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:

ReBroadcast	.c → × Application.c
💁 Applicati	on 🗸 (Global Scope)
12	*
13	*
14	⊟#include "main.h"
15	#include "Application.h"
16	
17	#define REBROADCAST_PROBABILITY 1.0
18	<pre>#define MAX_WAIT_FOR_REBROADCAST (100*SECOND)</pre>
19	

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:

 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.

Application.c	🖻 🗙 ReBroadcast.c		•	Solution Explorer 🔹 म 🗙
Application	•	(Global Scope)	▼ Ø fn_NetSim_Application_Run()	0048-0-58"
Application.c Image: Constraint of the second	x ReBroadcast.c	<pre>(Global Scope) >Info = fn_NetSim_Application_Em f(nappType==TRAFFIC_PEER_TO_PEER tSim_PACKET* packet=pstruPacket; ile(packet->pstruWextPacket) packet=packet->pstruWextPacket cket->pstruAppData->nAppEndFlag= f(nappType == TRAFFIC_EMULATION) do nothing</pre>	 Image: Second state of the second	Solution Explorer ♥ × ● ● ● ● ● × Search Solution Explorer (Ctrl+:) ● ● ● Image: Solution NetSim (1 project) ● ● Image: Solution NetSim (1 project)
152 E 153]≢ifdef REBROADCAST if((appInfo->sourceList[0] == pstru	EventDetails->nDeviceId)	 C Distribution.c C Emulation_application.c
154 155 156 157 158 159 160	#endif fn_	_NetSim_Application_GenerateNext	Packet(appInfo, pstruPacket->nSourceId, destCount, dest, pstruEventDetails->dEventTime);	 C Encryption.c C Multicast.c C RandomNumber.c C ReBroadcast.c

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.

B Application ▼ (Global Scope) ▼ ♥ fn_NetSim_4	Application_Run()
160 }	
161	
162 //Add the dummy payload to packet	
<pre>163 fn_NetSim_Add_DummyPayload(pstruPacket,appInfo);</pre>	
164 //Place the packet to socket buffer	
165 fn_NetSim_Socket_PassPacketToInterface(nDeviceId, pstruPacket, s);	
166 🛱 #ifdef REBROADCAST	
<pre>167 if (appInfo->sourceList[0] == pstruEventDetails->nDeviceId)</pre>	
168 #endif	
169 appmetrics_src_add(appInfo,pstruPacket);	
170	
171 🛱 #ifdef REBROADCAST	
172 if(!dest[0])	
173 rebroadcast_add_packet_to_info(pstruPacket, pstruEventDetails->dEventTime);	
174 #endif // REBROADCAST	
175	
176 }	
177 }	
178 break;	

 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 👳 🗙
Applicati	on	- (Global Scope)
200	Ti i i	{
201		process saei2735 packet(pstruPacket);
202		}
203		-
204	=#ifdef REBROADC	ST
205		UINT destCount;
0 206		<pre>NETSIM_ID* dest = get_dest_from_packet(pstruPacket, &destCount);</pre>
207		if (!dest[0])
208		(
209		rebroadcast_packet(pstruPacket,
210		pstruEventDetails->nDeviceId,
211		<pre>pstruEventDetails->dEventTime);</pre>
212		}
213		else
214		
215	-#elit	
216		//Delete the packet
217	Heredie // DEDDO	fn_wetSim_Packet_FreePacket(pstruPacket);
210	#endin // REDRO	
219	E#ITGET KEDKOADC	
220	#endif	1
222	THE REAL PROPERTY OF THE REAL	
223		
225		
ReBroadcast.c	Application.c 👳	×
Application	n	 Global Scope)
186	fn	alidatePacket(pstruPacket);
187	ps	<pre>ruappinfo=appInfo[pstruPacket->pstruAppData->nApplicationId-1];</pre>
188	ps 	ruPacKet->pstruAppData->dEndlime = pstruEventDetails->dEventTime;
190	Th H#ifdef REBROADCAST	Netsim_Application_fiot(pstrumatket);
191	if	<pre>(pstruappinfo->sourceList[0] == pstruPacket->nSourceId)</pre>
192	#endif	
193	ар	<pre>metrics_dest_add(pstruappinfo, pstruPacket, pstruEventDetails->nDeviceId);</pre>
194	Ė if	pstruappinfo->nAppType==TRAFFIC_PEER_TO_PEER && pstruPacket->pstruAppData->nAppEndFlag==1)
195	10 0 1 (

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

ReBroadcast.c	Application.h 🔹 🗙 Application.c
Application	 (Global Scope)
434	<pre>void fnCreatePort(APP INFO* info);</pre>
435	<pre>int fnCreateSocketBuffer(APP_INFO* appInfo);</pre>
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	<pre>int fn_NetSim_Add_DummyPayload(NetSim_PACKET* packet, APP_INFO*);</pre>
444	
445	//Encryption
446	<pre>char xor_encrypt(char ch, long key);</pre>
447	
448	
449	
450	/****REBROADCAST *****/
451	#define REBROADCAST
452	<pre>void rebroadcast_add_packet_to_info(NetSim_PACKET* packet,</pre>
453	double time);
454	<pre>void rebroadcast_packet(NetSim_PACKET* packet,</pre>
455	NETSIM_ID devId,
456	double time);
457	#endif
458	

Steps:

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

reb	rebroadcasting_in_v10.2								
^	Name	Date modified	Туре	Size					
	Code	15-09-2018 18:04	File folder						
	Configuration file	15-09-2018 18:05	File folder						
		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.

Application.c	₩ <mark>×</mark>		-	Solution Explorer	ųΧ
S Application	 (Global Scope) 			0001 H - 10-5 7 10 0 / -	
1		•••	÷		
2	* Copyright (C) 2014	*		Search Solution Explorer (Ctrl+;)	ρ-
3	* TETCOS, Bangalore, India	*	internation in the	Solution 'NetSim' (1 project)	
4	*	•	-	4 to Application	
5	* Tetcos owns the intellectual property rights in the Product and its content.	+ · · · · · · · · · · · · · · · · · · ·		h an Deferment	
6	* The copying, redistribution, reselling or publication of any or all of the	*	7.95	V M References	
7	* Product or its content without express prior written consent of Tetcos is	•	March 19	P Sternal Dependencies	
8	* prohibited. Ownership and / or any other right relating to the software and all	+ · · · · · · · · · · · · · · · · · · ·	No. of Concession, Name	C aes.c	
9	* intellectual property rights therein shall remain at all times with Tetcos.	*	SARANA SARAN	A aes.h	
10	*	•		C aes_run.c	
11	* Author: Shashi Kant Suman	+ · · · · · · · · · · · · · · · · · · ·	Conception of the local division of the loca	C Application.c	
12	*	*	-	Application.h	
13	*	-*/	Charles Agreen	b C BSM c	
14				h C Calle	
15	#include "Application.h"		CONTRACTOR OF THE OWNER.	P C COAPIC	
16	#include "CoAP.h"			P 🖻 CoAP.h	
17	/**		TATAL LIBERTING	C Database_FTP_Custom.c	
18	This function is used to initialize the parameter for all the application based on			▷ C des.c	
10	and another and the second s		VALUE OF A	h C dar nin c	

• Right click on Solution in Solution Explorer and select 'Rebuild solution'.

ج File	NetSim - Micro Edit View	soch Visual Studio Project Build Debug Team Tools Test Analyze Window Help	3 2	 회 위 위 및 	T & Q	uick Launch (Ctrl+Q)	ndala - S
Server E	pplication.c + Application	• (Global Scope) •			Solution Ex	plorer ───────────────────────────────	- ₽ ×
xplorer Toolbox	1 E 2 3 4 5 6 7 8 9 10 11 12 13 14 E 15 16 17 E	/*************************************		Build Rebuild Clean View Analyze Project Only Retarget Projects Scope to This New Solution Explorer View Build Dependencies Add Class Wizard.) Ctrl+Shift+X	n Esplorer (Ctrl) NetSim (T project) tation dirences ternsl Dependencies esc. ts.h t.Junc. opplication.h SMAC oAP.A oAP.A tabase_FTP_Custom.c	ρ.

- 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.

	А	В	С	D	E	F	G	Н
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.