Rebroadcasting packet in NetSim MANET\VANETs

Software Used: NetSim Standard v13.0 (32bit/ 64bit), Microsoft Visual Studio2019

Project Download Link:

https://github.com/NetSim-TETCOS/Probability-based-rebroadcast_v13.0/archive/refs/heads/main.zip

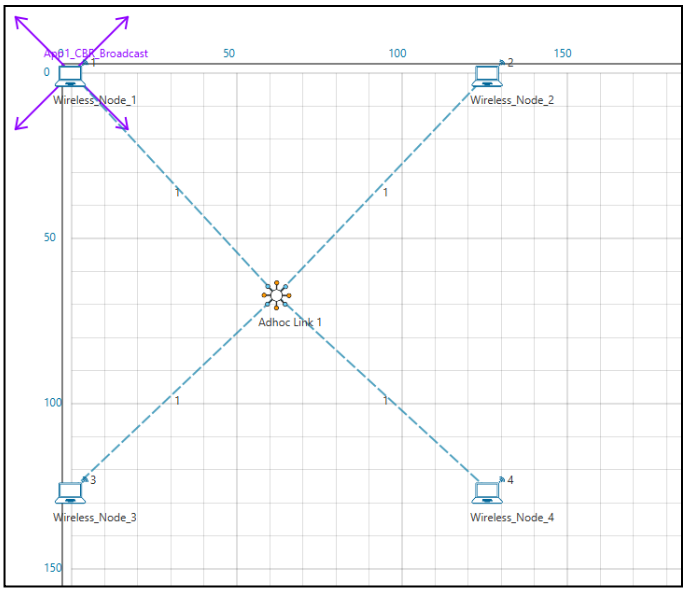
Follow the instructions specified in the following link to download and setup the Project in NetSim:

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

Broadcasting:

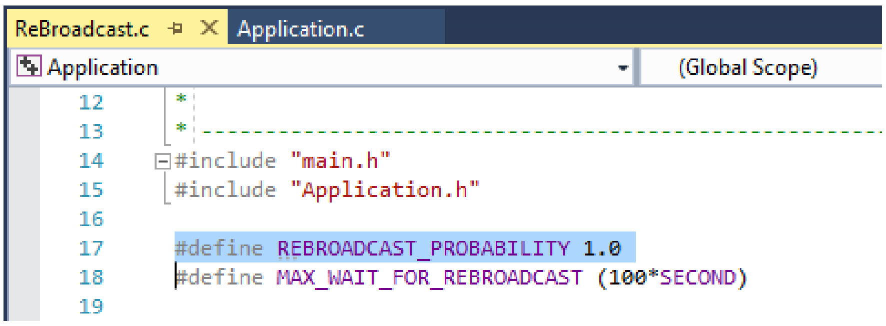
Broadcasting is the process of sending a message from one node to all other nodes in an ad-hoc network. It is a fundamental operation for communication in ad-hoc networks as it allows for the update of network information and route discovery at every node.

Rebroadcasting:



Wireless Node 1 initiates a broadcast message and the message is received by nodes 2, 3 and 4. 2, 3 and 4 rebroadcast the message if they have not broadcasted that before. Furthermore, this implementation involves a Rebroadcast_Probability based on which the nodes resend the packets.

Probability-based rebroadcasting - The decision of rebroadcasting is based upon a random probability. This probability may be as simple as flipping a coin or it may be very complex involving probabilities which include parameters such as node density, duplicate packets received, battery power or a particular nodes participation within the network etc. Users can change the Rebroadcast_Probability macros present in Rebroadcast.c file as shown below:



```
 *
 * ---------------------------------------------------------------
#include "main.h"
#include "Application.h"

#define REBROADCAST_PROBABILITY 1.0
#define MAX_WAIT_FOR_REBROADCAST (100*SECOND)

```

Rebroadcasting in NetSim:

To implement this project in NetSim, we have created an additional Rebroadcast.c file inside Application project. The file contains the following functions:

- void rebroadcast_packet();

  This function is used to rebroadcast the packet.

- static bool isRebroadcastAllowed();

  This function is used to check whether rebroadcasting is allowed or not.

- void rebroadcast_add_packet_to_info();

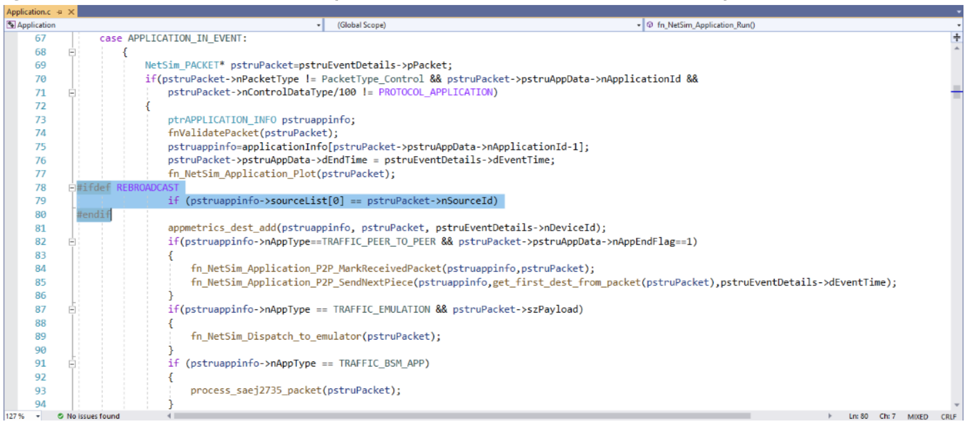
  This function is used to add the packet to rebroadcast list.

- static void cleanup_broadcast_info();

  This function is used to clean the broadcast information.

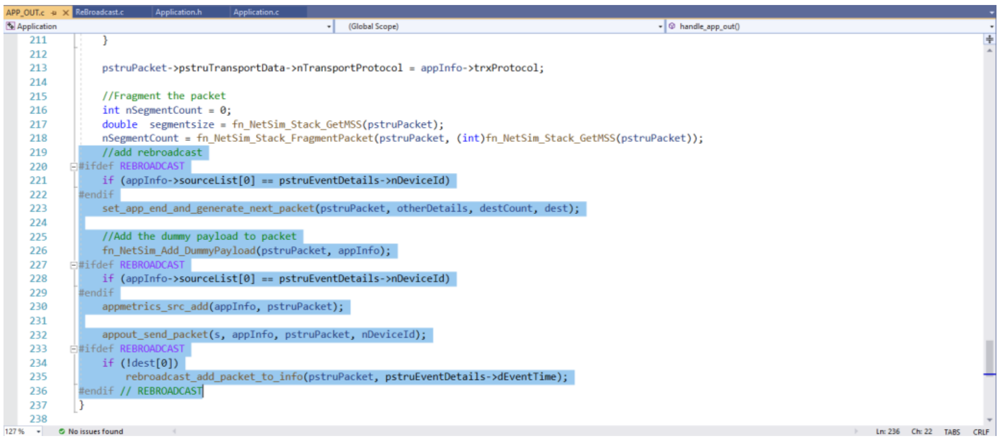
Code modifications done in NetSim:

1. We have added the following lines of code in fn_NetSim_Application_Run() function in the APPLICATION_IN_EVENT present in Application.c file inside Application project. This is used to generate next broadcast packet if the current device is present in the source list.



```
	case APPLICATION_IN_EVENT:
		{
			NetSim_PACKET* pstruPacket=pstruEventDetails->pPacket;
			if(pstruPacket->nPacketType != PacketType_Control && pstruPacket->pstruAppData->nApplicationId &&
				pstruPacket->nControlDataType/100 != PROTOCOL_APPLICATION)
			{
				ptrAPPLICATION_INFO pstruappinfo;
				fnValidatePacket(pstruPacket);
				pstruappinfo=applicationInfo[pstruPacket->pstruAppData->nApplicationId-1];
				pstruPacket->pstruAppData->dEndTime = pstruEventDetails->dEventTime;
				fn_NetSim_Application_Plot(pstruPacket);
#ifdef REBROADCAST
				if (pstruappinfo->sourceList[0] == pstruPacket->nSourceId)
#endif
				appmetrics_dest_add(pstruappinfo, pstruPacket, pstruEventDetails->nDeviceId);
				if(pstruappinfo->nAppType==TRAFFIC_PEER_TO_PEER && pstruPacket->pstruAppData->nAppEndFlag==1)
				{
					fn_NetSim_Application_P2P_MarkReceivedPacket(pstruappinfo,pstruPacket);
					fn_NetSim_Application_P2P_SendNextPiece(pstruappinfo,get_first_dest_from_packet(pstruPacket),pstruEventDetails->dEventTime);
				}
				if(pstruappinfo->nAppType == TRAFFIC_EMULATION && pstruPacket->szPayload)
				{
					fn_NetSim_Dispatch_to_emulator(pstruPacket);
				}
				if (pstruappinfo->nAppType == TRAFFIC_BSM_APP)
				{
					process_saej2735_packet(pstruPacket);
				}
```

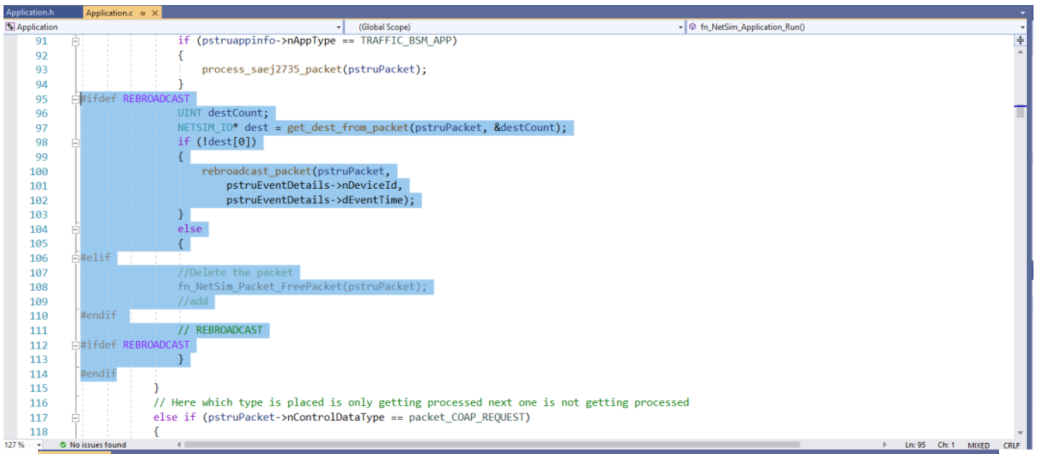
2. The following lines of code are added in the handle_app_out() function present in APP_OUT.c file inside Application project. The code checks if the destination is ‘0’ i.e., Broadcast packet, then it adds the packet to rebroadcast list.



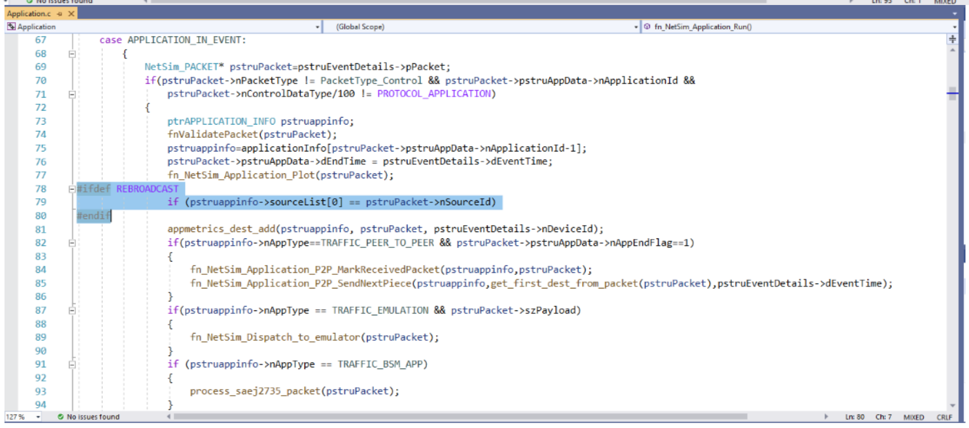
```
	}

	pstruPacket->pstruTransportData->nTransportProtocol = appInfo->trxProtocol;

	//Fragment the packet
	int nSegmentCount = 0;
	double  segmentsize = fn_NetSim_Stack_GetMSS(pstruPacket);
	nSegmentCount = fn_NetSim_Stack_FragmentPacket(pstruPacket, (int)fn_NetSim_Stack_GetMSS(pstruPacket));
	//add rebroadcast
#ifdef REBROADCAST
	if (appInfo->sourceList[0] == pstruEventDetails->nDeviceId)
#endif
	set_app_end_and_generate_next_packet(pstruPacket, otherDetails, destCount, dest);

	//Add the dummy payload to packet
	fn_NetSim_Add_DummyPayload(pstruPacket, appInfo);
#ifdef REBROADCAST
	if (appInfo->sourceList[0] == pstruEventDetails->nDeviceId)
#endif
	appmetrics_src_add(appInfo, pstruPacket);

	appout_send_packet(s, appInfo, pstruPacket, nDeviceId);
#ifdef REBROADCAST
	if (!dest[0])
		rebroadcast_add_packet_to_info(pstruPacket, pstruEventDetails->dEventTime);
#endif // REBROADCAST
}
```

3. Now add the following code in fn_NetSim_Application_Run() function in APPLICATION_IN_EVENT present in Application.c file inside Application project. It checks

whether the destination is ‘0’ or not. If it is ‘0’, then it rebroadcasts the packet or else deletes the packet.

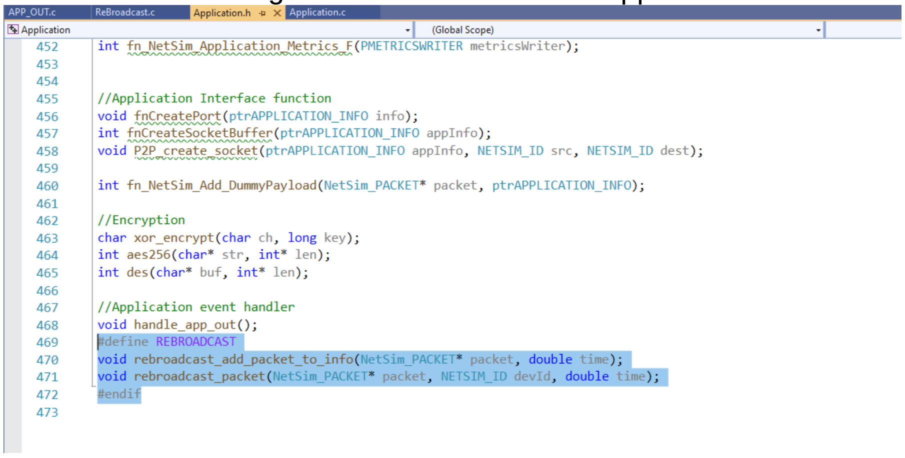


```c
                if (pstruappinfo->nAppType == TRAFFIC_BSM_APP)
                {
                    process_saej2735_packet(pstruPacket);
                }
#ifdef REBROADCAST
                UINT destCount;
                NETSIM_ID* dest = get_dest_from_packet(pstruPacket, &destCount);
                if (!dest[0])
                {
                    rebroadcast_packet(pstruPacket,
                        pstruEventDetails->nDeviceId,
                        pstruEventDetails->dEventTime);
                }
                else
                {
#elif
                //Delete the packet
                fn_NetSim_Packet_FreePacket(pstruPacket);
                //add
#endif
                // REBROADCAST
#ifdef REBROADCAST
                }
#endif
            }
            // Here which type is placed is only getting processed next one is not getting processed
            else if (pstruPacket->nControlDataType == packet_COAP_REQUEST)
            {
```



```c
        case APPLICATION_IN_EVENT:
            {
                NetSim_PACKET* pstruPacket=pstruEventDetails->pPacket;
                if(pstruPacket->nPacketType != PacketType_Control && pstruPacket->pstruAppData->nApplicationId &&
                    pstruPacket->nControlDataType/100 != PROTOCOL_APPLICATION)
                {
                    ptrAPPLICATION_INFO pstruappinfo;
                    fnValidatePacket(pstruPacket);
                    pstruappinfo=applicationInfo[pstruPacket->pstruAppData->nApplicationId-1];
                    pstruPacket->pstruAppData->dEndTime = pstruEventDetails->dEventTime;
                    fn_NetSim_Application_Plot(pstruPacket);
#ifdef REBROADCAST
                    if (pstruappinfo->sourceList[0] == pstruPacket->nSourceId)
#endif
                    appmetrics_dest_add(pstruappinfo, pstruPacket, pstruEventDetails->nDeviceId);
                    if(pstruappinfo->nAppType==TRAFFIC_PEER_TO_PEER && pstruPacket->pstruAppData->nAppEndFlag==1)
                    {
                        fn_NetSim_Application_P2P_MarkReceivedPacket(pstruappinfo,pstruPacket);
                        fn_NetSim_Application_P2P_SendNextPiece(pstruappinfo,get_first_dest_from_packet(pstruPacket),pstruEventDetails->dEventTime);
                    }
                    if(pstruappinfo->nAppType == TRAFFIC_EMULATION && pstruPacket->szPayload)
                    {
                        fn_NetSim_Dispatch_to_emulator(pstruPacket);
                    }
                    if (pstruappinfo->nAppType == TRAFFIC_BSM_APP)
                    {
                        process_saej2735_packet(pstruPacket);
                    }
```

4. We have added the following function declarations in Application.h file.

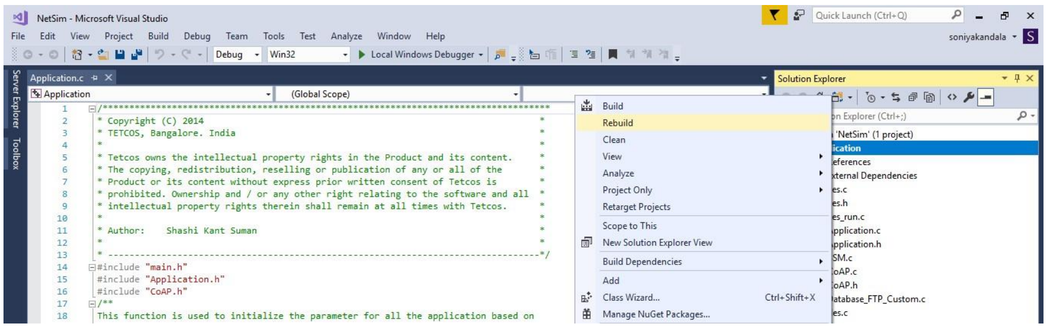


```c
int fn_NetSim_Application_Metrics_F(PMETRICSWRITER metricsWriter);


//Application Interface function
void fnCreatePort(ptrAPPLICATION_INFO info);
int fnCreateSocketBuffer(ptrAPPLICATION_INFO appInfo);
void P2P_create_socket(ptrAPPLICATION_INFO appInfo, NETSIM_ID src, NETSIM_ID dest);

int fn_NetSim_Add_DummyPayload(NetSim_PACKET* packet, ptrAPPLICATION_INFO);

//Encryption
char xor_encrypt(char ch, long key);
int aes256(char* str, int* len);
int des(char* buf, int* len);

//Application event handler
void handle_app_out();
#define REBROADCAST
void rebroadcast_add_packet_to_info(NetSim_PACKET* packet, double time);
void rebroadcast_packet(NetSim_PACKET* packet, NETSIM_ID devId, double time);
#endif
```

Steps:

- Go to home page, Click on **Your work→ Workspace options→Open code**

- Right click on Solution in Solution Explorer and select ‘Rebuild solution’.

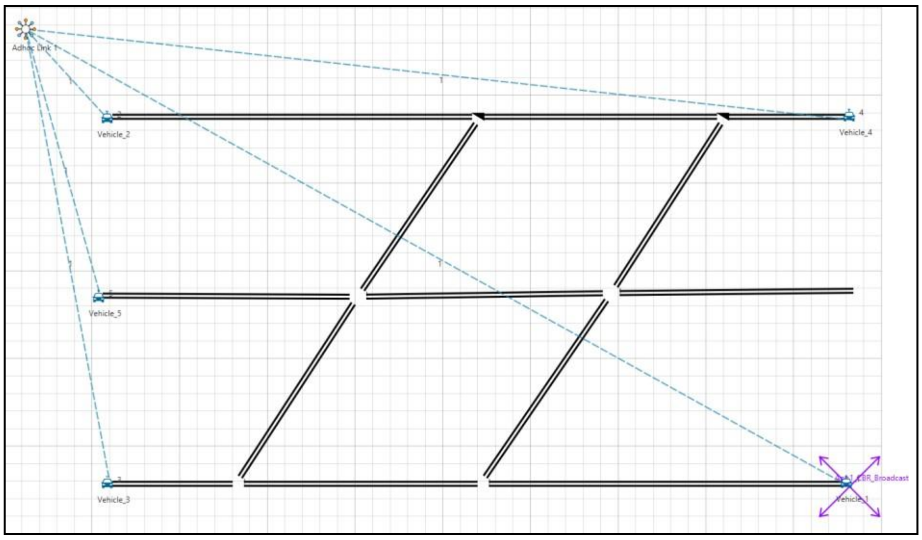

NetSim - Microsoft Visual Studio
File Edit View Project Build Debug Team Tools Test Analyze Window Help
Debug Win32 Local Windows Debugger
Application.c
Application
(Global Scope)
Build
Rebuild
Clean
View
Analyze
Project Only
Retarget Projects
Scope to This
New Solution Explorer View
Build Dependencies
Add
Class Wizard... Ctrl+Shift+X
Manage NuGet Packages...
Solution Explorer
'NetSim' (1 project)

```
/*************************************************************************************
* Copyright (C) 2014                                                                 *
* TETCOS, Bangalore. India                                                           *
*                                                                                    *
* Tetcos owns the intellectual property rights in the Product and its content.      *
* The copying, redistribution, reselling or publication of any or all of the        *
* Product or its content without express prior written consent of Tetcos is         *
* prohibited. Ownership and / or any other right relating to the software and all   *
* intellectual property rights therein shall remain at all times with Tetcos.       *
*                                                                                    *
* Author:    Shashi Kant Suman                                                       *
*                                                                                    *
* ---------------------------------------------------------------------------------*/
#include "main.h"
#include "Application.h"
#include "CoAP.h"
/**
This function is used to initialize the parameter for all the application based on
```

- Upon rebuilding, **libApplication.dll** will automatically get updated in the respective bin folder of the current workspace.

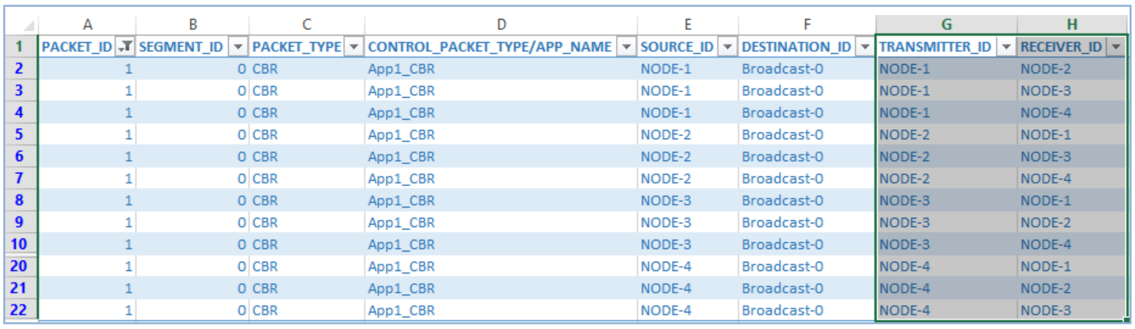
- Go to NetSim home page, click on **Your work**, Click on **Rebroadcasting_VANET_Example/ Rebroadcasting_MANET_Example** and run the simulation for 100 seconds.

VANET SCENARIO:



- In the above scenario, Vehicle-1 is broadcasting the packet and it is received by the Vehicles 2, 3 and 4. Then Vehicles 2, 3, and 4 will rebroadcast the same packet based on the probability value in Rebroadcast.c file.

- After simulation, open Packet Trace and filter Packet_Id to '1' or any other id and observe that the nodes other than source are rebroadcasting the same packet.



	A	B	C	D	E	F	G	H
1	PACKET_ID	SEGMENT_ID	PACKET_TYPE	CONTROL_PACKET_TYPE/APP_NAME	SOURCE_ID	DESTINATION_ID	TRANSMITTER_ID	RECEIVER_ID
2	1	0	CBR	App1_CBR	NODE-1	Broadcast-0	NODE-1	NODE-2
3	1	0	CBR	App1_CBR	NODE-1	Broadcast-0	NODE-1	NODE-3
4	1	0	CBR	App1_CBR	NODE-1	Broadcast-0	NODE-1	NODE-4
5	1	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-1
6	1	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-3
7	1	0	CBR	App1_CBR	NODE-2	Broadcast-0	NODE-2	NODE-4
8	1	0	CBR	App1_CBR	NODE-3	Broadcast-0	NODE-3	NODE-1
9	1	0	CBR	App1_CBR	NODE-3	Broadcast-0	NODE-3	NODE-2
10	1	0	CBR	App1_CBR	NODE-3	Broadcast-0	NODE-3	NODE-4
20	1	0	CBR	App1_CBR	NODE-4	Broadcast-0	NODE-4	NODE-1
21	1	0	CBR	App1_CBR	NODE-4	Broadcast-0	NODE-4	NODE-2
22	1	0	CBR	App1_CBR	NODE-4	Broadcast-0	NODE-4	NODE-3

- Note that Users SHOULD NOT use the performance metrics provided at the end of simulation but should rather calculate the network performance metrics from the packet trace.
- Users can also create their own network scenarios in **Single MANET/VANET** and run the simulation.