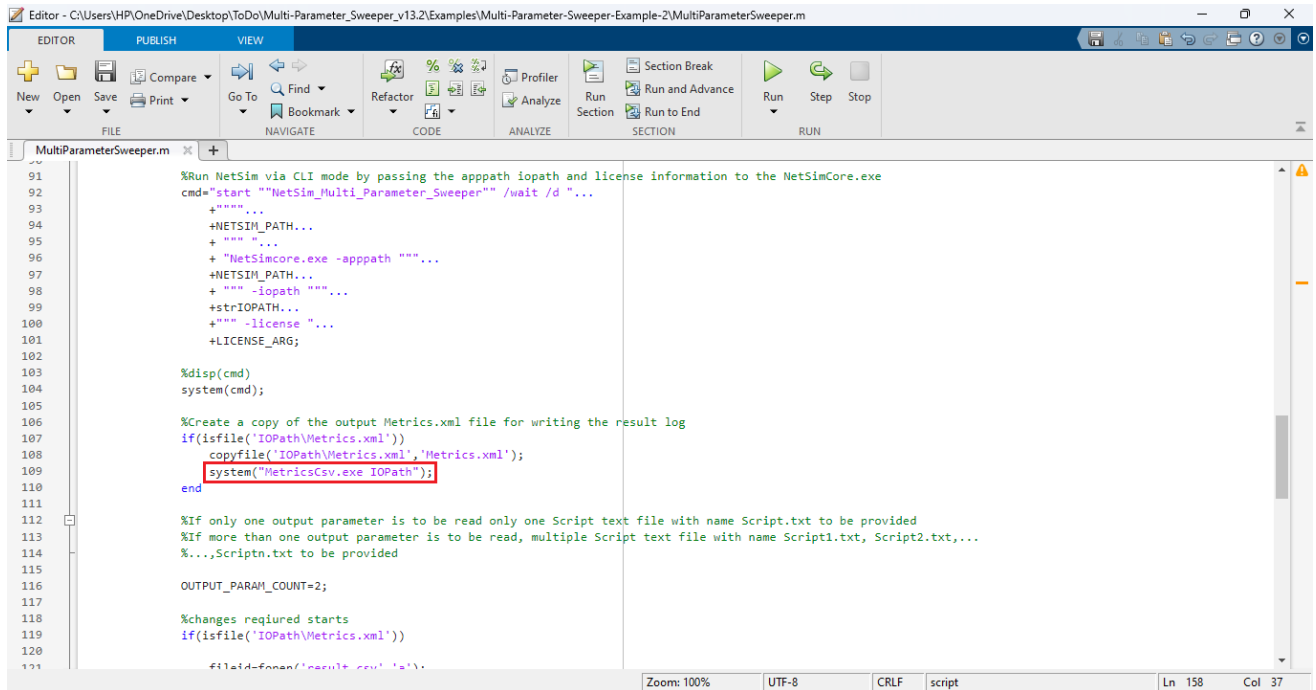


Figure 7: MetricsPrint.csv file will be created in the IOPATH and then copied into the respective output folder

8. Multi-Parameter Sweeping process can be started from MATLAB command line in the directory of the Multi-Parameter-Sweeping project as shown below:

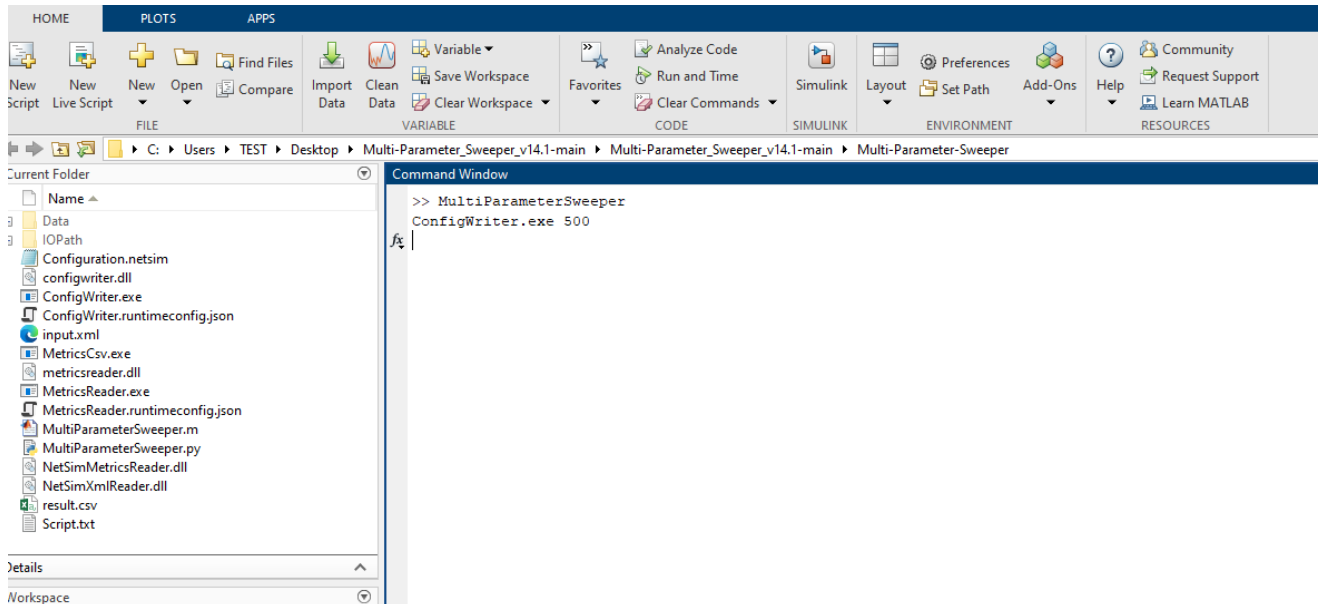


Figure 8: Running MATLAB script using cmd prompt

9. This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Y parameter of UE.

10. At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:

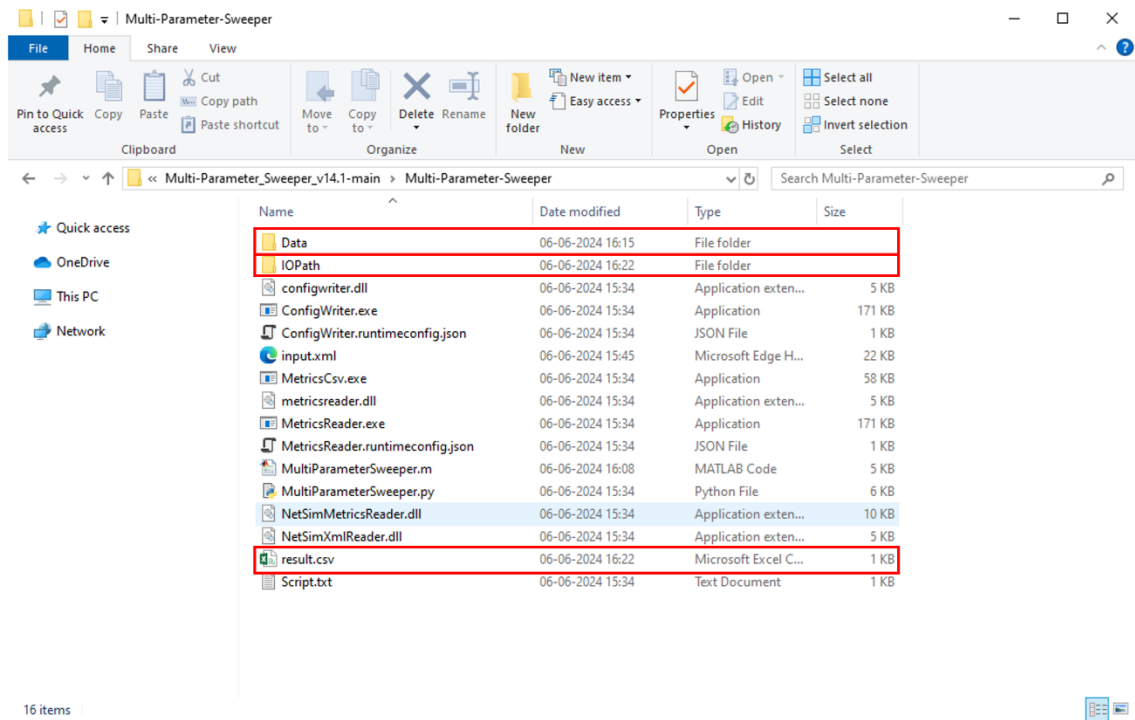


Figure 9: After Simulation Multi-Parameter-Sweeping folder contains output files like result.scv, Data etc

- **Data:** Contains multiple folders created based on date and time of simulation inside which multiple output folders corresponding to each simulation run, with its name including the value of the parameters in that iteration gets created.

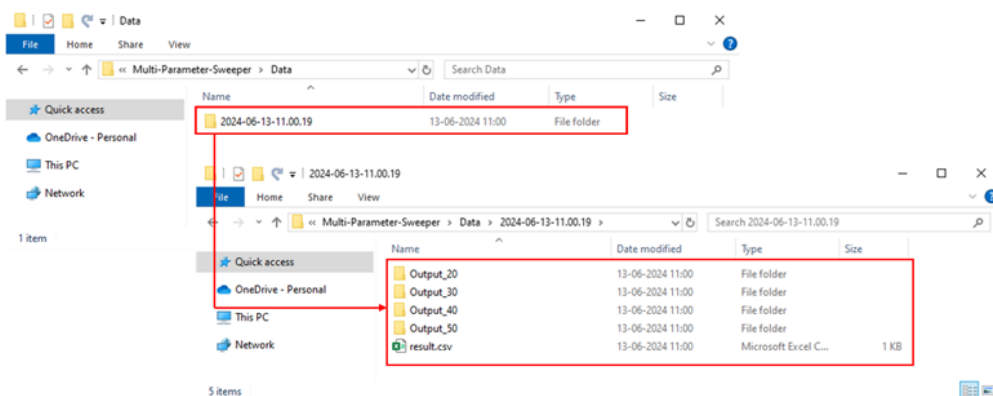


Figure 10: Based on values in the iteration, Output folder gets created in the Data directory inside a folder named as per date and time of simulation along with a copy of result.csv file.

- Each folder contains the all the output files associated with the simulation run.

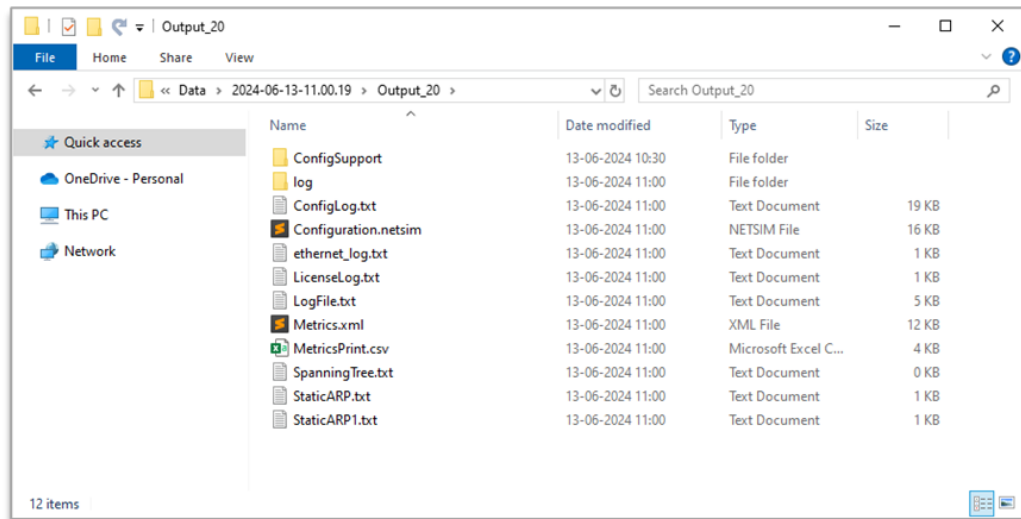


Figure 11: Each folder contains all the output files and the Metrics.xml file converted to MetricsPrint.csv file

Note: User should keep a back-up of the data folder to avoid data loss.

- **IO Path:** Used for storing the Configuration.netsim file and the simulation files generated during each simulation run.
- **Result.csv:** This is the output log which contains the parameter varied during each simulation run and the output parameter associated with each run. The result.csv file will also be copied into the output folder after each simulation.

	A	B	C	D	E
1	Y	THROUGHPUT(Mbps)			
2	20	21.616176			
3	30	14.703952			
4	40	9.561248			
5	50	0			
6					

Figure 12: Iterated value and Throughput obtained listed in result.csv

Example 1: Modifying a single input parameter and logging a single output parameter

Consider the following network Internetwork (WiFi) network scenario in NetSim, comprising of a Wired Node, Router, Access Point and a Wireless Node.

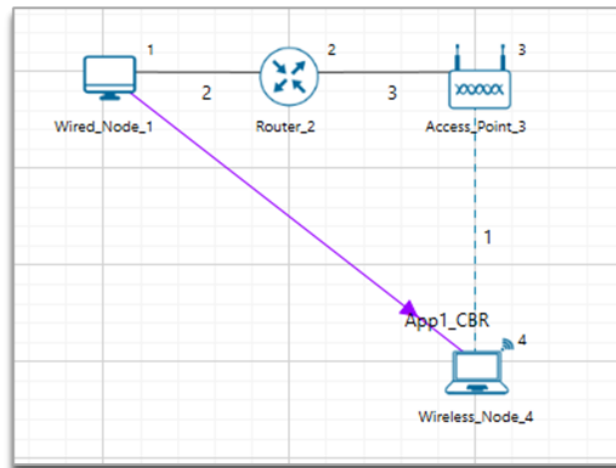


Figure 13: Network Topology

The network configuration has the initial distance between the Access Point and Wireless Node as 20 meters with the Access Point located at (15,10) and UE located at (15,20).

Multi-Parameter Sweeper is configured to run simulations for different distance between the gNB and UE by varying the UE Y coordinate value from 20 to 50 in steps of 10 meters.

The network scenario is saved and the content of the Configuration.netsim file is copied to the Multi-Parameter-Sweeper directory and renamed as input.xml.

Refer to the Example 1 directory which is part of the project folder (Multi-Parameter-Sweeper_v14.1-main\Examples\Multi-Parameter-Sweeper-Example-1)

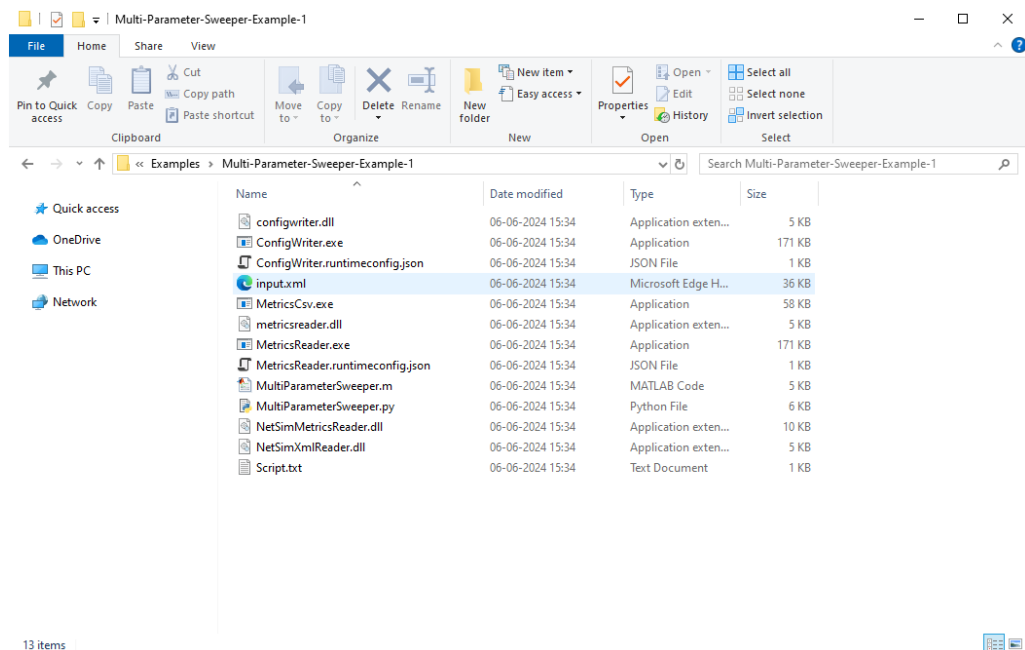


Figure 14: Renamed Configuration.netsim to input.xml and pasted in Multi-Parameter-Sweeper directory

1. The value of the Y coordinate of UE that is to be modified during each simulation run is updated (“{0}”) in the configuration file as shown below:

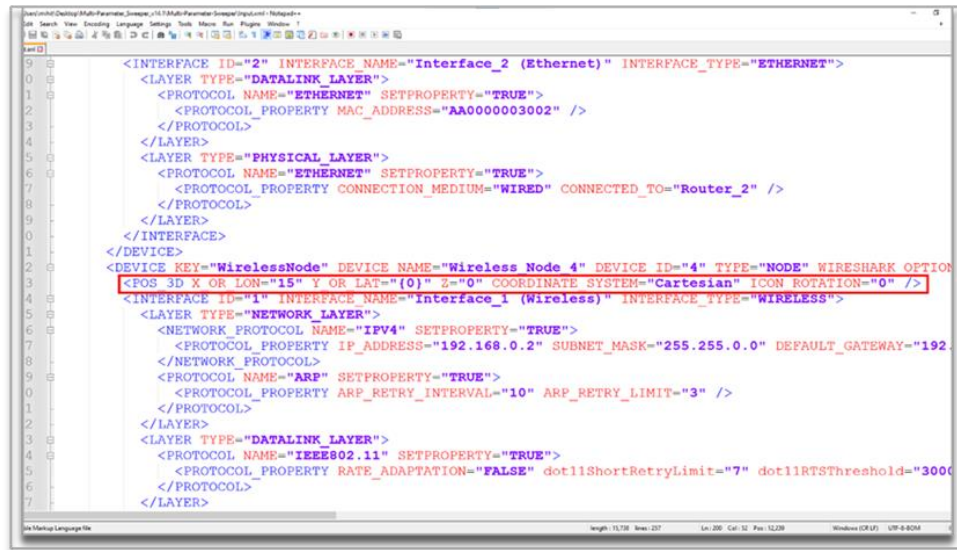


Figure 15: Modified Y coordinate of UE in input.xml

2. The Script.txt file is updated with the details of the output parameter to be read from the Metrics.xml file and added to the result csv log file. In this case the Application throughput is to be logged for each simulation run.

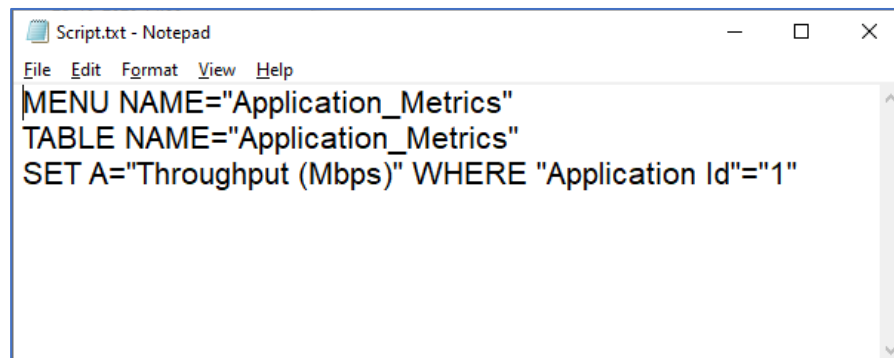


Figure 16: The application throughput is to be logged for each simulation modified in Script.txt

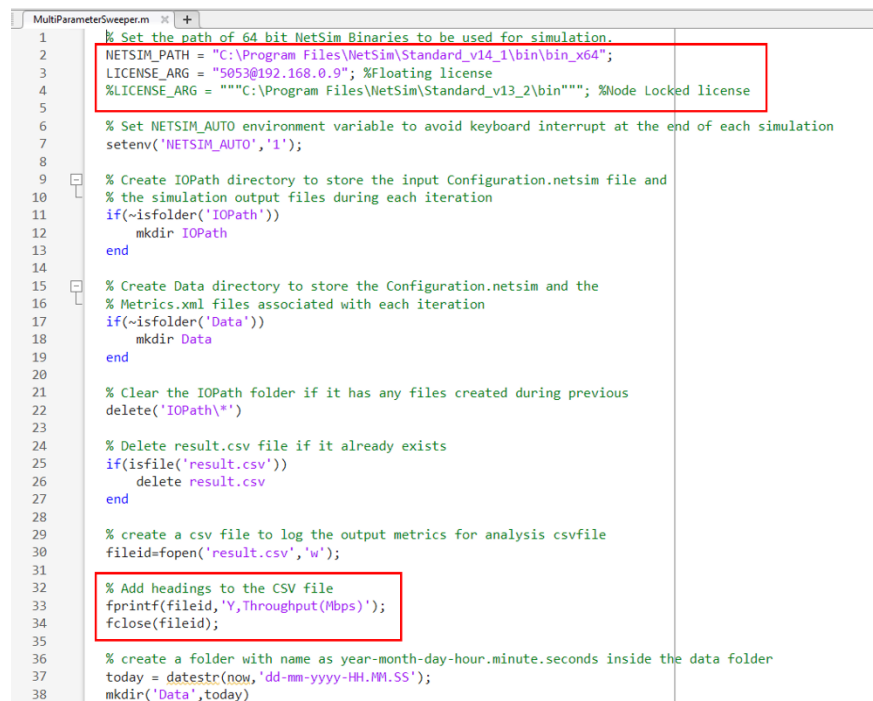
3. MultiParameterSweeper.m is updated to pass the Y coordinate value during each iteration to generate Configuration file run simulation and update the result csv log.

The MultiParameterSweeper.bat batch script modified for running simulations for different values of Y coordinates starting from 20 up to 50 in steps of 10 is shown below:

- A result.csv file is created and added with headings Y and Throughput (Mbps).
- A MetricsPrint.csv is created inside every output folder.

- For loop is set to iteratively run simulations for values starting from 20 to 50 in steps of 10.
- The value of the parameter Y in the current iteration is written to the result log file for analysis.
- The value of the parameter Y in the current iteration is passed as input to ConfigWriter executable to generate Configuration.netsim file for each simulation.
- NetSim simulation is run via CLI mode by passing the apppath, iopath and license server information
- Configuration file and Metrics file are copied and renamed appending the value of the parameter in the current iteration.

The MultiParameterSweeper.m matlab script modified for running simulations for different values of Y coordinates starting from 20 up to 50 in steps of 10 is shown below:



```

1 % Set the path of 64 bit NetSim Binaries to be used for simulation.
2 NETSIM_PATH = "C:\Program Files\NetSim\Standard_v14_1\bin\bin_x64";
3 LICENSE_ARG = "5053@192.168.0.9"; %Floating license
4 %LICENSE_ARG = ""C:\Program Files\NetSim\Standard_v13_2\bin""; %Node Locked license
5
6 % Set NETSIM_AUTO environment variable to avoid keyboard interrupt at the end of each simulation
7 setenv('NETSIM_AUTO','1');
8
9 % Create IOPath directory to store the input Configuration.netsim file and
10 % the simulation output files during each iteration
11 if(~isfolder('IOPath'))
12     mkdir IOPath
13 end
14
15 % Create Data directory to store the Configuration.netsim and the
16 % Metrics.xml files associated with each iteration
17 if(~isfolder('Data'))
18     mkdir Data
19 end
20
21 % Clear the IOPath folder if it has any files created during previous
22 delete('IOPath\*')
23
24 % Delete result.csv file if it already exists
25 if(isfile('result.csv'))
26     delete result.csv
27 end
28
29 % create a csv file to log the output metrics for analysis csvfile
30 fileid=fopen('result.csv','w');
31
32 % Add headings to the CSV file
33 fprintf(fileid,'Y,Throughput(Mbps)');
34 fclose(fileid);
35
36 % create a folder with name as year-month-day-hour.minute.seconds inside the data folder
37 today = datestr(now,'dd-mm-yyyy-HH.MM.SS');
38 mkdir('Data',today)

```

Figure 17: To Create result.csv file, added with headings Y and Throughput (Mbps) and NetSim installation Path and License information

- NETSIM_PATH variable is set to the path of NetSim 64-bit binaries in the install directory or workspace in the system.
- LICENSE_ARG variable is set to the license server details
- A result.csv file is created and added with headings Y and Throughput (Mbps).

```

41 % Iterate based on the number of time simulation needs to be run and the
42 for i = 20:10:50
43     if(isfile('Configuration.netsim'))
44         delete Configuration.netsim
45     end
46     if(isfile('IOPATH\Configuration.netsim'))
47         delete('IOPATH\Configuration.netsim')
48     end
49     if(isfile('IOPATH\Metrics.xml'))
50         delete('IOPATH\Metrics.xml')
51     end
52     %Call ConfigWriter.exe with arguments as per the number of variable parameters in the input.xml file
53     cmd='ConfigWriter.exe '+string(i);
54     disp(cmd);
55     system(cmd);
56     %Copy the Configuration.netsim file generated by ConfigWriter.exe to IOPATH directory
57     if(isfile('Configuration.netsim'))
58         copyfile('Configuration.netsim','IOPATH');
59     end
60     strIOPATH=pwd+'\IOPATH';
61     %Run NetSim via CLI mode by passing the appath iopath and license information to the NetSimCore.exe
62 end

```

Figure 18: Varying Distance and set license server information

- For loop is set to iteratively run simulations for values starting from 20 to 50 in steps of 10.
- The value of the parameter Y in the current iteration is passed as input to ConfigWriter executable to generate Configuration.netsim file for each simulation.

```

94 %changes required starts
95 if(isfile('IOPATH\Metrics.xml'))
96     fileid=fopen('result.csv','a');
97     if(fileid)
98         fprintf(fileid,'%d\n',i);
99         fclose(fileid);
100     end
101     if(OUTPUT_PARAM_COUNT==1)
102         system('MetricsReader.exe result.csv');
103     else
104         for n = 1 : OUTPUT_PARAM_COUNT
105             filename='Script'+string(n)+'.txt';
106             movefile(filename,'Script.txt');
107             system('MetricsReader.exe result.csv');
108             fileid=fopen('result.csv','a');
109             if(fileid)
110                 fprintf(fileid,'%d\n',i);
111                 fclose(fileid);
112             end
113             movefile('Script.txt',filename);
114         end
115     end
116     %Update the output Metric as crash if Metrics.xml file is missing
117     fileid=fopen('result.csv','a');
118     if(fileid)
119         fprintf(fileid,'%d\n',crash);
120         fclose(fileid);
121     end
122 end
123 % Name of the Output folder to which the result will be saved
124 OUTPUT_PATH = 'Data\'+today+'Output_'+string(i);

```

Figure 19: Modify parameters in MultiParameterSweeper.m

- The value of the parameter Y in the current iteration is written to the result log file for analysis.
 - Configuration file and Metrics file are copied and renamed appending the value of the parameter in the current iteration.
4. Multi-Parameter Sweeping process can be started from MATLAB command line in the directory of the Multi-Parameter-Sweeping project as shown below:

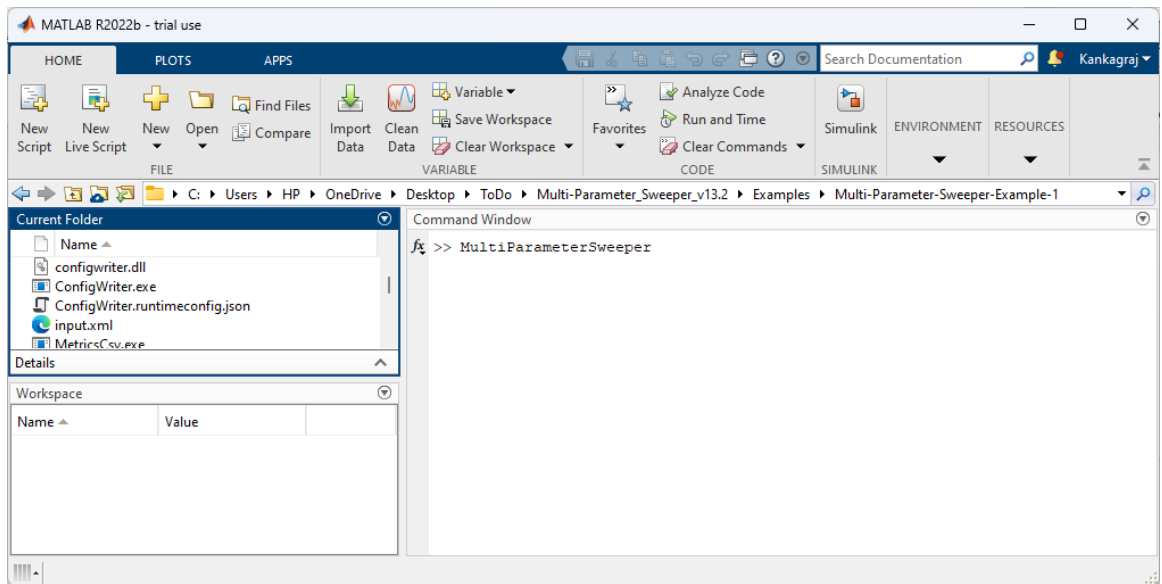


Figure 20: Running MATLAB script using cmd prompt

This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different values of Y parameter of UE.

At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:

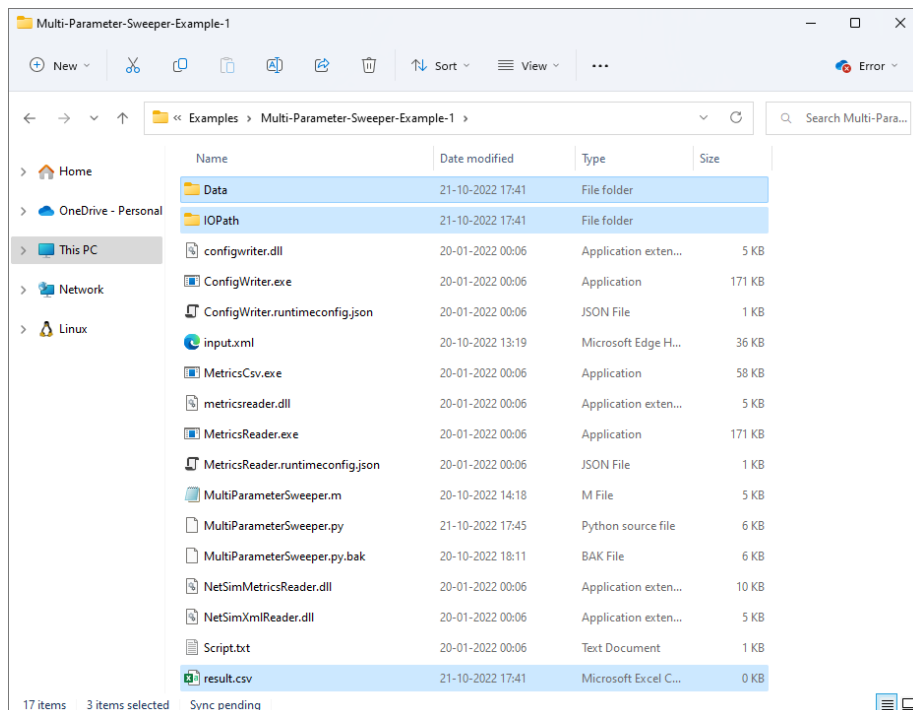


Figure 21: After Simulation Multi-Parameter-Sweeping folder contains output files like result.csv, Data etc

- **Data:** Contains multiple folders corresponding to each simulation run, with its name including the value of the parameters in that iteration. The output folders will be created inside folder with name in the format Year-Month-Day-Hours-Minutes-Seconds.

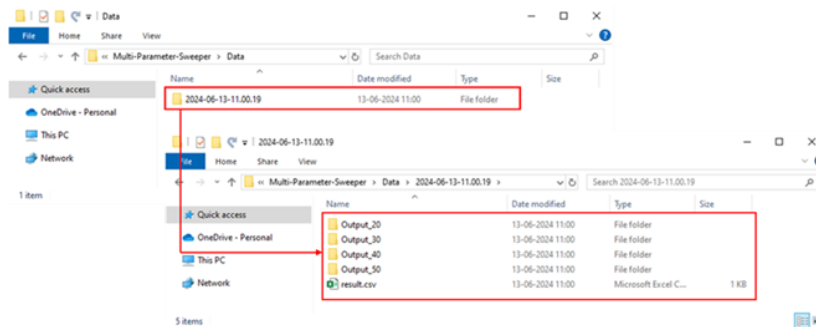


Figure 22: Based on distance Configuration.netsim files created in output folder

- Each folder contains the all the output files associated with the simulation run.

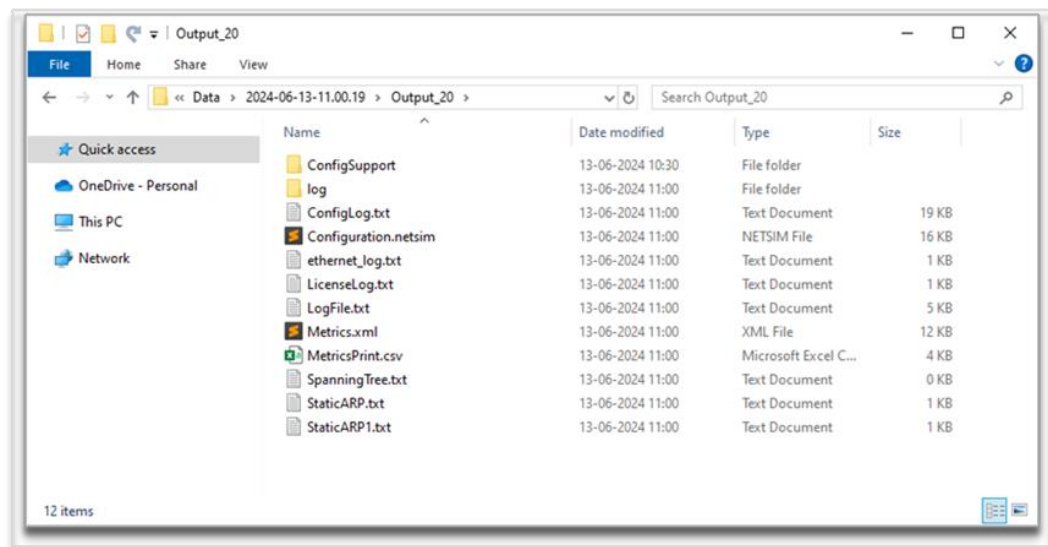


Figure 23: Each folder contains the all the output files

- **IOPath:** Used for storing the Configuration.netsim file and the simulation files generated during each simulation run.
- **Result.csv:** This is the output log which contains the parameter varied during each simulation run and the output parameter associated with each run.

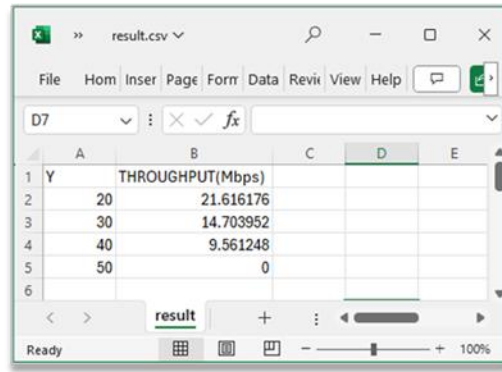


Figure 24: Distance Vs. Throughput obtained in result.csv

Varying multiple network parameters

In order to vary multiple network parameters during the multi-parameter sweep process each parameter in the input.xml file can be modified as {0},{1},{2},{3},...{n} respectively.

Logging multiple output parameters

Each output parameter that is to be logged should be part of the Script.txt file. However, the Script.txt file should contain only the details of one output parameter during the call to MetricsReader.exe.

To log multiple parameters, multiple script files can be used. If n output parameters are to be logged, then there can be script1.txt, script2.txt, script.txt in the sweeper folder.

For Example, there can be two Script files as shown below:

Name	Date modified	Type	Size
Data	21-10-2022 17:16	File folder	
IOPath	21-10-2022 17:17	File folder	
configwriter.dll	20-01-2022 00:06	Application exten...	5 KB
ConfigWriter.exe	20-01-2022 00:06	Application	171 KB
ConfigWriter.runtimeconfig.json	20-01-2022 00:06	JSON File	1 KB
input.xml	20-10-2022 18:40	Microsoft Edge H...	37 KB
MetricsCsv.exe	20-01-2022 00:06	Application	58 KB
metricsreader.dll	20-01-2022 00:06	Application exten...	5 KB
MetricsReader.exe	20-01-2022 00:06	Application	171 KB
MetricsReader.runtimeconfig.json	20-01-2022 00:06	JSON File	1 KB
MultiParameterSweeper.m	21-10-2022 17:16	M File	7 KB
MultiParameterSweeper.py	21-10-2022 17:45	Python source file	8 KB
NetSimMetricsReader.dll	20-01-2022 00:06	Application exten...	10 KB
NetSimXmlReader.dll	20-01-2022 00:06	Application exten...	5 KB
result.csv	21-10-2022 17:18	Microsoft Excel C...	1 KB
Script1.txt	20-01-2022 00:06	Text Document	1 KB
Script2.txt	20-01-2022 00:06	Text Document	1 KB

Figure 25: Multiple output parameters

Example 2: Modifying multiple input parameters and logging multiple output parameter

1. Consider the following network WiFi network scenario in NetSim, comprising of a Wired Node, Router, Access Point and a Wireless Node.

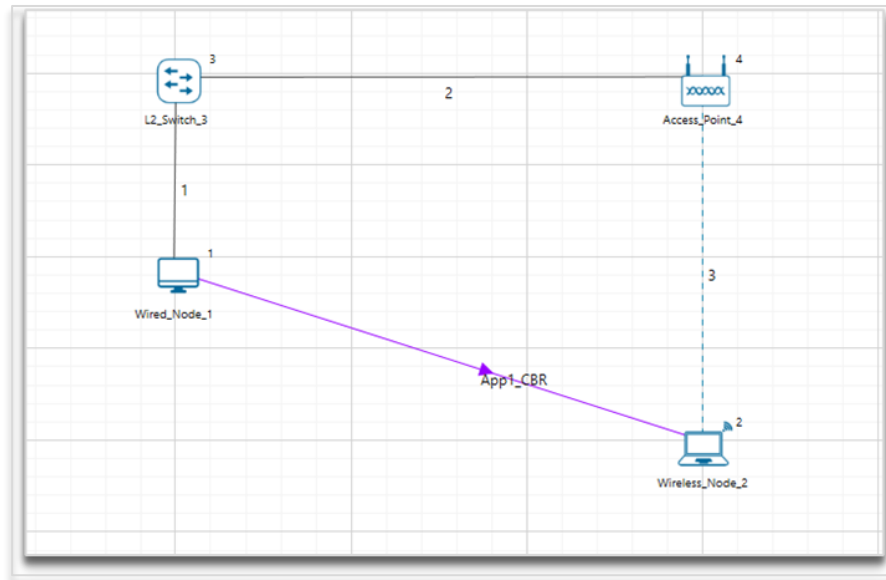


Figure 26: Network Scenario

2. Consider a Grid Length of 50m X 50 m and distance between AP and Wireless node is 20m.
3. Set DCF as the medium access layer protocol under datalink layer properties of access_ point and wireless node
4. WLAN Standard is set to 802.11n and No. of Tx and Rx Antennas is set to 1 in both access point and wireless node.
5. Channel Characteristics -> Path Loss only, Path Loss Model -> Log Distance and Path loss Exponent as 3. (Wireless Link Properties).
6. Click on Set Traffic tab and set CBR application with 50Mbps generation rate. (Set Inter Arrival Time: 233.6 (micro sec)).
7. Set transport protocol to UDP.
8. The network scenario is saved and the content of the Configuration.netsim file is copied to the Multi-Parameter-Sweeper directory and renamed as input.xml.
9. Refer to the Example 2 directory which is part of the project folder (Multi-Parameter-Sweeper_v14.1\Examples\Multi-Parameter-Sweeper-Example-2).

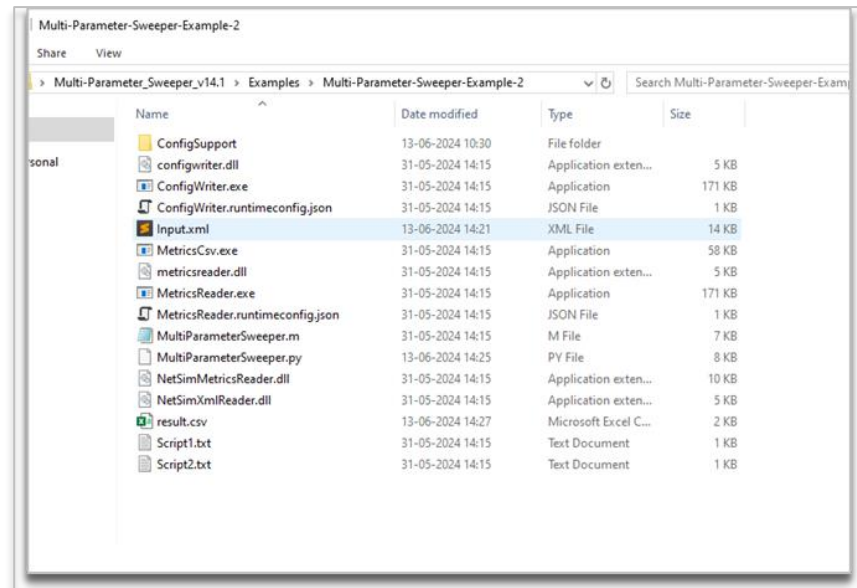


Figure 27: Renamed Configuration.netsim to input.xml and pasted in Multi-Parameter-Sweeper directory
10. In the Input.xml file the value of the input variables are modified as shown in the table below:

Input Variables	
Channel Bandwidth (MHz)	{0}
Transmitting Antennas	{1}
Receiving Antennas	{2}
Inter Arrival Time (Microseconds)	{3}

Table 1: Variables are modified to the input.xml file

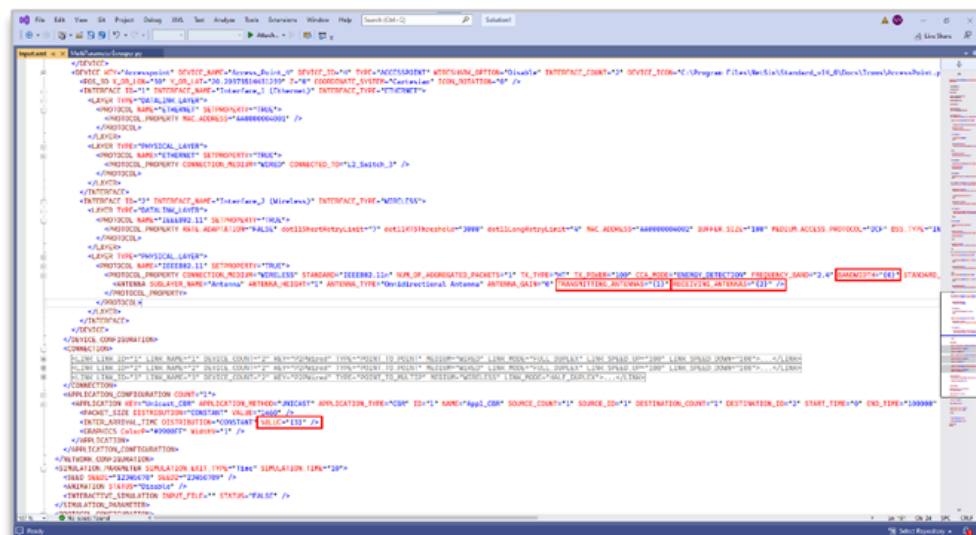


Figure 28: The above table 3 Variables are modified in input.xml file

11. The MATLAB script MultiParameterSweeper.m is modified to run simulation for all possible combinations of Bandwidth and Tx Antenna Count and Rx Antenna Count with the respective IAT that is calculated.

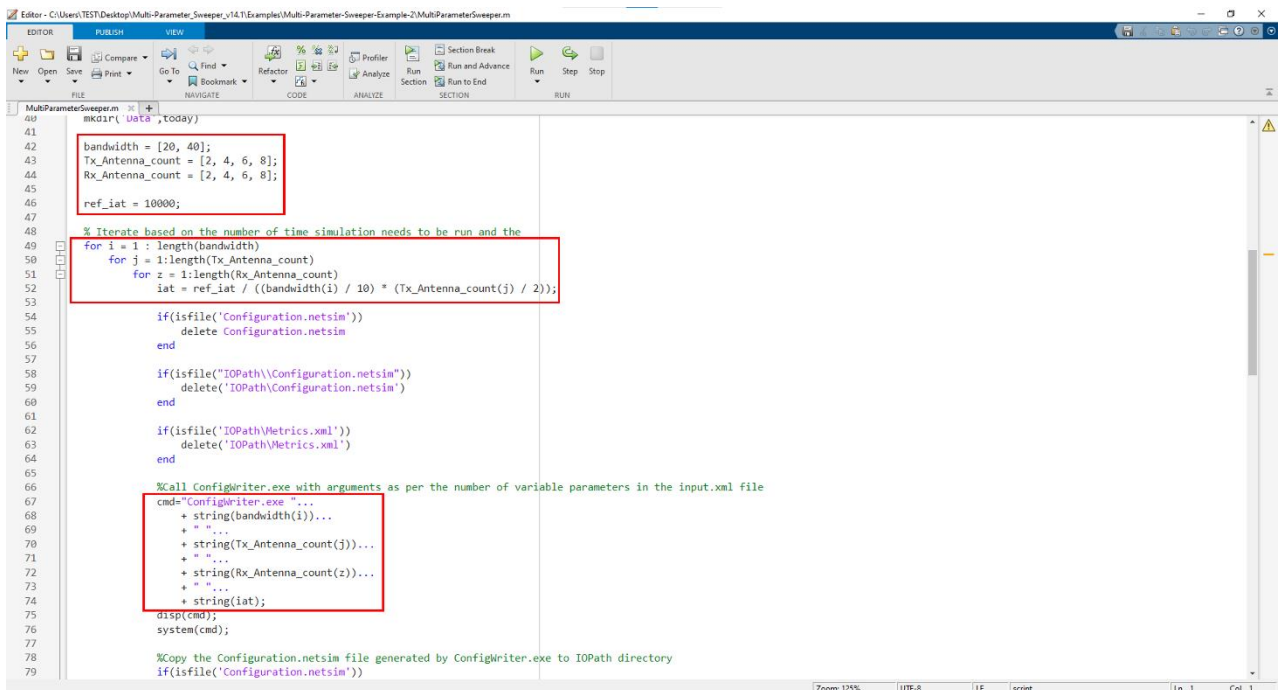


Figure 29: Modified MultiParameterSweeper.m based on input parameter

12. Multiple parameters are read from the Metrics.xml file and logged in the results.csv file along with the input parameters such as BANDWIDTH, TRANSMITTING_ANTENNAS, RECEIVING_ANTENNAS, INTER_ARRIVAL_TIME (micro sec).

Output Parameters
Throughput (Mbps)
Data Packets transmitted

Table 2: User need to modify these two parameters in Script files

13. Two script text files namely Script1.txt and Script2.txt are created with information to read each of the parameters from the Metrics.xml file. The variable OUTPUT_PARAM_COUNT is set to 2 as per the number of Script files.

Script1.txt

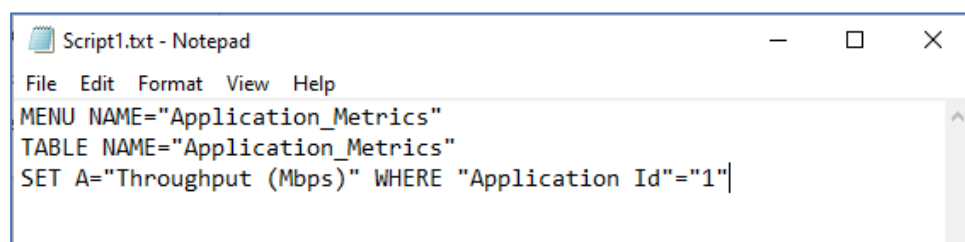
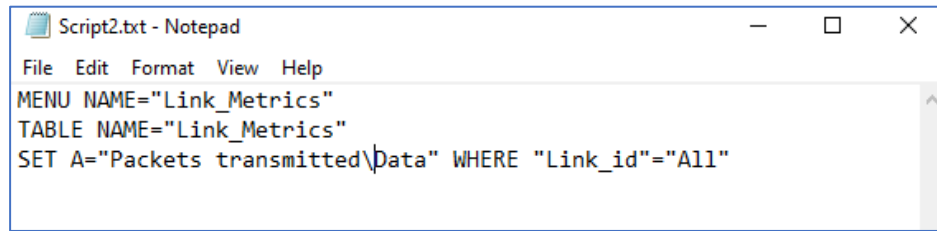


Figure 30: The application throughput is to be logged for each simulation modified in Script1.txt

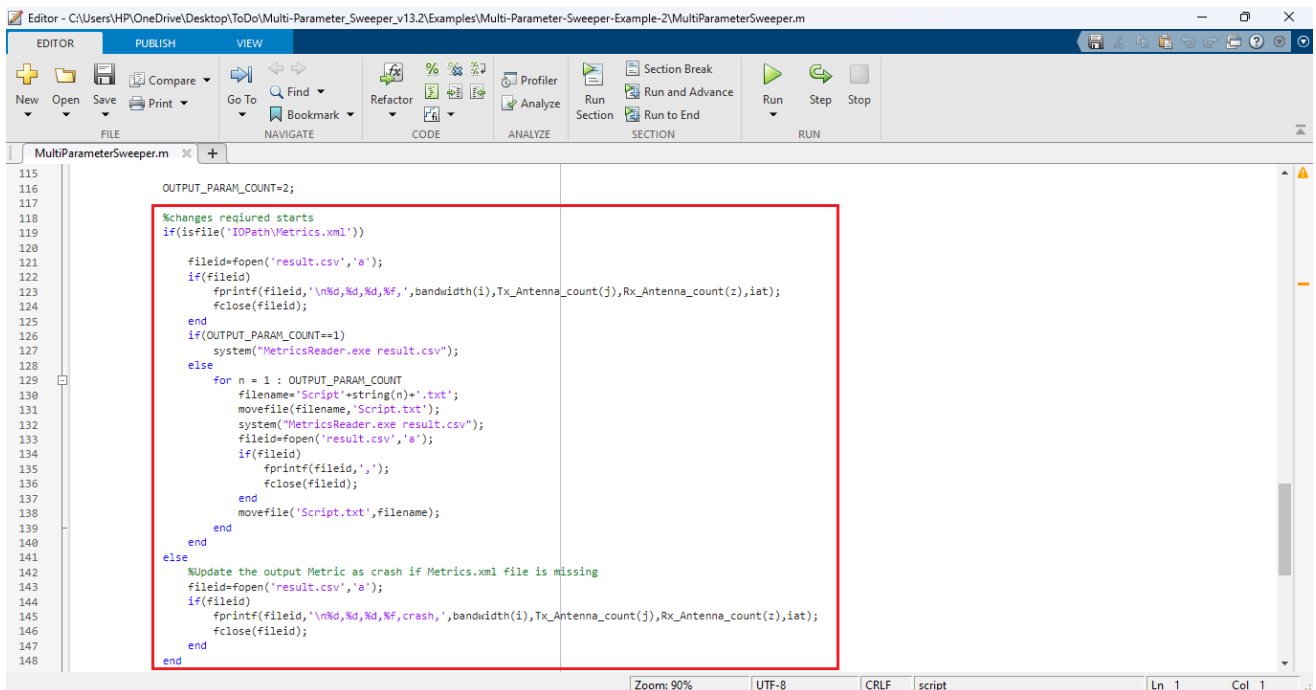
Script2.txt



```
Script2.txt - Notepad
File Edit Format View Help
MENU NAME="Link_Metrics"
TABLE NAME="Link_Metrics"
SET A="Packets transmitted\\Data" WHERE "Link_id"="All"
```

Figure 31: The Data Packets transmitted is to be logged for each simulation modified in Script2.txt

14. In the MATLAB script MultiParameterSweeper.m, MetricsReader is called to log each parameter specified in the script text files separating the entries with a comma (","). If simulation crashes, without generating the output Metrics.xml, then "crash" message is written to the log for each output parameter. The input parameters that were varied during each simulation run are also logged in the results.csv file.



```
Editor - C:\Users\HP\OneDrive\Desktop\ToDo\Multi-Parameter_Sweeper_v13.2\Examples\Multi-Parameter-Sweeper-Example-2\MultiParameterSweeper.m
EDITOR PUBLISH VIEW
New Open Save Compare Go To Find Refactor Profiler Run Section Run and Advance Run Step Stop
FILE NAVIGATE CODE ANALYZE SECTION RUN
MultiParameterSweeper.m
115 OUTPUT_PARAM_COUNT=2;
116
117 %changes required starts
118 if(isfile('IOPath\Metrics.xml'))
119
120     fileid=fopen('result.csv','a');
121     if(fileid)
122         fprintf(fileid,'\n%d,%d,%d,%f','bandwidth(i),Tx_Antenna_count(j),Rx_Antenna_count(z),iat);
123         fclose(fileid);
124     end
125
126     if(OUTPUT_PARAM_COUNT==1)
127         system('MetricsReader.exe result.csv');
128     else
129         for n = 1 : OUTPUT_PARAM_COUNT
130             filename='Script'+string(n)+'.txt';
131             movefile(filename,'Script.txt');
132             system('MetricsReader.exe result.csv');
133             fileid=fopen('result.csv','a');
134             if(fileid)
135                 fprintf(fileid,',');
136                 fclose(fileid);
137             end
138             movefile('Script.txt',filename);
139         end
140     end
141
142     %Update the output Metric as crash if Metrics.xml file is missing
143     fileid=fopen('result.csv','a');
144     if(fileid)
145         fprintf(fileid,'\n%d,%d,%d,%f,crash','bandwidth(i),Tx_Antenna_count(j),Rx_Antenna_count(z),iat);
146         fclose(fileid);
147     end
148 end
```

Figure 32: Modify MATLAB script MultiParameterSweeper.m file

15. The simulation Configuration file and all the output files associated with each simulation run is saved to folders with name including the bandwidth and DL MIMO count values that were used during each simulation run.

16. Multi-Parameter Sweeping process can be started from MATLAB command line in the directory of the Multi-Parameter-Sweeping project as shown below:

The screenshot shows the MATLAB R2022b interface. The top menu bar includes 'HOME', 'PLOTS', and 'APPS'. Below the menu bar are several toolbars: 'Script' (New, Live Script), 'File' (New, Open, Compare), 'Variable' (Import Data, Clean Data, Clear Workspace), 'Code' (Analyze Code, Run and Time, Clear Commands), and 'Simulink'. The 'Current Folder' pane on the left shows a list of files, with 'MultiParameterSweeper.m' selected. The 'Command Window' on the right displays the command 'MultiParameterSweeper' entered at the prompt '>>'.

This starts the Multi-Parameter-Sweeping process which runs NetSim simulations iteratively for different combinations of input parameters.

At the end of the process the Multi-Parameter-Sweeping folder will have the following file and folders created:

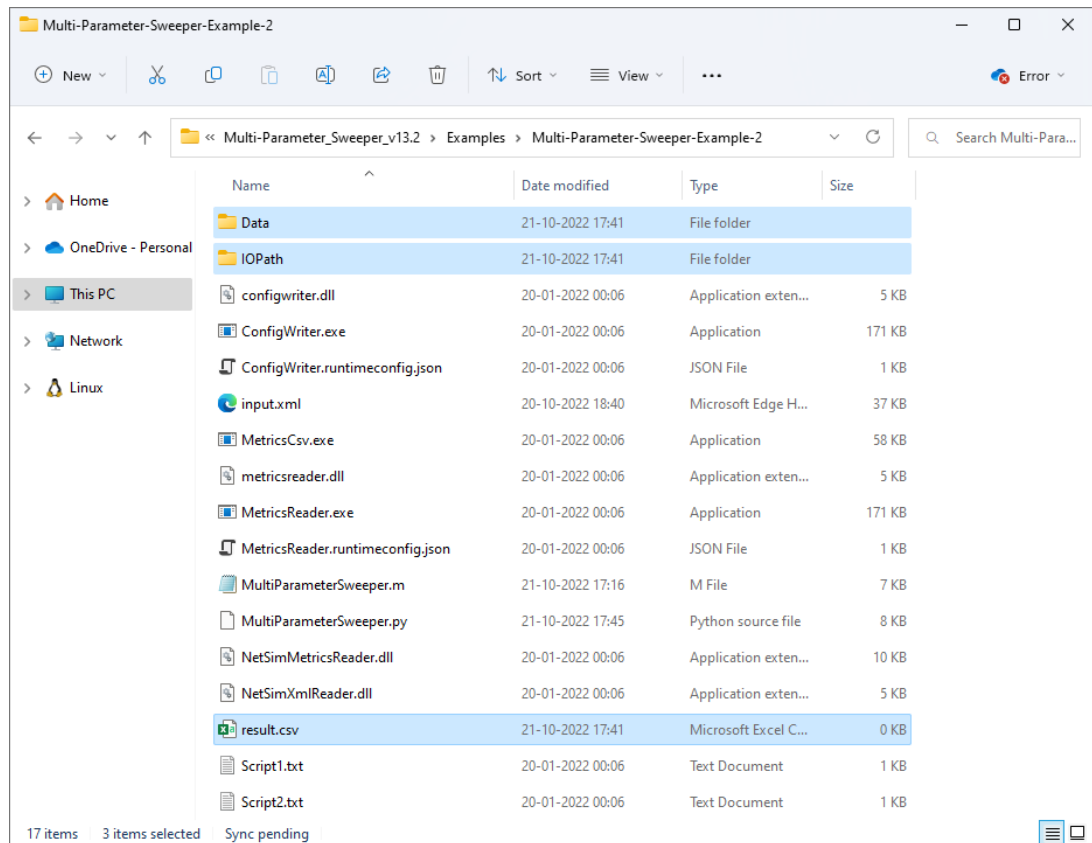


Figure 35: After Simulation Multi-Parameter-Sweeping folder contains output files like result.csv, Data etc

17. **Data:** The Data directory contains multiple output folders with the output files associated with each simulation run.

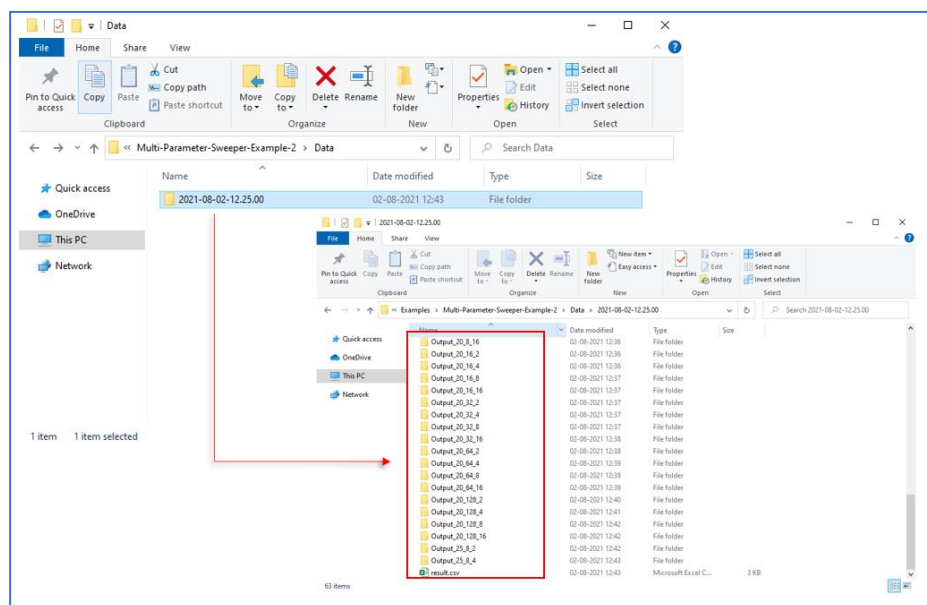
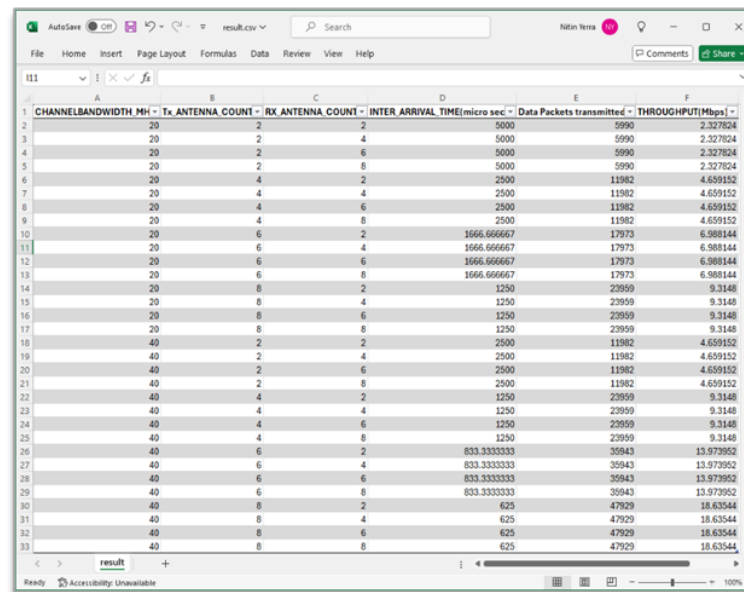


Figure 36: Based on input parameter Configuration.netsim and other files created in output folder

18. **IOPath:** Used for storing the Configuration.netsim file and the simulation files generated during each simulation run.

19. **Result.csv:** This is the output log which contains the parameter varied during each simulation run and the output parameter associated with each run.



CHANNELBANDWIDTH MHz	Tx ANTENNA COUNT	RX ANTENNA COUNT	INTER ARRIVAL TIME[micro sec]	Data Packets transmitted	THROUGHPUT[Mbps]
20	2	2	5000	5990	2.327824
20	2	4	5000	5990	2.327824
20	2	6	5000	5990	2.327824
20	2	8	5000	5990	2.327824
20	4	2	2500	11982	4.659152
20	4	4	2500	11982	4.659152
20	4	6	2500	11982	4.659152
20	4	8	2500	11982	4.659152
20	6	2	1666.666667	17973	6.988144
20	6	4	1666.666667	17973	6.988144
20	6	6	1666.666667	17973	6.988144
20	6	8	1666.666667	17973	6.988144
20	8	2	1250	23959	9.3148
20	8	4	1250	23959	9.3148
20	8	6	1250	23959	9.3148
20	8	8	1250	23959	9.3148
40	2	2	2500	11982	4.659152
40	2	4	2500	11982	4.659152
40	2	6	2500	11982	4.659152
40	2	8	2500	11982	4.659152
40	4	2	1250	23959	9.3148
40	4	4	1250	23959	9.3148
40	4	6	1250	23959	9.3148
40	4	8	1250	23959	9.3148
40	6	2	833.333333	35843	13.973952
40	6	4	833.333333	35843	13.973952
40	6	6	833.333333	35843	13.973952
40	6	8	833.333333	35843	13.973952
40	8	2	625	47929	18.63544
40	8	4	625	47929	18.63544
40	8	6	625	47929	18.63544
40	8	8	625	47929	18.63544

Figure 37: Based on script result stored in result.csv file

Advanced use cases:

- Running the sweeper with binaries of a modified workspace.

When the source codes of a workspace are modified, a warning message will be displayed in the simulation console waiting for user interrupt, as shown below:

```
C:\Users\HP\OneDrive\Documents\NetSim\Workspaces\Default_Workspace_v13_2_34\bin_x64\NetSimCore.exe

Today's date is "Oct 22 2022.14:56:23"
Binary build date is "Oct 13 2022.20:14:17"

NetSim License Manager will first check for node lock licenses.
If not available, it will then check for floating/cloud licenses
NetSim License Manager Start. Checking for licenses available (this may take upto 2 min) -
No license for product (-1)

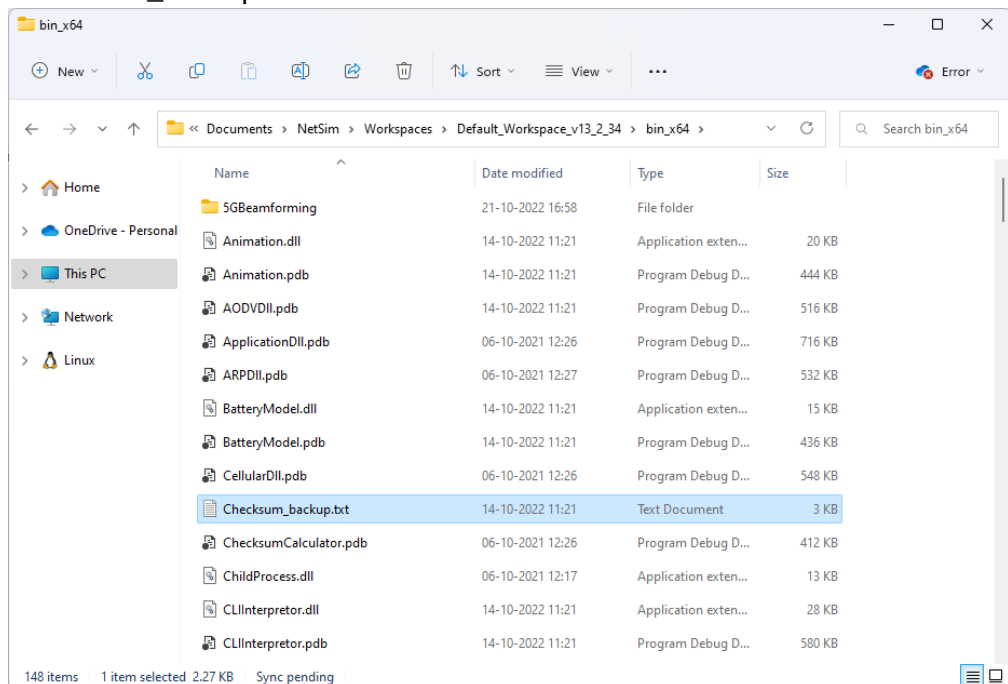
NetSim License Manager Start. Checking for licenses available (this may take upto 2 min) -

License Manager Output. Product>Edition>Maj_ver>Min_ver>Lic_type>Components>
netsim>std>13>2>rlm_hw>111111111111>10000>
+ [33m-[1m
*****
WARNING:
Detected a change in following:
libLTE_NR.dll
This message is normally shown if users link their own code to NetSim.
*****
Press any key to continue....
```

To suppress this message and user interrupt, so that the sweeper runs without requiring any intervention, follow the steps given below:

- Open the current workspace location by going to Your Work-> Workspaces-> Open Workspace Location.

- Go the bin_x64 directory and rename the Checksum.txt file to say, Checksum_backup.txt as shown below:



Now upon running any further simulations, warning messages will not be printed in the simulation console and no user intervention will be required.

2. Sweeping Configuration files that has other associated files

There can be cases where a Configuration file requires other supporting files such as:

- Mobility – Mobility text file.
- Static route text files
- ACL input
- SUMO configuration files in case of VANET, etc

In such cases, the associated files can also be placed in the sweeper folder which contains the input.xml file and code can be slightly modified to copy all associated files when copying the Configuration file to the IOPath.