Rebroadcasting packet in NetSim MANET\VANETs

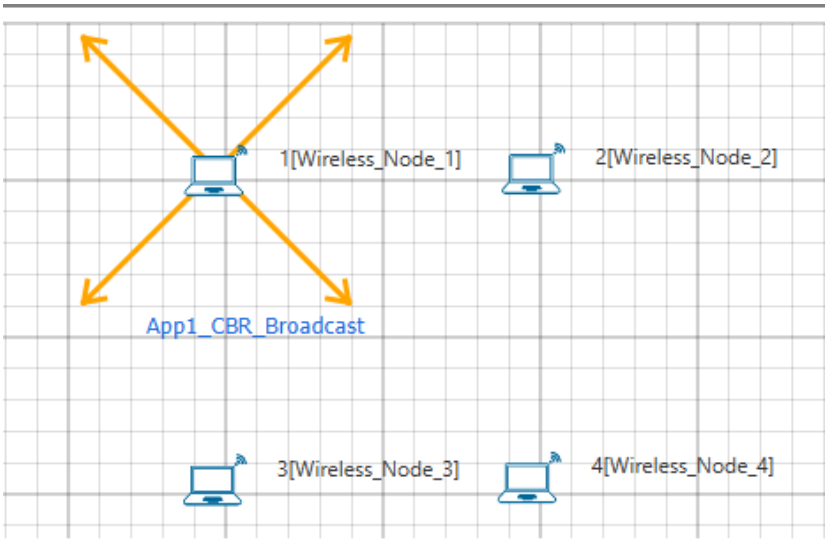
Software Used: NetSim Standard v11.0, Microsoft Visual Studio 2015/2017

Project Download Link: https://github.com/NetSim-TETCOS/Rebroadcasting_in_NetSim_v11.0/archive/master.zip

Broadcasting:

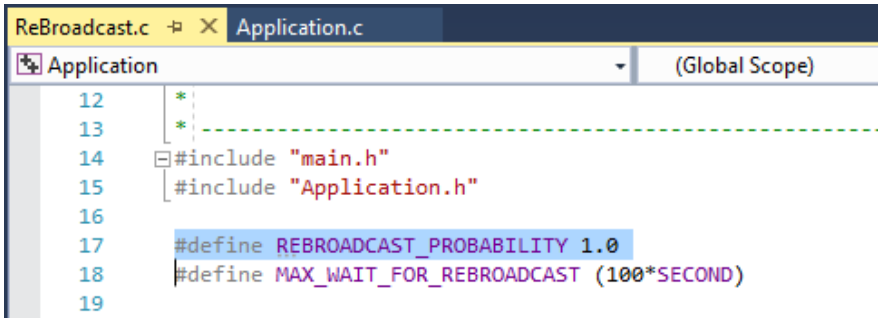
Broadcasting is the process of sending a message from one node to all other nodes in an ad-hoc network. It is a fundamental operation for communication in ad-hoc networks as it allows for the update of network information and route discovery at every node.

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:



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();

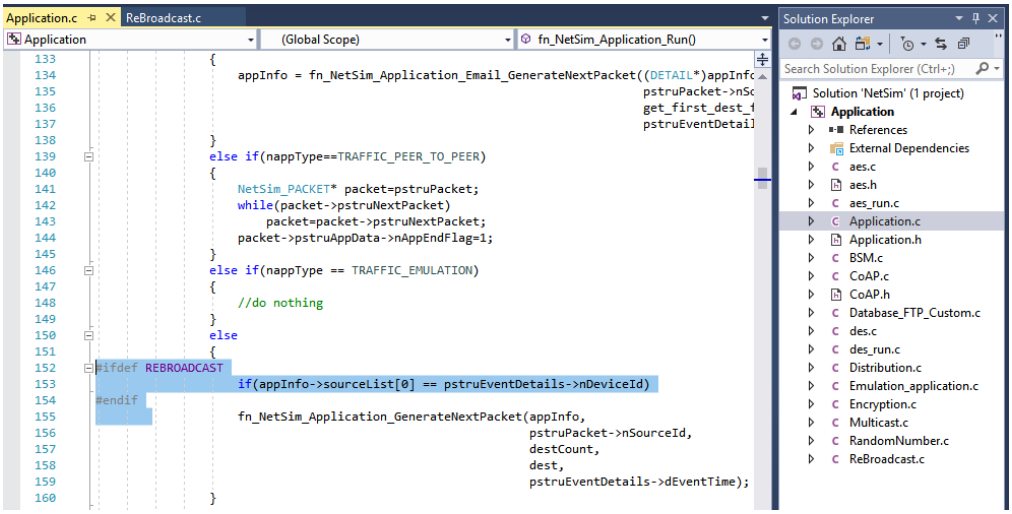
  This function is used to add the packet to rebroadcast list.

- static void cleanup_broadcast_info();

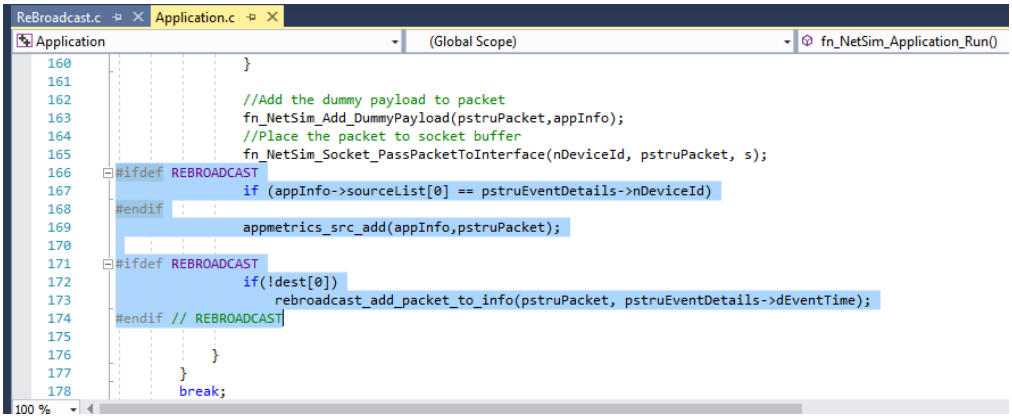
  This function is used to clean the broadcast information.

Code modifications done in NetSim:

1. We have added the following lines of code in fn_NetSim_Application_Run() function in the APPLICATION_OUT_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.



2. The following lines of code are added in the same fn_NetSim_Application_Run() function in the APPLICATION_OUT_EVENT present in Application.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.



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.

```
ReBroadcast.c   Application.h   Application.c
Application                                  (Global Scope)
200                 {
201                     process_saej2735_packet(pstruPacket);
202                 }
203
204 #ifdef REBROADCAST
205                 UINT destCount;
206                 NETSIM_ID* dest = get_dest_from_packet(pstruPacket, &destCount);
207                 if (!dest[0])
208                 {
209                     rebroadcast_packet(pstruPacket,
210                                        pstruEventDetails->nDeviceId,
211                                        pstruEventDetails->dEventTime);
212                 }
213                 else
214                 {
215 #elif
216                 //Delete the packet
217                 fn_NetSim_Packet_FreePacket(pstruPacket);
218  #endif // REBROADCAST
219 #ifdef REBROADCAST
220                 }
221  #endif
222
223
```

```
ReBroadcast.c   Application.c
Application                                  (Global Scope)                    fn_NetSim_Application_Run()
186                 fnValidatePacket(pstruPacket);
187                 pstruappinfo=appInfo[pstruPacket->pstruAppData->nApplicationId-1];
188                 pstruPacket->pstruAppData->dEndTime = pstruEventDetails->dEventTime;
189                 fn_NetSim_Application_Plot(pstruPacket);
190 #ifdef REBROADCAST
191                 if (pstruappinfo->sourceList[0] == pstruPacket->nSourceId)
192  #endif
193                 appmetrics_dest_add(pstruappinfo, pstruPacket, pstruEventDetails->nDeviceId);
194                 if(pstruappinfo->nAppType==TRAFFIC_PEER_TO_PEER && pstruPacket->pstruAppData->nAppEndFlag==1)
195                 {
```

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

```
ReBroadcast.c   Application.h   Application.c
Application                                  (Global Scope)
434 void fnCreatePort(APP_INFO* info);
435 int fnCreateSocketBuffer(APP_INFO* appInfo);
436 _declspec(dllexport) unsigned int fnGetSocketId(NETSIM_ID nAppId,
437                                                 NETSIM_ID nSourceId,
438                                                 UINT destCount,
439                                                 NETSIM_ID* nDestinationId,
440                                                 NETSIM_ID nSourcePort,
441                                                 NETSIM_ID nDestPort); //Function present in NetworkStack.dll.
442
443 int fn_NetSim_Add_DummyPayload(NetSim_PACKET* packet, APP_INFO*);
444
445 //Encryption
446 char xor_encrypt(char ch, long key);
447
448
449
450 /*****REBROADCAST *****/
451 #define REBROADCAST
452 void rebroadcast_add_packet_to_info(NetSim_PACKET* packet,
453                                     double time);
454 void rebroadcast_packet(NetSim_PACKET* packet,
455                         NETSIM_ID devId,
456                         double time);
457 #endif
458
```

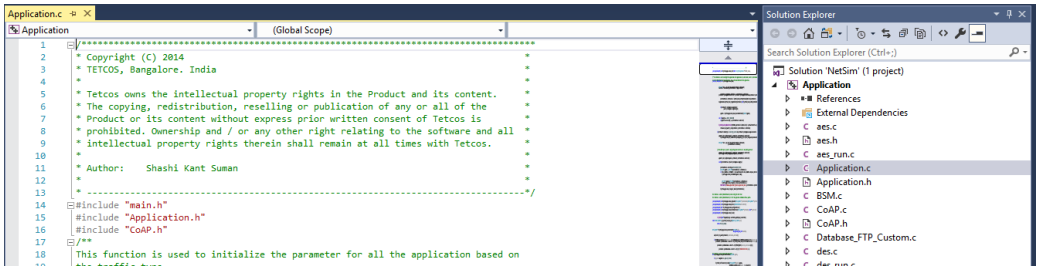
Steps:

- After you unzip the file, the folder would look like:

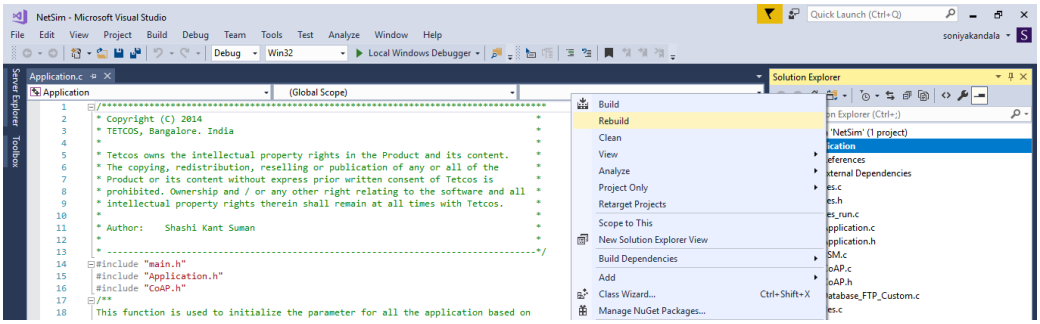
rebroadcasting_in_v10.2

Name	Date modified	Type	Size
Code	15-09-2018 18:04	File folder	
Configuration file	15-09-2018 18:05	File folder	
Documentation	15-09-2018 18:06	File folder	

- Open Code folder and double click on the NetSim.sln file present to open the project in Visual Studio 2017.



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



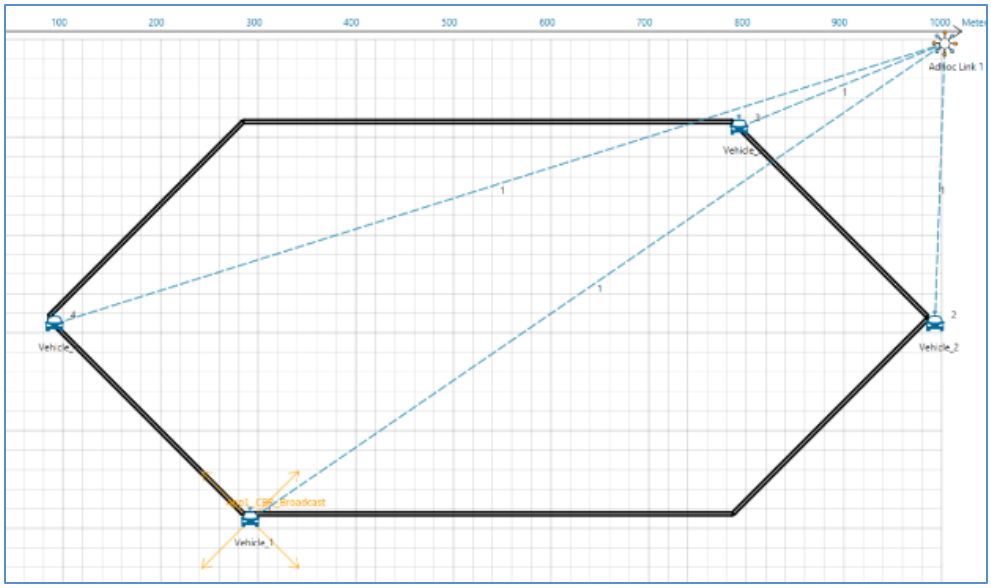
- Upon rebuilding, **libApplication.dll** will get created in the DLL folder.

- Now copy the **libApplication.dll** from the DLL folder and paste it in NetSim bin folder present in the NetSim installation directory. The NetSim install directory would look something like < C:\Program Files\NetSim Standard\bin>.

- Note that there already exists **libApplication.dll** in this bin folder. This is the default file being shipped with NetSim. The user has to replace this file with his newly built file.

- Therefore, take care to rename the original **libApplication.dll** file before replacing it, so that it is backed up. For example, you may rename it as **libApplication_default.dll**.

- Run NetSim and open **Configuration.netsim** file present inside the Configuration file/**VANET or MANET** folder and run the simulation for 100 seconds.
  VANET SCENARIO:



- In the above scenario, Vehice-1 is broadcasting the packet and it is received by the Vehicles 2, 3 and 4. Then Vehices 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.