Automatic plotting of DTDMA parameters in NetSim

Software: NetSim Standard/Pro v13.0 (32-bit/ 64-bit), Visual Studio 2017/2019.

Project Download Link:

https://github.com/NetSim-TETCOS/DTDMA_Radio_Measurements/archive/refs/heads/main.zip

Follow the instructions specified in the following link to download and setup the Project in NetSim:

https://support.tetcos.com/en/support/solutions/articles/14000128666-downloading-and-setting-up-netsim-file-exchange-projects

Features

Using this workspace:

1. Users can plot Pathloss, Shadow Loss, Fading Loss, Total Loss, Rx_Power, SNR, and BER vs. time using NetSim Plot.

2. Users can log Pathloss, Shadow Loss, Fading Loss, Total Loss, Rx_Power, SNR, and BER vs. time using NetSim Plot., with time stamps, to a CSV log file.

3. Users need to provide a file-based input (per a certain format) at the start of simulation for the parameters to be plotted or logged.

4. The plots are unique to Each Wireless Nodes pair.

5. The log entries are unique to Each Wireless Nodes pair.

6. Parameters are logged every slot time (1ms) and plotted.

7. There is no restriction in NetSim on the number of Wireless Nodes in the network.

Example

In the below scenario

- Two Wireless Nodes are dropped on the grid and both Fast Fading and ShadowFadingLoss are enabled.
- Wireless Node-2 moves in a straight line away from the Wireless Node_1.
- The network is simulated for 50 s.

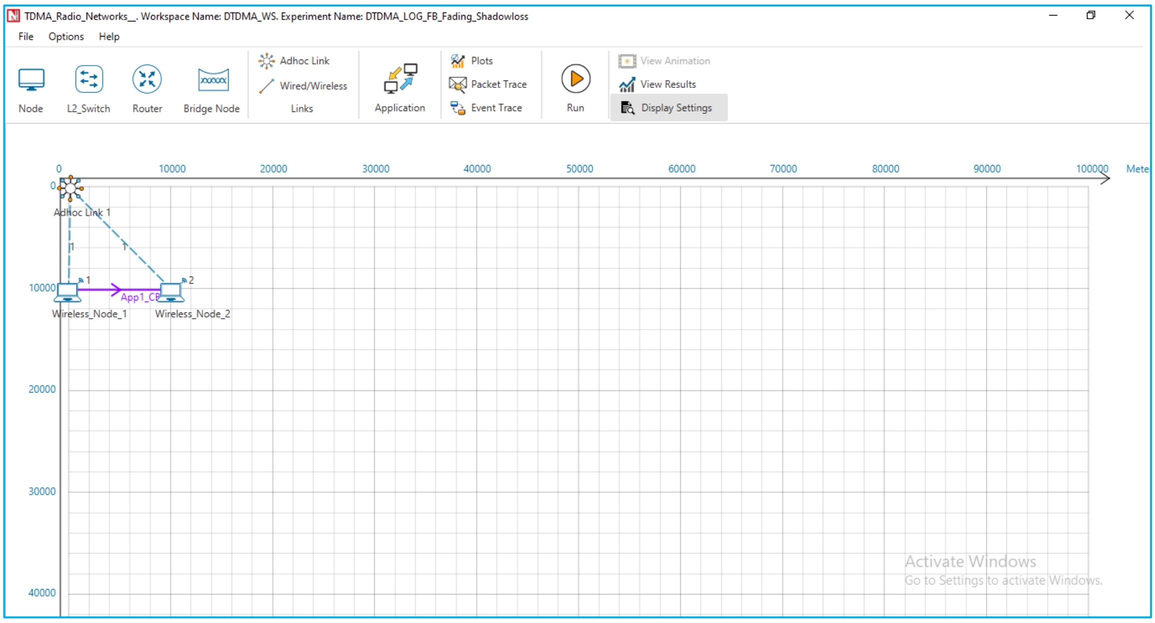


Figure 1: Network Topology in this experiment

- Upon running the simulation, a text file will open for the user to input the parameters and devices (tx-rx pair) for which parameters need to be logged and plotted. The input is per the format of <parameter>, <device1>, <device2> in the text file. To log Wireless Node data flow would be between two Wireless Nodes. In case of multiple Nodes, this input can be given for various Wireless node pairs. Inputs are not case sensitive.

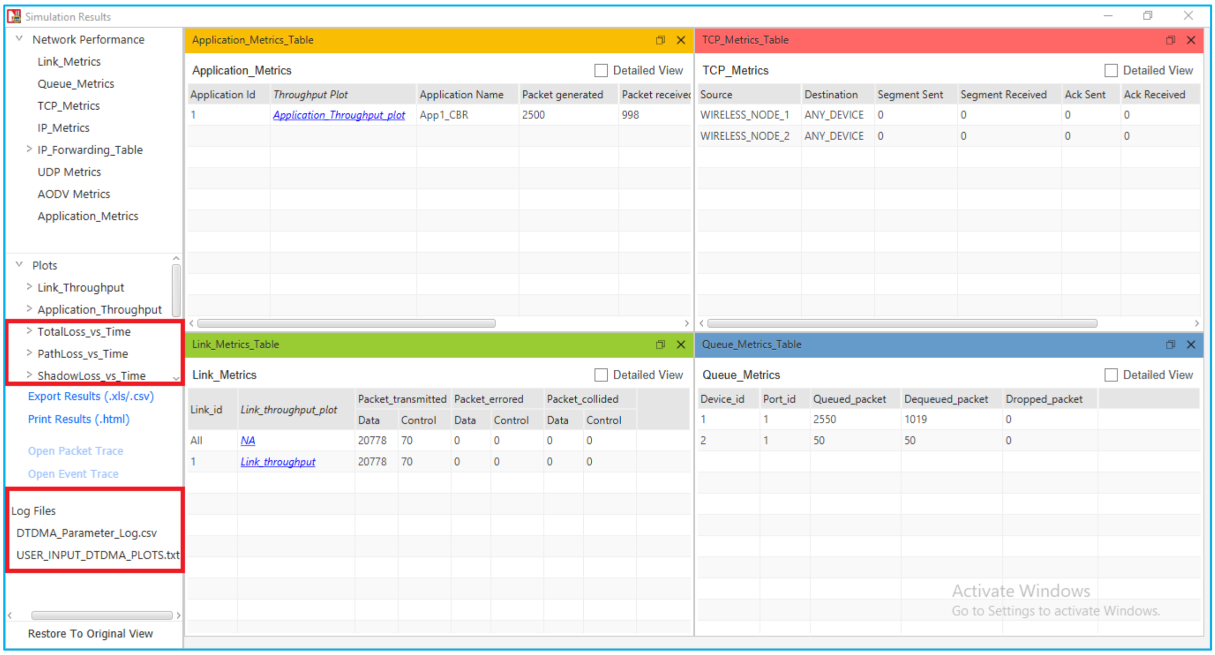
- For the above example, the input text file is as follows.

  TOTALLOSS, Wireless_Node_1, Wireless_Node_2
  PATHLOSS, Wireless_Node_1, Wireless_Node_2
  SHADOWLOSS, Wireless_Node_1, Wireless_Node_2
  FADINGLOSS, Wireless_Node_1, Wireless_Node_2
  SNR, Wireless_Node_1, Wireless_Node_2
  RX_POWER, Wireless_Node_1, Wireless_Node_2
  BER, Wireless_Node_1, Wireless_Node_2

- Once the simulation starts, in the command prompt window it will show a message as "**Please update, Save and close the file and press any key to continue**".
- Add the parameters to be logged, close the input text file and press any key.
- Simulation starts running.

Results and discussion

Upon completion of simulation in the result window users can view the various plots.



Application_Metrics

Application Id	Throughput Plot	Application Name	Packet generated	Packet receivec
1	Application_Throughput_plot	App1_CBR	2500	998

TCP_Metrics

Source	Destination	Segment Sent	Segment Received	Ack Sent	Ack Received
WIRELESS_NODE_1	ANY_DEVICE	0	0	0	0
WIRELESS_NODE_2	ANY_DEVICE	0	0	0	0

Link_Metrics

Link_id	Link_throughput_plot	Packet_transmitted		Packet_errored		Packet_collided	
		Data	Control	Data	Control	Data	Control
All	NA	20778	70	0	0	0	0
1	Link_throughput	20778	70	0	0	0	0

Queue_Metrics

Device_id	Port_id	Queued_packet	Dequeued_packet	Dropped_packet
1	1	2550	1019	0
2	1	50	50	0

Figure 2: NetSim results dashboard with throughput highlighted

The Pathloss, shadow fading loss, and total loss remains same across the layers

Result Plots

1. Pathloss Plot

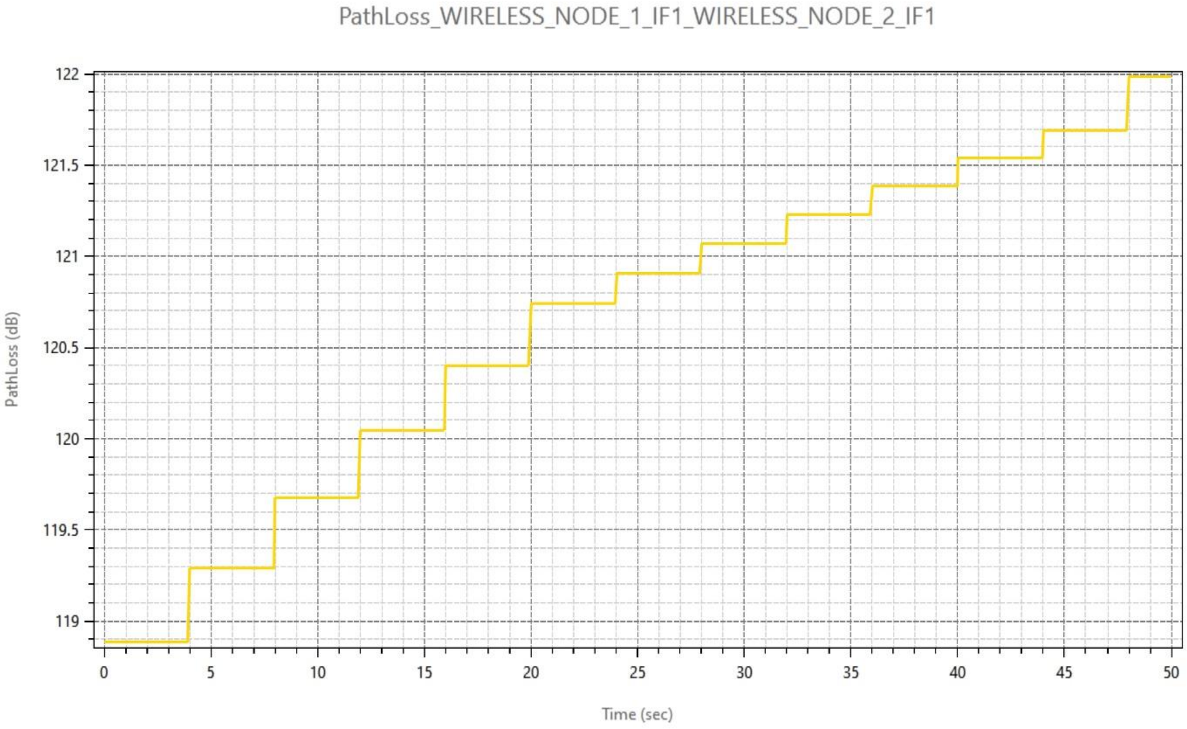


Figure 3: Pathloss Plot in NetSim

2. Total Loss (Shadow Fading loss plus Path loss)

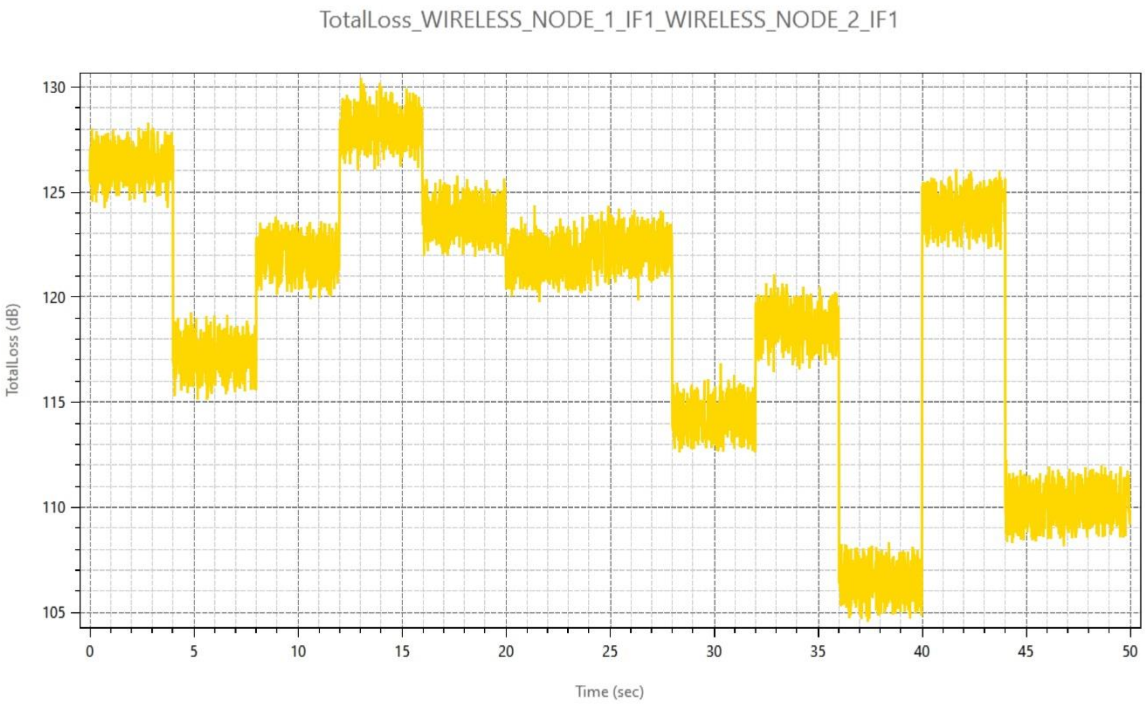


Figure 4: Total Loss (Shadow Fading loss plus Path loss) in NetSim

3. Shadow Loss

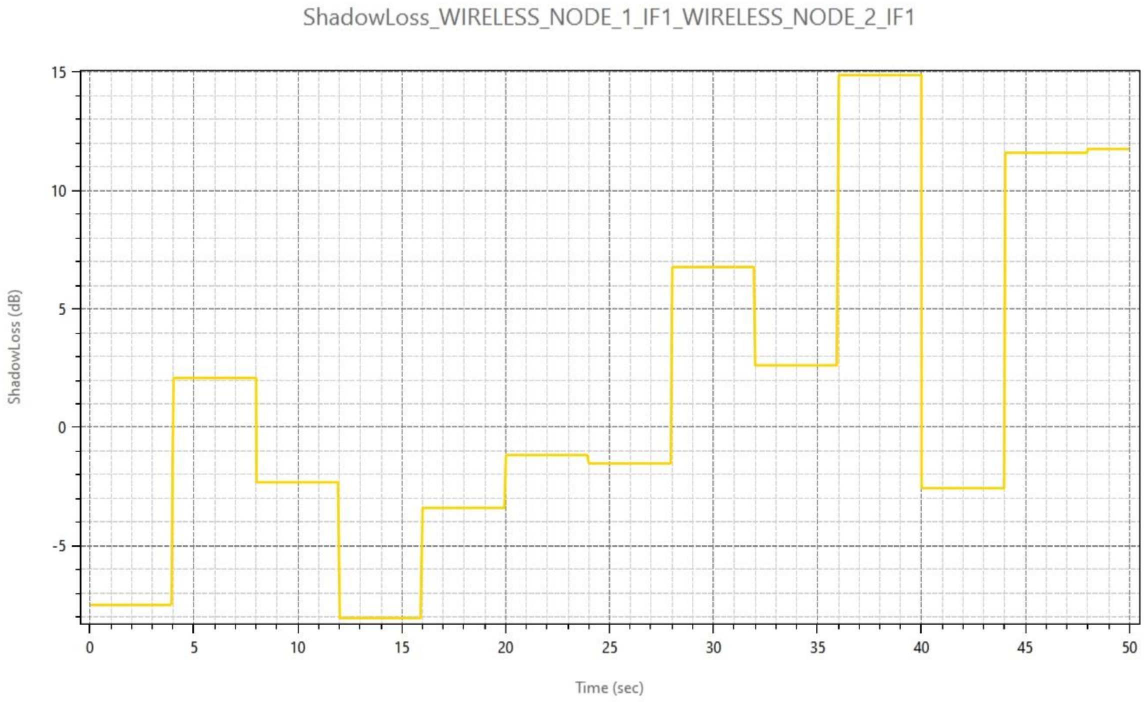


Figure 5: Shadow Fading Loss in NetSim

4. **Fading Loss.**

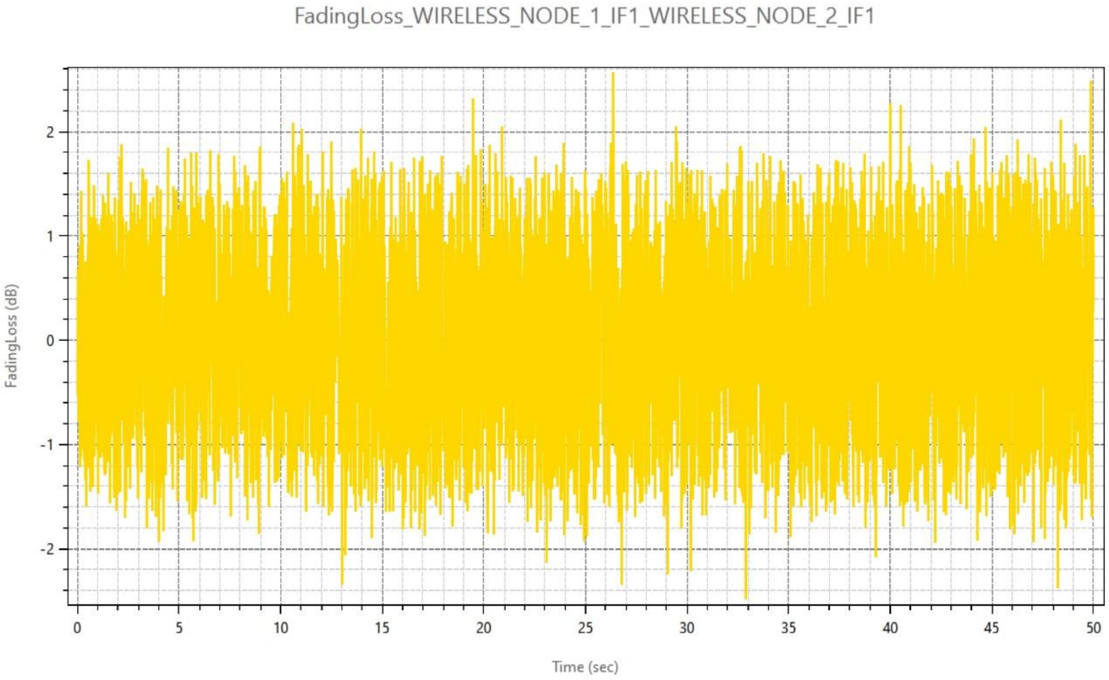


Figure 6: Fading Loss

The plot title is FadingLoss_WIRELESS_NODE_1_IF1_WIRELESS_NODE_2_IF1. And the naming convention is <ParameterType>_WIRELESS_NODE_<ID>_IF<InterfaceID>_WIRELESS_NODE_<ID>_IF <InterfaceID>

5. **Rx_Power Plot**

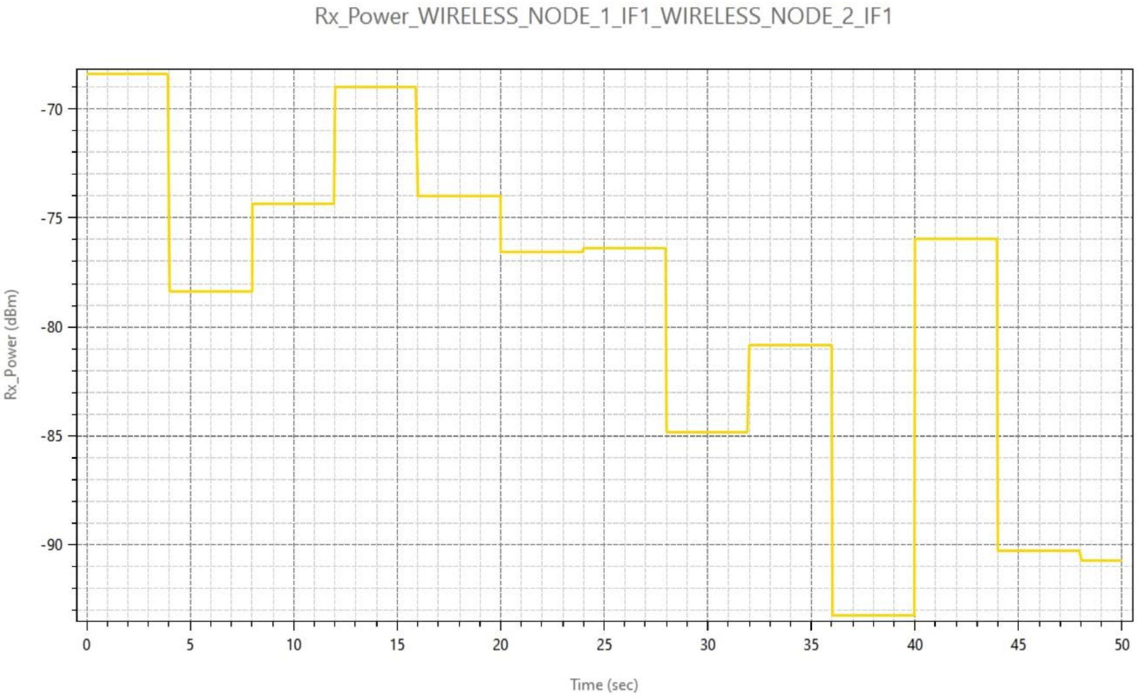


Figure 6: Rx_Power Plot in NetSim.

6. SNR Plot

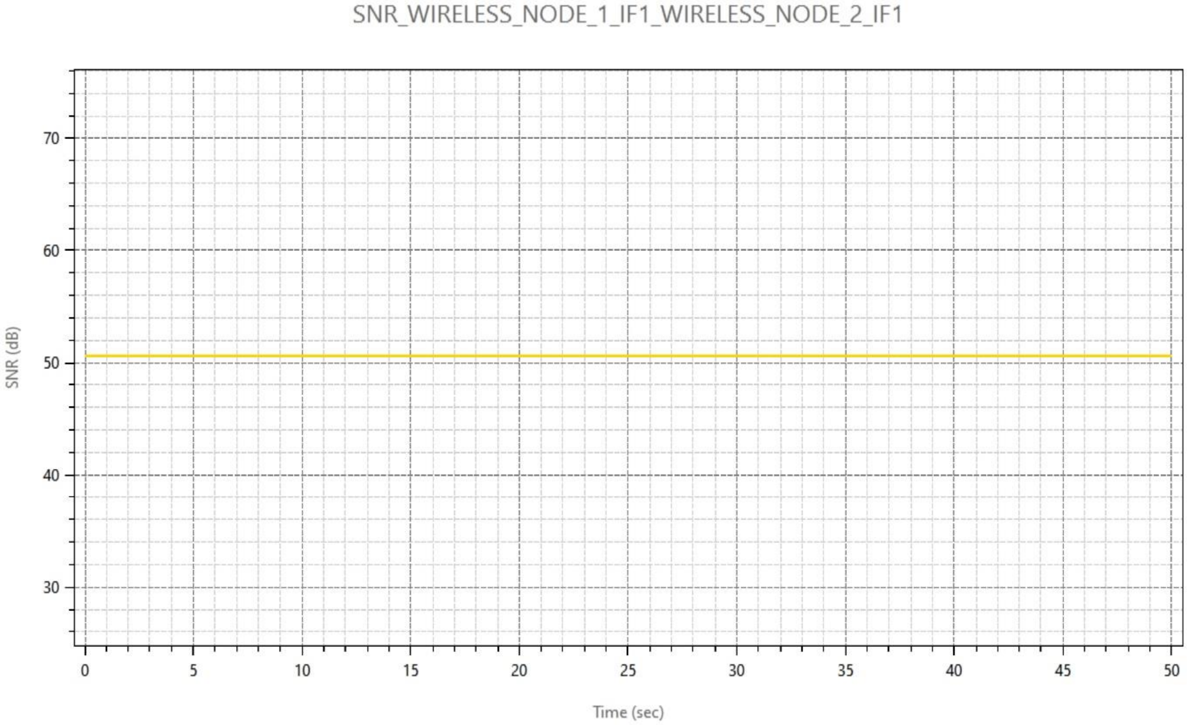
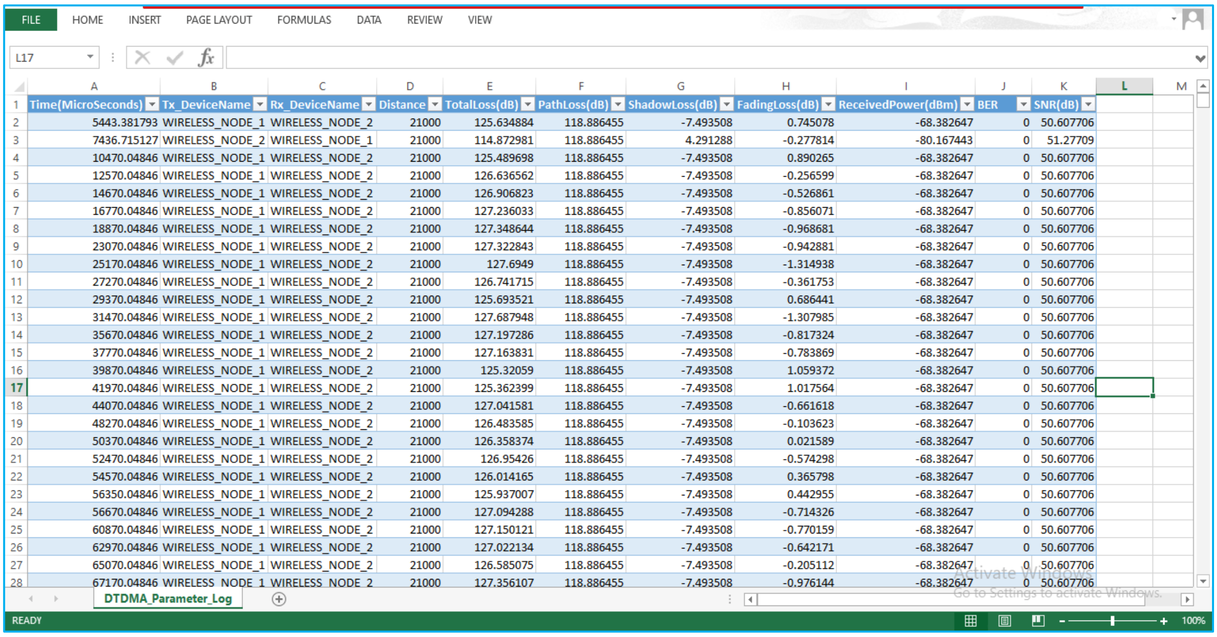


Figure 7: SNR Plot in NetSim

Parameter log file:



Time(MicroSeconds)	Tx_DeviceName	Rx_DeviceName	Distance	TotalLoss(dB)	PathLoss(dB)	ShadowLoss(dB)	FadingLoss(dB)	ReceivedPower(dBm)	BER	SNR(dB)
5443.381793	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	125.634884	118.886455	-7.493508	0.745078	-68.382647	0	50.607706
7436.715127	WIRELESS_NODE_2	WIRELESS_NODE_1	21000	114.872981	118.886455	4.291288	-0.277814	-80.167443	0	51.27709
10470.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	125.489698	118.886455	-7.493508	0.890265	-68.382647	0	50.607706
12570.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	126.636562	118.886455	-7.493508	-0.256599	-68.382647	0	50.607706
14670.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	126.906823	118.886455	-7.493508	-0.526861	-68.382647	0	50.607706
16770.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.236033	118.886455	-7.493508	-0.856071	-68.382647	0	50.607706
18870.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.348644	118.886455	-7.493508	-0.968681	-68.382647	0	50.607706
23070.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.322843	118.886455	-7.493508	-0.942881	-68.382647	0	50.607706
25170.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.6949	118.886455	-7.493508	-1.314938	-68.382647	0	50.607706
27270.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	126.741715	118.886455	-7.493508	-0.361753	-68.382647	0	50.607706
29370.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	125.693521	118.886455	-7.493508	0.686441	-68.382647	0	50.607706
31470.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.687948	118.886455	-7.493508	-1.307985	-68.382647	0	50.607706
35670.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.197286	118.886455	-7.493508	-0.817324	-68.382647	0	50.607706
37770.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.163831	118.886455	-7.493508	-0.783869	-68.382647	0	50.607706
39870.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	125.32059	118.886455	-7.493508	1.059372	-68.382647	0	50.607706
41970.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	125.362399	118.886455	-7.493508	1.017564	-68.382647	0	50.607706
44070.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.041581	118.886455	-7.493508	-0.661618	-68.382647	0	50.607706
48270.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	126.483585	118.886455	-7.493508	-0.103623	-68.382647	0	50.607706
50370.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	126.358374	118.886455	-7.493508	0.021589	-68.382647	0	50.607706
52470.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	126.95426	118.886455	-7.493508	-0.574298	-68.382647	0	50.607706
54570.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	126.014165	118.886455	-7.493508	0.365798	-68.382647	0	50.607706
56350.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	125.937007	118.886455	-7.493508	0.442955	-68.382647	0	50.607706
56670.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.094288	118.886455	-7.493508	-0.714326	-68.382647	0	50.607706
60870.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.150121	118.886455	-7.493508	-0.770159	-68.382647	0	50.607706
62970.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.022134	118.886455	-7.493508	-0.642171	-68.382647	0	50.607706
65070.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	126.585075	118.886455	-7.493508	-0.205112	-68.382647	0	50.607706
67170.04846	WIRELESS_NODE_1	WIRELESS_NODE_2	21000	127.356107	118.886455	-7.493508	-0.976144	-68.382647	0	50.607706

Figure 7: DTDMA Log file parameter

The DTDMA_Parameter_Log.csv file logs the details of parameters specified in the input file with respect to time.

Appendix: NetSim source code modifications

Open the Source codes in Visual Studio by going to your work-> Workspace Options and Clicking on Open code button.

To the in DTDMA project, DTDMA.c file contain the definitions of the functions that responsible for plotting and logging parameters associated with DTDMA network in NetSim.

The function fn_NetSim_DTDMA_Init_Plots and fn_NetSim_DTDMA_init_Parameter_Log has been called in DTDMA.c file for initializing the plot.

```
int fn_NetSim_DTDMA_Mobility(NETSIM_ID nNodeId);
double codingrate_to_double_from_string(char* s);
static bool isplotinit = false;
/**
DTDMA Init function initializes the DTDMA parameters.
*/
_declspec (dllexport) int fn_NetSim_DTDMA_Init(struct stru_NetSim_Network *NETWORK_Formal,
    NetSim_EVENTDETAILS *pstruEventDetails_Formal,
    char *pszAppPath_Formal,
    char *pszWritePath_Formal,
    int nVersion_Type,
    void **fnPointer)
{

    if (!isplotinit)
    {
       fn_NetSim_DTDMA_Init_Plots ();
    fn_NetSim_DTDMA_init_Parameter_Log ();

        isplotinit = true;
    }
    fn_NetSim_DTDMA_NodeInit();
    fn_NetSim_DTDMA_CalulateReceivedPower ();
    init_dtdma_session();
    init_slot_formation();
    fn_NetSim_DTDMA_InitFrequencyHopping ();
    fnMobilityRegisterCallBackFunction (fn_NetSim_DTDMA_Mobility);
    fnNodeJoinRegisterCallBackFunction (fnDTDMANodeJoinCallBack);
    return 0;
}
```

The initialization of functions and functions to update the logs for plotting and logging to CSV file has been made as follows in fn_NetSim_DTDMA_Run() function and fn_NetSim_DTDMA_CalulateReceivedPower() function DTDMA_Phy.c file

```
case PHYSICAL_OUT_EVENT:
        {
            fn_NetSim_DTDMA_PhysicalOut();
        }
        break;
    case PHYSICAL_IN_EVENT:
        pstruEventDetails->pPacket->pstruPhyData->nPacketErrorFlag =
        fn_NetSim_DTDMA_CalculatePacketError(d, in, pstruEventDetails->pPacket);
```

```
//Function calls
                double dThemalNoise= 0; //in dBm
                double dBandwidth;


double dRx_Power = GET_RX_POWER_dbm(pstruEventDetails->nDeviceId, pstruEventDetails-
>nInterfaceId,d, in);


DTDMA_NODE_PHY* phy = DTDMA_PHY(pstruEventDetails->pPacket->nTransmitterId, in);
                PPROPAGATION_INFO**** info = propagationHandle;

PPROPAGATION_INFO pinfo = info [pstruEventDetails->pPacket-
>nTransmitterId][in][pstruEventDetails->nDeviceId][pstruEventDetails->nInterfaceId];

double fading = propagation_calculate_fadingloss (propagationHandle,pstruEventDetails->pPacket-
>nTransmitterId,in,pstruEventDetails->nDeviceId, pstruEventDetails->nInterfaceId);


double ber= calculate_BER(phy->modulation,GET_RX_POWER_dbm(pstruEventDetails->pPacket-
>nTransmitterId, pstruEventDetails->nInterfaceId,d, in),
                        phy->dBandwidth);

double snr= dRx_Power - dThemalNoise;


                fn_NetSim_DTDMA_Log_Parameters(pinfo, fading, snr, ber);
                fn_NetSim_DTDMA_add_PropagationInfo_Plot_data(pinfo, fading, snr, ber);
                fn_NetSim_DTDMA_add_Power_Plot_data(pinfo, snr, ber);



pstruEventDetails->pPacket->nPacketStatus = pstruEventDetails->pPacket->pstruPhyData-
>nPacketErrorFlag;
                fn_NetSim_Metrics_Add(pstruEventDetails->pPacket);
                fn_NetSim_WritePacketTrace(pstruEventDetails->pPacket);
                if(pstruEventDetails->pPacket->nPacketStatus == PacketStatus_NoError)
                {
                        pstruEventDetails->nEventType=MAC_IN_EVENT;
                        fnpAddEvent(pstruEventDetails);
                }
                else
                {
                        fn_NetSim_Packet_FreePacket(pstruEventDetails->pPacket);
                }
        break;
```

DTDMA_Phy.c

```
NETSIM_ID t, ti, r, ri;

        propagationHandle = propagation_init(MAC_PROTOCOL_DTDMA, NULL,DTDMA_gettxinfo,
check_interference);


        parameter_plot_info = (ptrplotINFO****)calloc(NETWORK->nDeviceCount + 1, sizeof *
parameter_plot_info);
```

```
    for (t = 0; t < NETWORK->nDeviceCount; t++)
    {

        parameter_plot_info[t + 1] = (ptrplotINFO***)calloc(DEVICE(t + 1)-
>nNumOfInterface+1, sizeof * parameter_plot_info[t + 1]);
        for (ti = 0; ti < DEVICE(t + 1)->nNumOfInterface; ti++)
        {
            if (!isDTDMAConfigured(t + 1, ti + 1))
                continue;

            parameter_plot_info[t + 1][ti + 1] = (ptrplotINFO**)calloc(NETWORK-
>nDeviceCount+1, sizeof * parameter_plot_info[t + 1][ti + 1]);
            for (r = 0; r < NETWORK->nDeviceCount; r++)
            {
                parameter_plot_info[t + 1][ti + 1][r + 1] =
(ptrplotINFO*)calloc(DEVICE(r + 1)->nNumOfInterface+1, sizeof * parameter_plot_info[t + 1][ti + 1][r +
1]);
                for (ri = 0; ri < DEVICE(r + 1)->nNumOfInterface; ri++)
                {
                    if (!isDTDMAConfigured(r + 1, ri + 1))
                        continue;
                    parameter_plot_info[t + 1][ti + 1][r + 1][ri + 1] =
(ptrplotINFO)calloc(1, sizeof * parameter_plot_info[t + 1][ti + 1][r + 1][ri + 1]);
                    dtdma_CalculateReceivedPower(t + 1, ti + 1, r + 1, ri + 1);

                    PPROPAGATION_INFO**** info = propagationHandle;
                    PPROPAGATION_INFO pinfo = info[t + 1][ti + 1][r + 1][ri + 1];

                    ptrINFO infolog = parameter_log_info;
                    ptrplotINFO**** infoplot = parameter_plot_info;

                    ptrplotINFO info1 = infoplot[pinfo->nTxId][pinfo-
>nTxInterface][pinfo->nRxId][pinfo->nRxInterface];
//                  infolog->isParameterlog = false;

                    info1->isFadingLossPlotEnable = false;
                    info1->isPathLossPlotEnable = false;
                    info1->isShadowLossPlotEnable = false;
                    info1->isTotalLossPlotEnable = false;
                    info1->isrxPowerPlotEnable = false;
                    fn_NetSim_DTDMA_init_PropagationInfo_Plots(pinfo);
                    fn_NetSim_DTDMA_init_Power_Plots(pinfo);

                }
            }
        }
    }
    return 0;
}
```