**Rebroadcasting packet in NetSim MANET\VANETs**

**Software:** NetSim Standard v13.1 (64bit), Microsoft Visual Studio2019

**Project Download Link:**

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

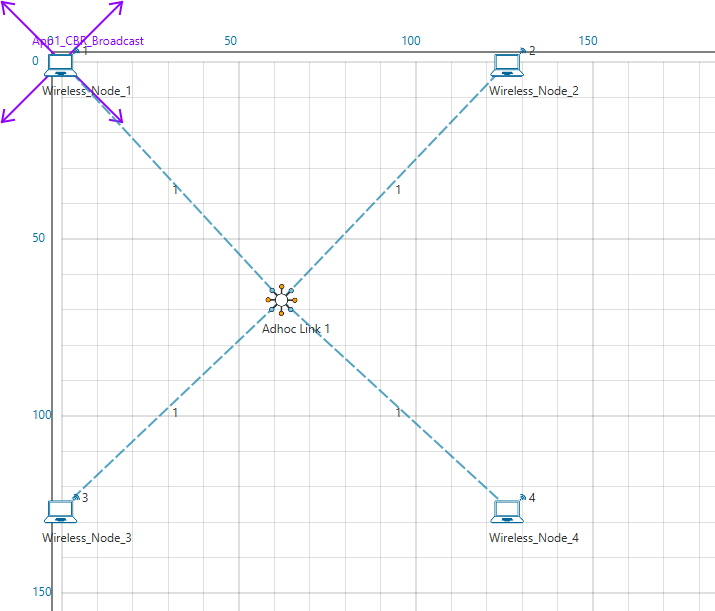
Follow the instructions specified in the following link to download and setup the Project in NetSim:

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**



**Figure 1:** Network Scenario created in MANET

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:

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 2:** Rebroadcast Probability

**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.

**Steps to simulate**

* Open the Source codes in Visual Studio by going to Your work-> Source Code and Clicking on Open code button in NetSim Home Screen window.
* Right click on Solution in Solution Explorer and select ‘Rebuild solution.

A screenshot of a computer

Description automatically generated with medium confidence

**Figure 3**: Screen shot of NetSim project source code in Visual Studio

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

**Example**

1. The Worksapce\_MANET\_VANET\_Rebroadcast comes with a sample network configuration that are already saved. To open this example, go to Your work in the Home screen of NetSim and click on the **Rebroadcast\_VANET\_Example/Rebroadcast\_MANET\_Example** from the list of experiments.
2. Run the simulation for 100 seconds.

**VANET SCENARIO**

**Chart, line chart

Description automatically generated**

**Figure 4:** Network Scenario created in VANET

**Results and discussion**

* In the above scenario, Vehicle-1 is broadcasting the packet and it is received by the Vehicles 2, 3, 4 and 5. Then Vehicles 2, 3, 4 and 5 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.

Table

Description automatically generated

**Figure 5:** NetSim Packet Trace

* 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.

**Appendix: NetSim source code modifications**

**Changes to int fn\_NetSim\_Application\_Run()function in the APPLICATION\_IN\_EVENT, in Application.c file, within Application project**

/\*This is used to generate next broadcast packet if the current device is present in the source list\*/

\_declspec (dllexport) int fn\_NetSim\_Application\_Run()

{

switch(pstruEventDetails->nEventType)

{

case APPLICATION\_OUT\_EVENT:

handle\_app\_out();

break;

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);

}

**Changes to handle\_app\_out() function, in APP\_OUT.c file, within Application project**

/\*The code checks if the destination is ‘0’ i.e., Broadcast packet, then it adds the packet to rebroadcast list\*/

//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

}

**Changes to int fn\_NetSim\_Application\_Run()function in the APPLICATION\_IN\_EVENT, in Application.c file, within 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.\*/

#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);

}

#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

}

**Added the following function declarations in Application.h file, within Application project**

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