Early Detection and Prevention of DDos Attack in VANET

Manish Naik

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela-769008, Odhisa

Early Detection and Prevention of DDos Attack in VANET

Dissertation submitted in

May 2014

to the department of

Computer Science and Engineering

of

National Institute of Technology

for the degree of

Masters of technology (Dual Degree)

(Specialization in Information Security)

By

Manish Naik

(Roll 710cs2037)

under the esteemed guidance of

Prof. Ashok Kumar Turuk

National Institute of Technology

Rourkela-769008, Odisha

Dedicated to my Parents

Certificate

This is to certify that the thesis entitled Early Detection and Prevention of DDos Attack in VANET submitted by Manish Naik bearing roll number 710cs2037 in partial fulfilment of the requirement for the award of Masters of Technology (Dual Degree) in Computer Science and Engineering with the specialization in Information Security to National Institute of Technology, Rourkela is a record of own work carried out by him under the guidance of my supervision. The content embodied in this thesis is original and has not been submitted for the award of any other degree.

Ashok Kumar Turuk
Computer Science and Engineering, NIT Rourkela

Acknowledgment

This project work would not have completed without the support of many individuals. I would like to express my warm gratitude to them who has helped me on many aspects from giving simple hints, providing moral support to motivating for embarking on the path of security of networks.

Most importantly I am indebted to Prof. Ashok Kumar Turuk who has guided me through this complex journey of network security. His knowledge and excellence in the field and never ending passion to constantly share ideas among his students to help them shape the future of security in networks has been the prime motivating factor for carrying out my work. I shall remain deeply grateful to him.

I would like to thank my friends for keeping the interest alive in me through regular discussions. They have helped me on every setback by letting me know that time doesn’t wait for winners and losers don’t chose to walk. I am very thankful to my close friend Govind who has provided me with tidy solutions to every intermediate problem, be it installing the GNU Linux System or self configuring a Linux Kernel. I wish to have him in every time of my life.

Above all, the love and affection of my family has helped in ups and downs of my life. I hold their teaching and morals values to be the foremost.

Declaration

I hereby declare that all the work presented in this thesis work is my own research and effort unless sources are otherwise acknowledged. I acknowledge that core part of my work has not been previously submitted for award of any academic degree. References are given to cite difference sources.

Manish Naik

710cs2037

Department of Computer Science and Engineering

National Institute of Technology

Rourkela-769008

CONTENTS

Early Detection and Prevention of DDos Attack in VANET ii
Certificate iv
Acknowledgment v
Declaration vi
Abstract 1
List of Acronyms 2
List of Figures 3
Chapter 1 4
Introduction 4
1.1 Introduction to VANET 4
1.2 Problem Statement and Motivation 4
1.3 Thesis Objective 5
1.4 Thesis Organization 6
Chapter 2 7
Basic concepts of VANET 7
2.1 Architecture of VANET 7
2.1.1 On board unit: 8
2.1.2 Application unit: 9
2.1.3 Road side unit: 10
2.2 Characteristics and Challenges of VANET 10
2.2.1 Different Channel 10
2.2.2 Vehicular Devices 11
2.2.3 Mobility in VANET 11
2.2.4 Network Constraints 11
2.2.5 Applications 12
2.2.6 Security 12
2.3 WAVE Protocol Stack 13
Chapter 3 14
Distributed Denial of Service 14
Chapter 4 16
Literature Review 16
4.1 Channel Switching 16

4.2 Technology Switching 16
4.3 Frequency Hopping Spread Spectrum 16
Chapter 5 18
Proposed Solution and Simulation Work 18
5.1 Introducing Honeypots 18
5.2Implementation and Results 21
The Receiver :: 22
The Sender :: 23
The SYN MESSAGE IMPLEMENTATION:: 24
RESULTS:: 25
5.3 Setting the simulation environment 26
Chapter 6 30
Future Work and Conclusion 30
Bibliography 31

Abstract

Growing number of vehicles in use has ushered in the service to provide human and resource safety. The present trend calls for the application of technology to automate safety measures in road traffic and since has been known as Intelligent Transport System (ITS). Vehicular Ad hoc Network is like a fork to Mobile Ad hoc Network, where the nodes are mobile vehicles moving in constrained road topology. VANET networks are envisioned to be used in practical ITS systems around the world. A network standard has been developed as Wireless Access In Vehicular Environment (IEEE 802.11p) to be used in VANET which is an amendment to IEEE 802.11 standard. With every new technological applications especially computers and network applications, come new security challenges. Every network in modern day is susceptible to security attacks and VANET is no exception. The most infamous of those attacks is the Distributed Denial of Service Attack which is unavoidable because unlike other security attacks the data packets used in it are legitimate packets. In this thesis work previous solutions are reviewed and a new offensive measure for detection, mitigation and prevention has been proposed.

Keywords: VANET, vehicular communication, network, security, DDos

List of Acronyms

VANET	Vehicular Ad Hoc Network
MANET	Mobile Ad Hoc Network
Dos	Denial of Service
DDos	Distributed Denial of Service
WAVE	Wireless Access in Vehicular Environment
DSRC	Dedicated Short Range Communication
TCP	Transmission Control Protocol
VEINS	Vehicles in Network Simulation
SUMO	Simulation in Urban Mobility
OBU	On Board Unit
RSU	Road Side Unit
AU	Application Unit

List of Figures

Figure 1 VANET used in ITS 8
Figure 2 MIT developed test OBUs 9
Figure 3 DSRC channels 11
Figure 4 VANET applications 12
Figure 5 WAVE protocol stack 13
Figure 6 A TCP handshake 15
Figure 7 OBU making decision to mitigate DDos 17
Figure 9 Normal nodes 21
Figure 8 Normal nodes 21
Figure 10 Honeypot 21
Figure 11 Attacker 21
Figure 12Receiver 22
Figure 13Sender 23
Figure 14SYN implementation 24
Figure 16 Running SUMO 26
Figure 17 The corresponding network topology running in Omnetpp 27
Figure 18 Timestamped coupling of two simulators 27
Figure 19 Example network 29

Chapter 1

Introduction

This chapter covers the basics of thesis work with introduction to VANET and Distributed Denial of Service. Further, the problem statement and motivation for project work has been defined.

1.1 Introduction to VANET

Vehicular ad hoc network is a communication network for vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) communications developed mainly for establishing an Intelligent Transport System (ITS) in road traffic for the purpose of prevention of accident, post-accident investigation, mitigation of traffic jams and other non-safety applications. There are 3 basic components of VANET communication i.e. On board unit (OBU), Road side unit (RSU) and Application unit (AU). For communication among vehicles and in between RSU and vehicles a communication channel namely Dedicated Short Range Communication (DSRC) is reserved by different governing authorities around the world. DSRC is an umbrella term for communication channel meant for automotive use. The Federal Communication Commission of United States allocated 75 MHz of spectrum line in the 5.9 GHz band for ITS. Similarly, the European Telecommunication Standards Institute allocated 30MHz of spectrum in 5.9 GHz band. The different DSRC systems of world differ thus rendering them incompatible with one another. India is yet to define such communication channel for ITS. As for development and standardization of VANET IEEE community named Technical Subcommittee on Vehicular Networks and Telematics Application (VNTA) has been established.

1.2 Problem Statement and Motivation

Like any other network, vehicular network is also prone to security attacks. The three main security goals are confidentiality, integrity and availability. The focus of this thesis work is availability. Availability is most important security requirement because as the name suggests any service must be available for use whenever requested by the users. Denial of Service attacks is the most ubiquitous, easy to implement and unavoidable for most of the time. The attack gets worse when a distributed denial of service attack is implemented where there is more than one perpetrator are executing the attack. Ddos attack can occur in any layer of network communication model and accordingly the methods of attacks are distinguished. In our present case of VANET which uses wave protocol stack, the obvious layer susceptible to DDos attack is transport layer. Transport layer is responsible for end to end connection, flow control and error recovery. This layer responds to service request from application layer which is running for software application and provides a transparent data transfer between two nodes. Traditional wired network like Ethernet, WLAN uses Transmission Control Protocol and User Datagram Protocol in transport layer. The fact is Mobile Ad hoc Network (MANET) and more especially VANET has variable network topology which creates problem for TCP to work efficiently. To be more specific VANET is prone to constant packet loss because vehicles are constantly moving and as such TCP misinterpret this as network congestion. TCP will then execute congestion control algorithms to mitigate the congestion but will fail since congestion was not the cause for packet loss. Nevertheless TCP is presently used in VANET as most of the application layer protocols like FTP, HTTP and TELNET which constitute framework the applications in application layer are build on top of TCP. Research and development process for new transport layer schemes are undergoing but TCP is the current standard for transport layer in VANET. As such I have considered this for my thesis work.

1.3 Thesis Objective

Given the nature of VANET, denial of service attacks and distributed denial of service attacks are unavoidable. The main objective of my thesis work is to provide a clever approach to not only detect the attack but also to prevent it in future time. Though the phrase “future time” is overkill as perpetrators would again find a way to execute their dirty business but at least for the time whence network can be made available. I have proposed a method to zone out the attacks so that legitimate users can carry on using the network and attackers are deceived into thinking that they are being successful in denying users of network services. Furthermore, present challenges and a possible solution to identify the adversary are given.

1.4 Thesis Organization

The contents of rest of the thesis are summarized below:

Chapter 2 gives overview architecture, components, characteristics and applications of VANET followed by brief summary of WAVE protocol stack.

Chapter 3 covers the denial of Service and distributed denial of service attacks and their relevance in VANET network.

Chapter 4 covers the literature review of various sources and previous works.

Chapter 5 covers the proposed solution to attack and setup of simulating environment for VANET.

Chapter 6 is the conclusion of the thesis work.

Chapter 2

Basic concepts of VANET

Following technological advancement and fulfilling of every human need with its application, safety in transportation is the next step. Road accidents are more frequent in the present time than before because of high traffic. The reduced cost of owning a vehicle especially cars in urban areas has further worsen the traffic and road accidents. Government and transportation industry has called for a system to maintain this chaotic scenario using growing technologies. The need for Intelligent Transport Systems has ushered in the development of vehicular communication system. This has led to the notion of Vehicular Network and the community word given to this system is known as Vehicular Ad Hoc Network (VANET) . The following section of this chapter gives description of VANET components, architecture, characteristics and application.

2.1 Architecture of VANET

The three main components of VANET are on board unit (OBU), application unit (AU) and road side unit (RSU). Analogous to computer networks each vehicle is represented as a network node and OBU and AU sit on the node. The term "ad hoc" implies that these kinds of networks are self organizing and providing extemporaneous services. Although MANETs, the parent networks, are supposed to be non-infrastructure self establishing communication network a road side unit is needed in VANET to facilitate internet connectivity and information collections for maintenance of network. Thus we can say VANET are hybrid networks whose design has its root in MANET but with few modifications to meet the requirement of feasible vehicular environment. There are three main communication domains, one in vehicle to vehicle communication (V2V) , second is vehicle to infrastructure communication (V2I) and third one of is infrastructural domain through which RSU can provide cellular network radio services.

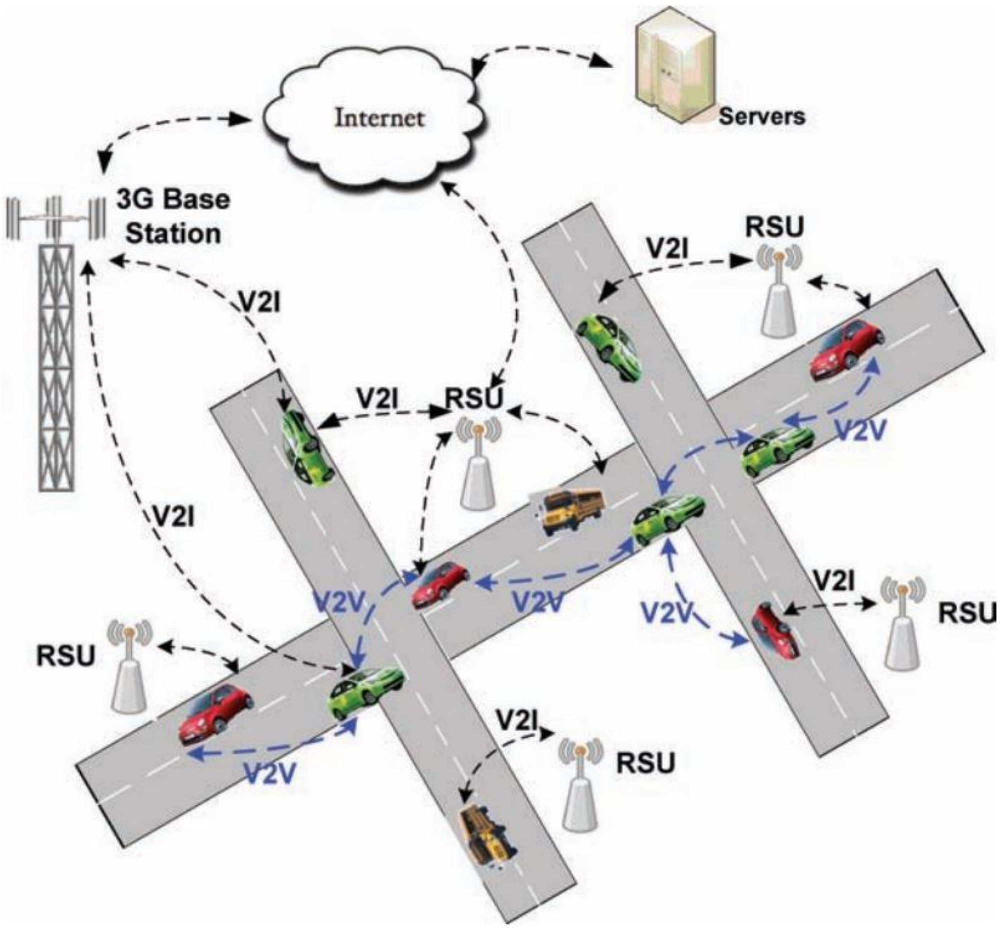


Figure 1 VANET used in ITS

Following is the details of components of VANET:

2.1.1 On board unit:

This is the central processing power house of vehicular node installed in vehicle. This unit can contain a variety of devices that are necessary for communication and information processing. A list is given below:

I. A processor that is needed to process application and communication protocols.
II. A wireless transceiver whose job is to transmit and receive data among itself and other vehicles and road side unit.
III. A GPS receiver for positioning system.

IV. A set of sensors to measure various parameters which can then be processed into information and distributed in network. Special sensors to measure driver physical and mental state can also be employed.

V. Network interfaces for VANET like IEEE 802.11p card and other networks like Bluetooth and infrared communication.

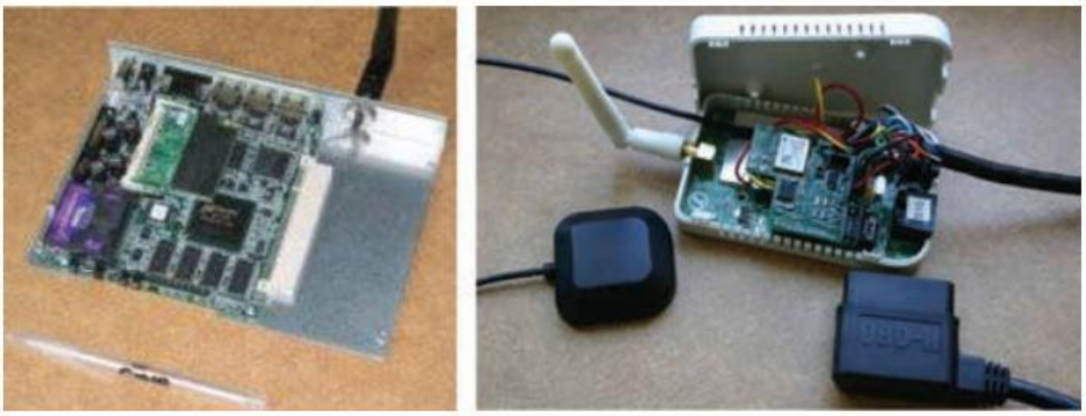


Figure 2 MIT developed test OBUs

Thus OBUs are like computing devices equipped on vehicles to make then work as a network node. Their main objectives are information processing, network access, message transfer and positioning. There are two possibilities of commercialization of OBUs. Either car companies themselves provide their proprietary OBUs fulfilling an established standard or they are commercialized as third party. The VANET resolution is carried out by most Vehicular companies thus there are more odds of former possibility.

2.1.2 Application unit:

The application layer of the network is intended to host a variety of safety and non-safety applications. Application unit is the unit meant to be operated by drivers. AU can be a device with input output interfaces like monitor, keypad, headphone jack, USB port etc. The AU is connected is either wired connected or wirelessly connected to OBU which provide the backbone of network communication. The AU interacts with OBU for accessing the network and internet. Its main function is to host user level safety and non-safety applications.

2.1.3 Road side unit:

It is fixed device located on road side that helps in maintaining the network. It is also equipped with network interfaces compatible with DSRC and IEEE 802.11p. The RSU also facilitates the routing mechanism. The RSU receives and process area oriented information for warnings and advertisement. For example messages informing about an accident in one area get transmitted to other vehicles by hoping through RSU. It can run its own safety and non-safety applications. It also provides internet connectivity to vehicles' OBUs.

The cellular network companies can provides cellular gateway to the VANET whereby they publicize commercial advertisement. In this mode the VANET is said to be infrastructure based network. On the other hand hen vehicles communicate among themselves establishing an ad hoc network, the network is said to be pure ad hoc network. These two models go hand in hand in VANET.

2.2 Characteristics and Challenges of VANET

Unlike computer network VANET has some differentiating characteristics that bring challenges to the direct deployment of existing network standards and hence creates urgency for modification of existing standards or development of standalone standards for VANET ([5]Shuai Chen, 2013). Some of the main characteristics of VANET are described below:

2.2.1 Different Channel

For obvious reasons wired channels cannot be used in VANET but utilizing existing wireless channels is on speculation. As such various organizations around world has been registering dedicated channel for VANET. This channel is collectively known as dedicated short range communication. United States and European Union has defined their own channel specifications. The U.S version is a 75MHz communication line in the 5.9GHz band. A diagram showing different channels of DRSC is shown below:

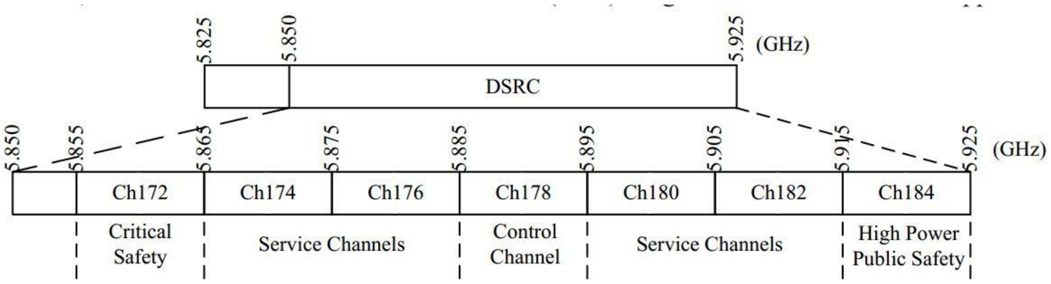


Figure 3 DSRC channels

A global specification of VANET needed to be established to reduce ambiguity.

2.2.2 Vehicular Devices

On board unit and application unit constitute the devices that are set up inside vehicles. On top of that other devices might be needed to facilitate desired objectives. For example just like aeroplanes have black boxes that records crucial events and occurrences inside it, vehicles can also have such event recording devices. In VANET context it is termed as Event Data Recorder. Another device is Global Positioning System device which can be integrated with OBU for secure directions. An electronic plate can be installed that can act as identifying parameter in the network. Thus there is challenge for the development and commercialization of these devices.

2.2.3 Mobility in VANET

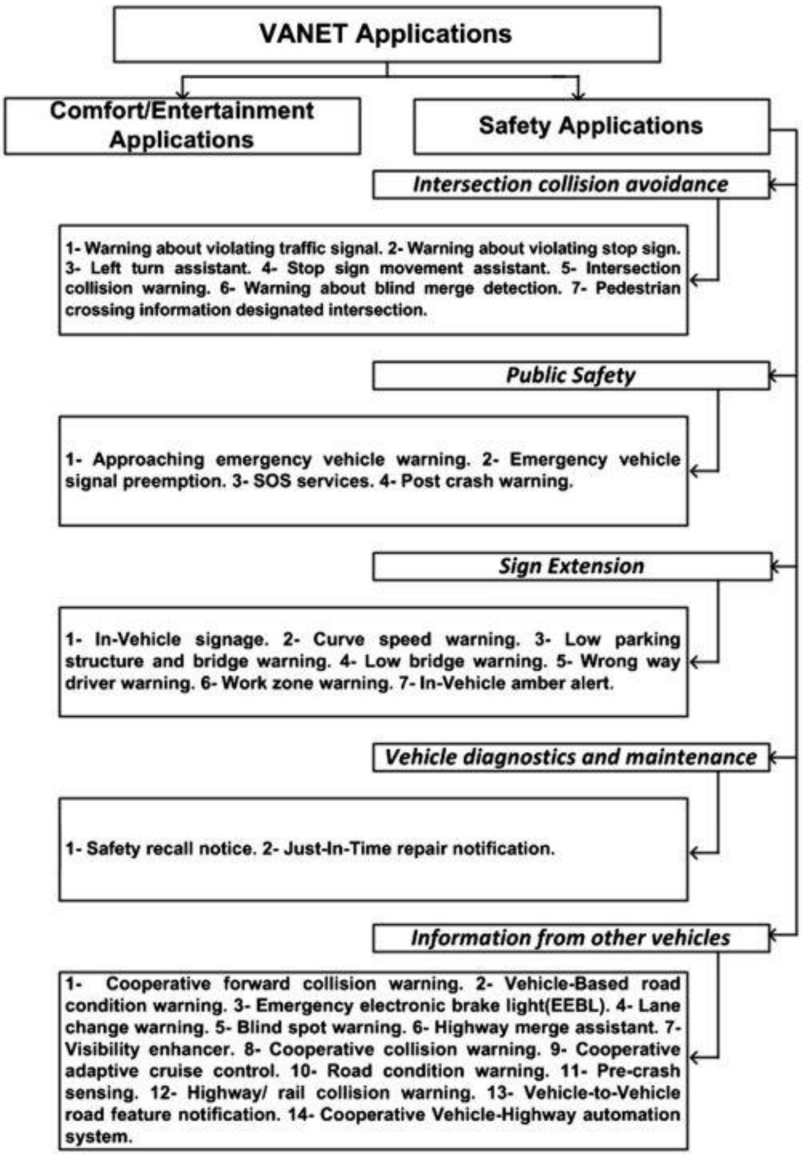
VANET is different from wired network like Ethernet because nodes are not fixed in this case. Unlike MANET where the nodes have to freedom to move randomly nodes in VANET are constrained to road topology and rules and regulation of traffic. The speed and position of every car is variable and as such pose a challenge for multi-hop networking.

2.2.4 Network Constraints

There are various network constraints that make it difficult for existing standards to be used in VANET and thus call for development of new standards. The presences of obstacle in traffic environment can lead to signal fading. This can impact proper delivery of crucial information in time. The DSRC has very short range of bandwidth (10-20Mhz) which can lead to frequent congestions.

2.2.5 Applications

VANET applications can be broadly divided into safety and non safety application ([1] Saif Al-Sultan, 2014). A diagram detailing different applications is shown below:


VANET Applications
Comfort/Entertainment Applications
Safety Applications
Intersection collision avoidance
Public Safety
Sign Extension
Vehicle diagnostics and maintenance
Information from other vehicles

1- Warning about violating traffic signal. 2- Warning about violating stop sign. 3- Left turn assistant. 4- Stop sign movement assistant. 5- Intersection collision warning. 6- Warning about blind merge detection. 7- Pedestrian crossing information designated intersection.

1- Approaching emergency vehicle warning. 2- Emergency vehicle signal preemption. 3- SOS services. 4- Post crash warning.

1- In-Vehicle signage. 2- Curve speed warning. 3- Low parking structure and bridge warning. 4- Low bridge warning. 5- Wrong way driver warning. 6- Work zone warning. 7- In-Vehicle amber alert.

1- Safety recall notice. 2- Just-In-Time repair notification.

1- Cooperative forward collision warning. 2- Vehicle-Based road condition warning. 3- Emergency electronic brake light(EEBL). 4- Lane change warning. 5- Blind spot warning. 6- Highway merge assistant. 7- Visibility enhancer. 8- Cooperative collision warning. 9- Cooperative adaptive cruise control. 10- Road condition warning. 11- Pre-crash sensing. 12- Highway/ rail collision warning. 13- Vehicle-to-Vehicle road feature notification. 14- Cooperative Vehicle-Highway automation system.

Figure 4 VANET applications

2.2.6 Security

Just like any other network secursity threats loom the VANET environments. The main topic of discussion is security of availability which is described in later chapters.

2.3 WAVE Protocol Stack

IEEE 802.11p along with P1609 family of protocols constitutes the WAVE standard. Protocols for different layers are being either ported from previous networks or developed separately. The current protocol stacks is shown in the diagram below.

HTTP	**1609.1 –Application layer**
IEEE 1609.2(Security) , IEEE 1609.3 (WSMP)	TCP/UDP –Transport layer IPV6 Network layer
IEEE 802.2	LLC sub layer
IEEE 802.11 p and 1609.4	MAC sub layer
IEEE 802.11p a	Physical layer

Figure 5 WAVE protocol stack

Chapter 3

Distributed Denial of Service

There are many types of attacks on VANET. Some of them are briefly given below:

1. Sybil Attack: ([2]Halabi Hasbullah, 2010)The objective of the attacker is to scam other nodes into thinking they received some legitimate messages and they should act accordingly. One possible scenario is when a driver wants to clear a traffic it can launch attacking by sending multiple messages to other nodes each with a fabricated source that accident has occurred in the road ahead. Victim nodes can withhold themselves from taking that road path while the attacker node can drive in cleared road without any hassle.

2. Node Impersonation: A vehicle node can send a modified message of a victim node claiming to be real originator. The message can be bogus or which can cause harm to victim node. This kind of attack can be solved by including unique identification number to nodes. Inclusion of ID can lead to another type of attack where victims are exposed of their identity where they wanted so have some privacy.

3. Sending false information: This kind of attack is very common as attacker would want to disrupt the proper functioning of traffic by sending false information and bringing chaos on road.

4. Distributed denial of service is the most infamous type of denial of service attack. In computer networks the attacker spoof network IP addresses and execute attack on a victim computer denying it resources and accessibility of network. The DDos can be classified as according to the layer it is attacking. In VANET context the layer I took for probing is transport layer and assumed existing TCP is used in it. In general DDos attack is executed by sending redundant messages over time making the victim node unable to respond to other legitimate messages and thus suspending it to either provide service or receive service. The TCP DDos attack is named as SYN flooding attack. Following is the description of SYN flooding attack.

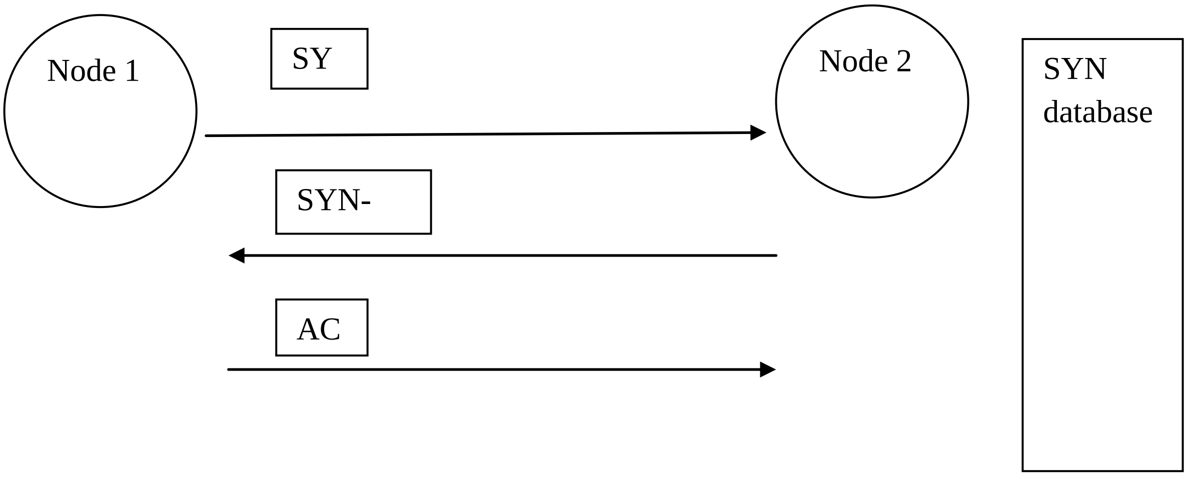


Figure 6 A TCP handshake

For two nodes to communicate a TCP connection should be established. A TCP handshaking establishes this connection. It consists of 3 steps. In first step the packets which are numbered (SYN for synchronization packets) for flow control mechanism is send to the receiver. In the second step the receiver send back SYN-ACK (acknowledge packet). The connection is established when sender send ACK packet and receiver receives it. When only SYN command is sent from one node to another the receiver save the SYN message in a data structure. At this point the connection is said to be half open. Many nodes can send multiple SYN messages to a node. In DDos attack the attacker along with compromised nodes (zombie nodes) send multiple SYN messages to victim. The victim node can be devoid of legitimate requests by other nodes when the data structure to hold SYN messages is filled with redundant SYN messages.

Chapter 4

Literature Review

There are existing solutions to mitigate the denial of service attack [2]. Those solutions are based switching channels or technology whenever one channel or technology is found to infected with Dos attack. A brief description of the method used is given below:

4.1 Channel Switching

DSRC model provides a number of channels for communication. There are 7 channels each of 10MHz bandwidth having upper cap on data transfer rate of 27Mbps. For communication it's not necessary for all the channels to be active. So any denial of service attack on a node can be mitigated by switching to other channel.

4.2 Technology Switching

Many communication technologies work with VANET as there are three main communication domains namely V2V, V2I and Vehicle to infrastructures. This method of solutions suggests the switching of technology whenever a attack is detected.

4.3 Frequency Hopping Spread Spectrum

This method increases the bandwidth of signals by adding keys so that packets are sent over a set of different frequency range. Whenever an attack is launched, the network communication hops into different frequency channel.

All the above solution is carried out by the OBU. OBU is programmed to detect denial of service attack. What this means is a load on vehicle resources which is not unfavourable unless this hinders with the default working of OBU.

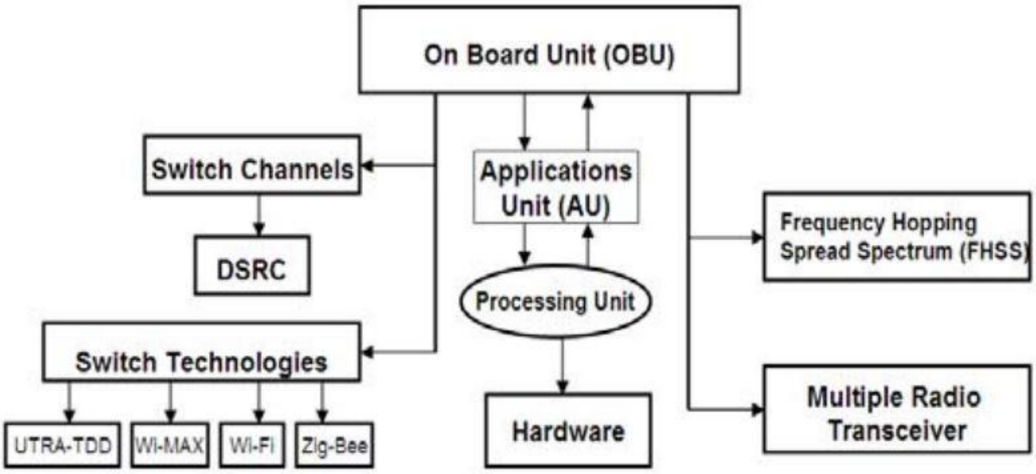


Figure 7 OBU making decision to mitigate DDos

Chapter 5

Proposed Solution and Simulation Work

Defence mechanisms against DDOs attacks can be classified into prevention, detection, mitigation and response. Previous works showed how DDos can be detected and mitigated by switching channels and technologies. This is a passive method of dealing with DDos where the nodes are attacked no matter what. In my proposal I have combined clever method of using SYN cookies and Intrusion detection system to not only detect before time but also to prevent any further attack.

Traditional detection schemes rely on signature based detection algorithms that is it run on the RSU or the OBU of some authorities like Police cars. The routing protocol generates a range of metadata that is then fed back to IDS which then detect the attack. But DDos attacks usually cannot be detected until any node is suspended of service because most of the packets used to carry out attack are legitimate. So it is given that a node will be attack now or in future.

5.1 Introducing Honeypots

My proposal is the deployment of nodes that has same computing resources as the normal node in the network whose sole job is to pose as victim. In other word this kind of node i.e. honeypot node is a closely monitored computing resource that we want to be probed, attacked or even compromised. The honeypot can be physical in which a real vehicle can carry this undercover zombie node or it can be virtual node where a vehicle (preferably Police cars) runs a parallel node along with its own operation. The production value of honeypot is zero and hence any attempt to contact will be suspicious by definition. Once a honeypot is deployed the next steps are detection and prevention.

The detection scheme is based on clever approach by D.J Bernstein who proposed this scheme for Linux kernel patch number 2.0.29. Basically in our SYN flooding example our victim node was suspended of services by the attacker when the data structure to hold SYN messages are filled. The solution is to send back a particular node a SYN cookie containing all the SYN only this node has sent to the receiver and delete the SYN messages from receiver end. The receiver end will only respond to the sender whenever the sender sends ACK along with SYN cookie. In the mean time space was preserved in the SYN data structure of receiver end. Thus a particular node having unique address can only flood SYN messages up to limit. My proposal is to give the honey node a small memory data structure that will trigger alert whenever redundant SYN messages are found filling the data structure. The detection algorithm can be summarized below.

Algorithm Detection

Input: SYN messages

Output: Alert on datastructure full

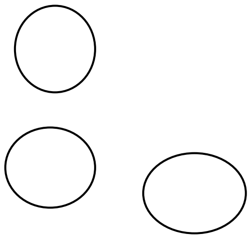
```
1   if(synid==SYN.ID&&i=!MEM.size)
2        cookie=merge(SYN)
3       send(cookie,ACK)
4       sum+=1 //global variable
5       if(ack()==ACK&&receive()!=cookie)
6       do nothing
7       elseif(ack()==ACK&&receive()==cookie)
8       sum-=1 //global variable
9       if(sum==MEM.size)
10          alert("DDOS")
12          break
13  else
14  Detection(SYN)
15  end
```

The algorithm works as follows. The detection program takes SYN, which is a data structure holding SYN messages of a particular node identified by synid, MEM which is the allocated memory of SYN database, synid and ACK data structure for a particular SYN. Our objective is to check whether the database to hold is filled or not. When a SYN message arrive line 2 check if it belongs to the same node or not and whether the

MEM data base is already filled. If SYN message belongs to same node as received previously then a cookie object is created by merging all SYN messages of the same node and is send back along with ACK. The sender has two options to send back. One is to send the cookie plus ACK or just send ACK. Sending of just ACK has no effect as it will not be acknowledged and the connection will be disconnected. If ACK with cookie is received then sum variable is decremented thus maintaining the MEM memory. Whenever sum equals to MEM then the database to hold SYN messages is filled an attack has been detected. The else condition passes control to recursive Detection program with different node id as determined by synid.

Up until now our honey has been compromised and detection has been done. The second part is to prevent further Dos attack. Just like any adversary attack the weakest link the DDos will be executed on similar node having security flaws. The security flaw mentioned here is the small memory of data structure to hold SYN messages. The DDos attack creates a network of zombie nodes by first compromising the nodes and installing a program in them which also makes them execute attack on other nodes. Two things can be said now. One is DDos is spread attack and hence up to a certain cap zombie nodes will be created. Second DDos can be facilitating itself i.e. in our cases a honey will facilitate further attack. But our intention is to prevent the attack we can deceive the attacker into thinking that many nodes are compromised and DDos attack is successful. In reality before DDos has spread to normal nodes we have contained the attack among a set of self deployed honeypot nodes. These set of nodes are collectively named as honey net. So in my proposal DDos is known by the detection scheme defined above and it is contained in such a manner that attacker is deceived into thinking DDos is ongoing and spreading in the network.

Figure 8 Normal nodes



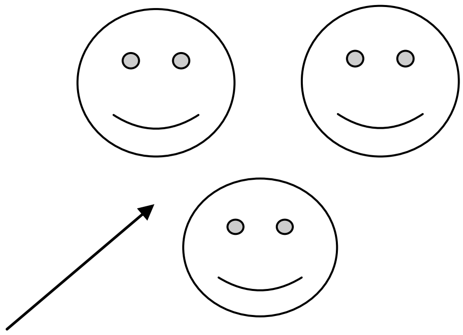


Figure 10 Honeypot

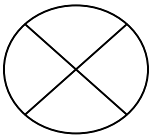


Figure 11 Attacker

As previously stated honeypots can be virtual or physical. In computer networks a physical computer node can pose as honeypot or it can run a virtual operating system connected to internet which will eventually act as honeypot. Another thing to be notified is there are high level honeypots and low level honeypots. The former is a full functioning network node whereas the later is a network node that simply implements the communication protocol stack. In VANET a physical node can be a police car or any law enforcer. The virtual node is what we should consider. In order to simulate the behaviour of virtual node in the simulation environment a node can be created on the simulating software.

5.2Implementation and Results

The code snippets and results are shown in this section:

The Receiver ::

```
#include<cstdio>

#include<cstdlib>

#include<iostream>

#include<cstring>

#include<unistd.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<fcntl.h>

#include "Syn.h"

#include<vector>

using namespace std;

void Cal(vector<Syn> &ob,int &sum){

    cout<<"\nTotal number of successful connections made:"<<sum<<"\n";

    for(int i=0;i<ob.size();i++){

        cout<<"NODE with SYN
id\t"<<ob[i].getid()<<"\tmade\t"<<ob[i].getcounter()<<" legitimate conncetions";

        cout<<"\nAnd made "<<ob[i].getho()<<" Half Open "<<"Connection\n\n";

    }

}

int Chk(vector<Syn> &ob,string s){

    for(int i=0;i<ob.size();i++){

        if(ob[i].getid()==s){

            cout<<ob[i].getid()<<"\n";

            return i;

        }

    }

    return 999;
```

Figure 12Receiver

The Sender ::

```
#include<cstdio>

#include<cstdlib>

#include<iostream>

#include<cstring>

#include<unistd.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<fcntl.h>

using namespace std;

int main(){

        int l;

        char buf[256];

        char buff[256];

        cout<<"Sender NODE is up...";

        int fd=open("./myfifo",O_WRONLY);

        int df=open("./mfifo",O_RDONLY);

        if(fd==-1){

                cerr<<"Error opening fifo:";

                return EXIT_FAILURE;

        }

        while(1&&strcmp(buf,"done")!=0){

        cout<<"\nEnter:";

        cin>>buf;

        write(fd,buf,sizeof(buf));

        read(df,buff,sizeof(buff));

        cout<<"Acknowledge "<<buff<<"?1)YES \n2)NO\n";

        cin>>buf;

        if(strcmp(buf,"YES")==0){

                write(fd,buf,sizeof(buf));
```

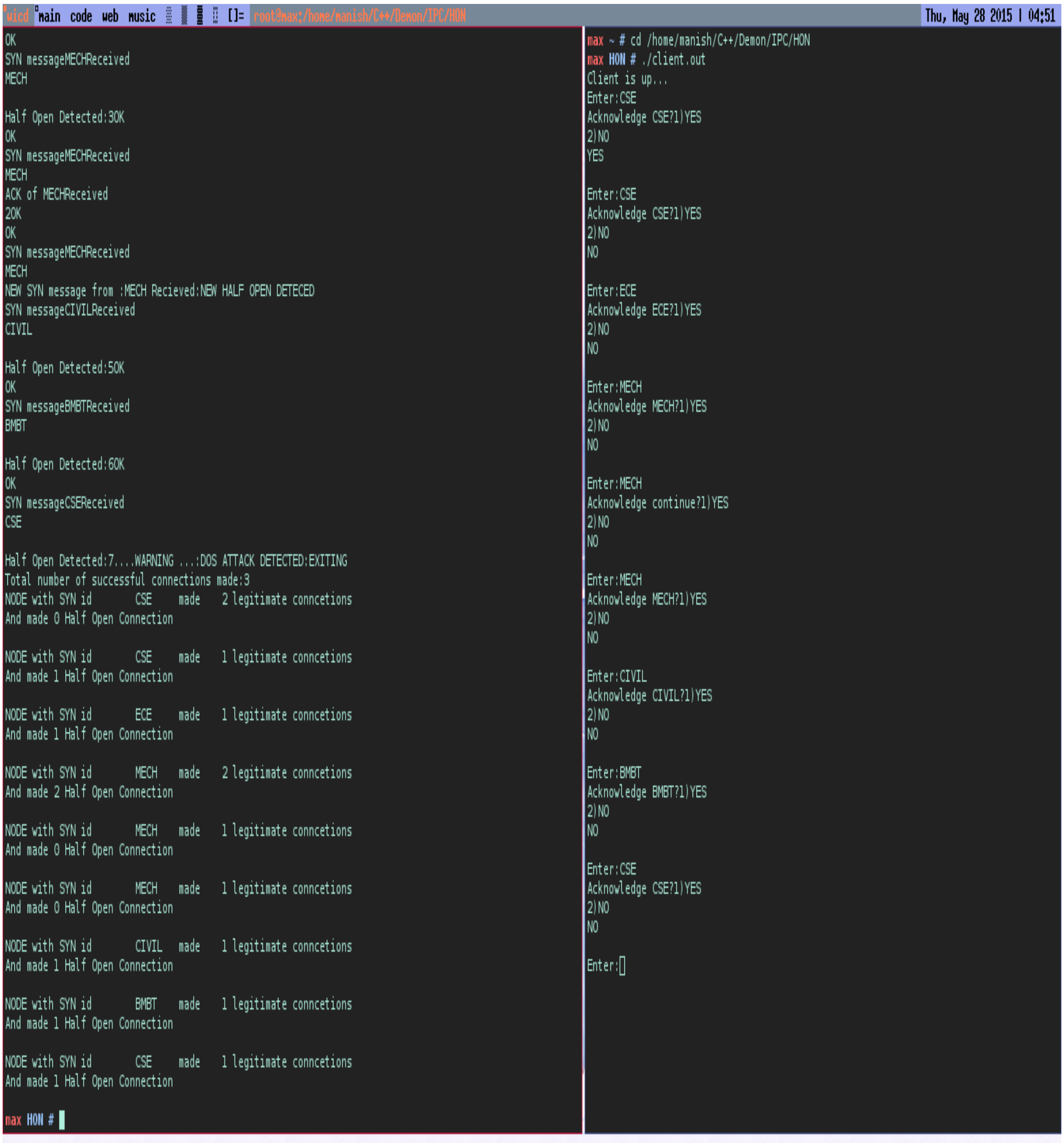
Figure 13Sender

The SYN MESSAGE IMPLEMENTATION::

```
#include "Syn.h"

Syn::Syn(){

        counter = 0;

        id="NONE";

        flag = false;

        ho=0;

}

void Syn::incho(){

        ho+=1;

}

void Syn::decho(){

        ho-=1;

}

Syn::Syn(string name="NONE"):id(name),flag(false){

}

int Syn::getcounter(){

        return counter;

}

void Syn::setcounter(int x){

        counter = x;

}

void Syn::increment(){

        counter+=1;

}

void Syn::decrement(){

        counter-=1;

}
```

Figure 14SYN implementation

RESULTS::



```
OK
SYN messageMECHReceived
MECH

Half Open Detected:3OK
OK
SYN messageMECHReceived
MECH
ACK of MECHReceived
2OK
OK
SYN messageMECHReceived
MECH
NEW SYN message from :MECH Recieved:NEW HALF OPEN DETECED
SYN messageCIVILReceived
CIVIL

Half Open Detected:5OK
OK
SYN messageBMBTReceived
BMBT

Half Open Detected:6OK
OK
SYN messageCSEReceived
CSE

Half Open Detected:7....WARNING ...:DOS ATTACK DETECTED:EXITING
Total number of successful connections made:3
NODE with SYN id        CSE     made    2 legitimate conncetions
And made 0 Half Open Connection

NODE with SYN id        CSE     made    1 legitimate conncetions
And made 1 Half Open Connection

NODE with SYN id        ECE     made    1 legitimate conncetions
And made 1 Half Open Connection

NODE with SYN id        MECH    made    2 legitimate conncetions
And made 2 Half Open Connection

NODE with SYN id        MECH    made    1 legitimate conncetions
And made 0 Half Open Connection

NODE with SYN id        MECH    made    1 legitimate conncetions
And made 0 Half Open Connection

NODE with SYN id        CIVIL   made    1 legitimate conncetions
And made 1 Half Open Connection

NODE with SYN id        BMBT    made    1 legitimate conncetions
And made 1 Half Open Connection

NODE with SYN id        CSE     made    1 legitimate conncetions
And made 1 Half Open Connection

max HON #
```

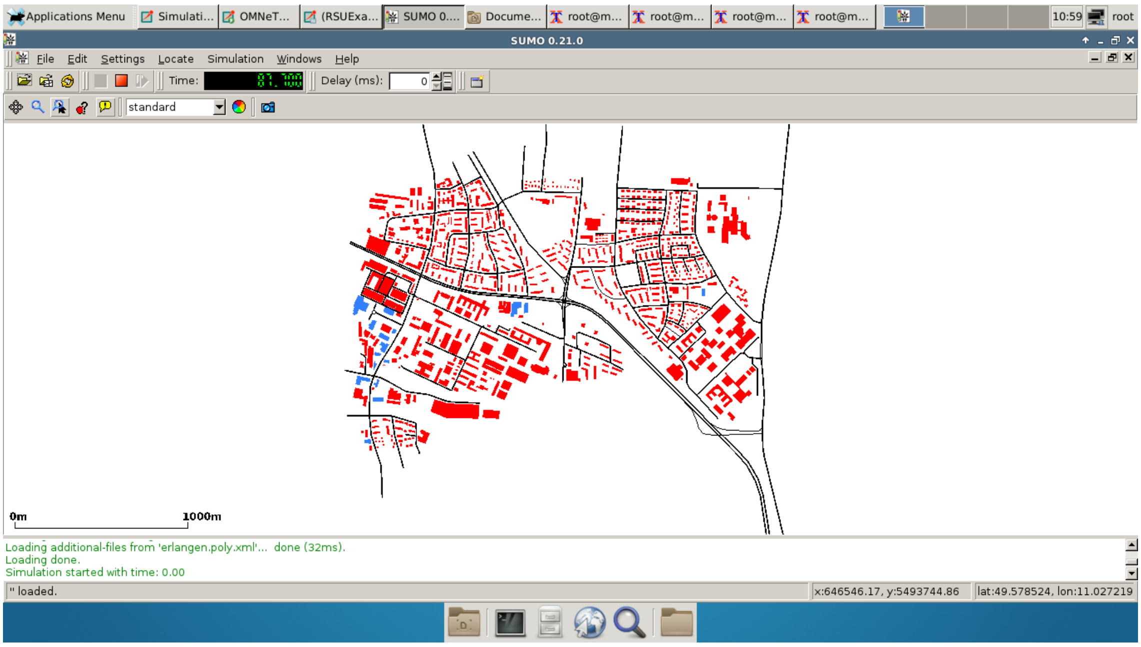
```
max ~ # cd /home/manish/C++/Demon/IPC/HON
max HON # ./client.out
Client is up...
Enter:CSE
Acknowledge CSE?1)YES
2)NO
YES

Enter:CSE
Acknowledge CSE?1)YES
2)NO
NO

Enter:ECE
Acknowledge ECE?1)YES
2)NO
NO

Enter:MECH
Acknowledge MECH?1)YES
2)NO
NO

Enter:MECH
Acknowledge continue?1)YES
2)NO
NO

Enter:MECH
Acknowledge MECH?1)YES
2)NO
NO

Enter:CIVIL
Acknowledge CIVIL?1)YES
2)NO
NO

Enter:BMBT
Acknowledge BMBT?1)YES
2)NO
NO

Enter:CSE
Acknowledge CSE?1)YES
2)NO
NO

Enter:[]
```

As shown in the output above 7 half open connections were detected.

5.3 Setting the simulation environment

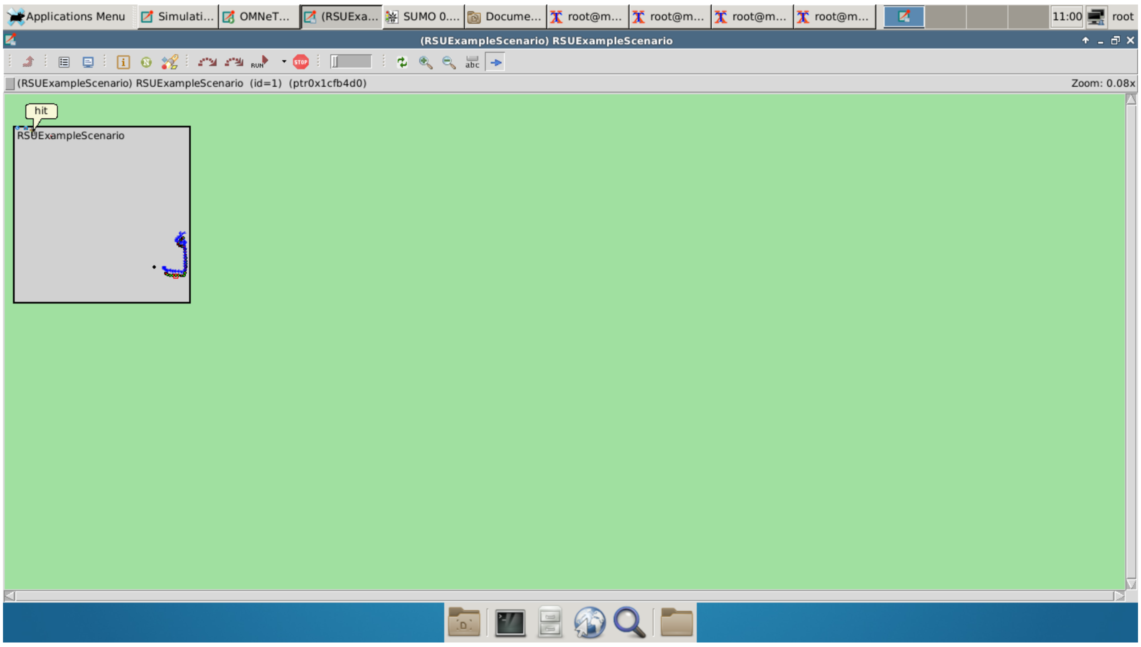
The simulation environment is setup using two different simulators. One simulator is used for simulating the vehicular movement and other simulator is set up to run the VANET network. SUMO is the simulator for vehicles and Omnetpp is the simulator for VANET network. Both simulator are coupled together to work in coordination. A TCP connection is used to connect two simulating environment. Omnetpp is a discrete event simulator meaning the state of system changes in discrete time. Thus at regular interval of time that is at regular timestamp both simulator share simulating parameters which is then used to simulate objects in other simulator. Both of these simulators are installed in a Linux system. Various packages to support a GUI interface are also installed.

Various networking frameworks can be created in omnetpp to model real networks. VEINS is the modelling framework to simulate VANET. The whole tcp session and its operations are written into VEINS framework. A demo to simulate the vehicle and network simulation in coordination is carried out.



```
Loading additional-files from 'erlangen.poly.xml'...  done (32ms).
Loading done.
Simulation started with time: 0.00
" loaded.
x:646546.17, y:5493744.86
lat:49.578524, lon:11.027219
```

Figure 15 Running SUMO

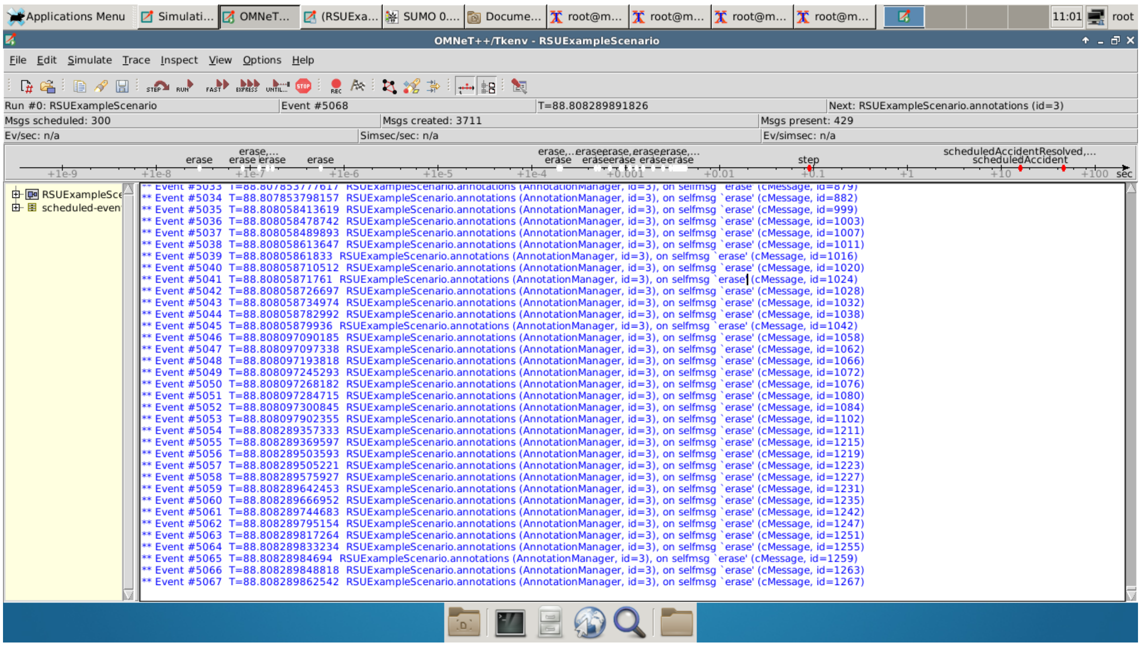


(RSUExampleScenario) RSUExampleScenario

(RSUExampleScenario) RSUExampleScenario (id=1) (ptr0x1cfb4d0)

Zoom: 0.08x

Figure 16 The corresponding network topology running in Omnetpp



OMNeT++/Tkenv - RSUExampleScenario

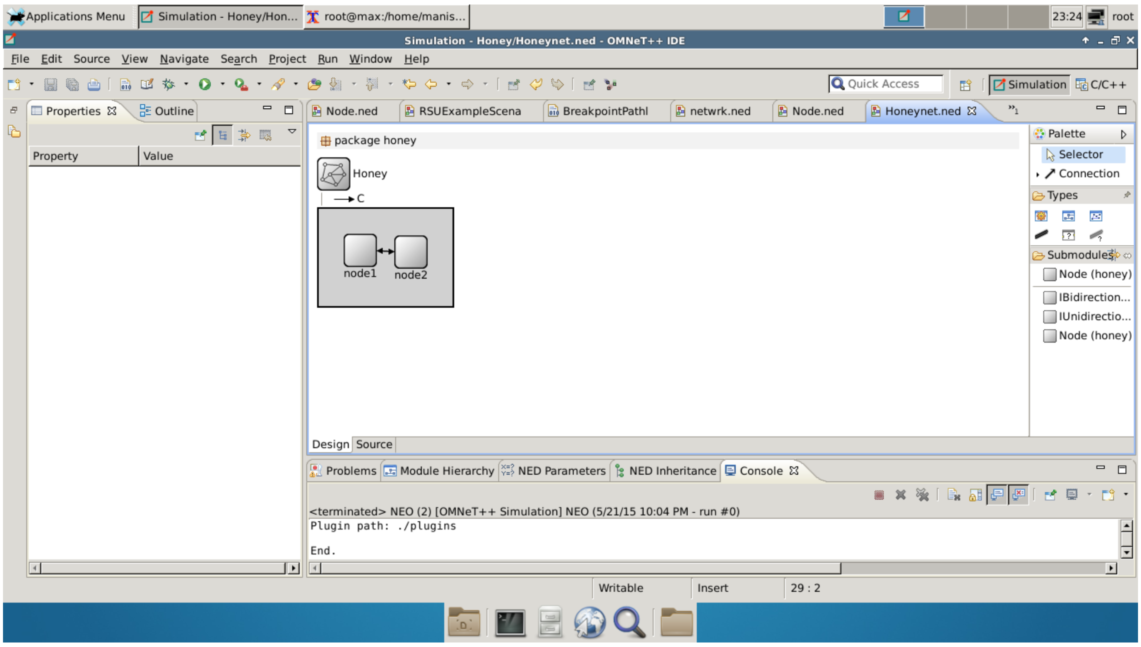
```
Run #0: RSUExampleScenario          Event #5068          T=88.808289891826          Next: RSUExampleScenario.annotations (id=3)
Msgs scheduled: 300                 Msgs created: 3711                              Msgs present: 429
Ev/sec: n/a                         Simsec/sec: n/a                                 Ev/simsec: n/a
```

```
** Event #5033  T=88.807853777617  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=879)
** Event #5034  T=88.807853798157  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=882)
** Event #5035  T=88.808058413619  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=999)
** Event #5036  T=88.808058478742  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1003)
** Event #5037  T=88.808058489893  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1007)
** Event #5038  T=88.808058613647  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1011)
** Event #5039  T=88.80805861833  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1016)
** Event #5040  T=88.808058710512  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1020)
** Event #5041  T=88.80805871761  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1024)
** Event #5042  T=88.808058726697  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1028)
** Event #5043  T=88.808058734974  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1032)
** Event #5044  T=88.808058782992  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1038)
** Event #5045  T=88.80805879936  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1042)
** Event #5046  T=88.808097090185  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1058)
** Event #5047  T=88.808097097338  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1062)
** Event #5048  T=88.808097193818  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1066)
** Event #5049  T=88.808097245293  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1072)
** Event #5050  T=88.808097268182  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1076)
** Event #5051  T=88.808097284715  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1080)
** Event #5052  T=88.808097300845  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1084)
** Event #5053  T=88.808097902355  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1102)
** Event #5054  T=88.808289357333  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1211)
** Event #5055  T=88.808289369597  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1215)
** Event #5056  T=88.808289503593  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1219)
** Event #5057  T=88.808289505221  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1223)
** Event #5058  T=88.808289575927  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1227)
** Event #5059  T=88.808289642453  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1231)
** Event #5060  T=88.808289666952  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1235)
** Event #5061  T=88.808289744683  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1242)
** Event #5062  T=88.808289795154  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1247)
** Event #5063  T=88.808289817264  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1251)
** Event #5064  T=88.808289833234  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1255)
** Event #5065  T=88.8082898694  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1259)
** Event #5066  T=88.808289848818  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1263)
** Event #5067  T=88.808289862542  RSUExampleScenario.annotations (AnnotationManager, id=3), on selfmsg `erase' (cMessage, id=1267)
```

Figure 17 Timestamped coupling of two simulators

In Omnetpp a network is called a model which can be created with components like nodes, channels and infrastructure. A model is created with keyword network. The components of network have type simple which stands for “simple module.” Simple modules can be combined to form a compound module or node. Each node communicates to other node by message passing. Each node has “gates” field which describe the input and output port. Messages are passed through a channel. The channel is described in network model. Both the network definition and node definition is saved a ned file. Below is the code for defining the network and nodes.

```
//Honeynet.net
package honey;
network Honey
{
        types :
                channel C extends ned.DatarateChannel {
                        datarate = 27Mbps;
                }
        submodules :
                node1: Node;
                node2: Node;
        connections :
                node1.in <-- C <-- node2.out;
                node1.out --> C --> node2.in;
}

//Node.ned
package honey;
simple Node
{
        parameters :
        int syn;
        int id;
gates:
        input in;
        output out;
}
```



```
<terminated> NEO (2) [OMNeT++ Simulation] NEO (5/21/15 10:04 PM - run #0)
Plugin path: ./plugins

End.
```

Figure 18 Example network

Chapter 6

Future Work and Conclusion

The predicament of developing VANET protocols and standardization of them do not allow proper implementation of any solutions to security threats. For example there is a naming problem in VANET as existing IP protocol suite cannot be applied because of ad hoc nature of it. Flooding is the basic technique presently in use to carry message from one hop to another. In my proposed solution to DDos I have assumed the sender node to have unique id throughout. But unlike computer networks where nodes are identified with their IP addresses which are then address resolved to respective MAC address. The nodes in VANET are dynamic and the connection is mostly extemporaneous in nature. Without a proper unique naming scheme the method described cannot be implemented.

In future VANET will continue to mature and new standards will be introduced for different layers especially transport layer. My work can be extended to devise modified mechanism to detect and isolate the DDos attack. Honeypots can not only be used to tackle DDos but can also be used to defend other security breach like privacy and integrity or to catch a cyber criminal. The abstract analogy is of an undercover agent hired to infiltrate the organized crime. Thus this field has many open calls for protecting the security goals and will continue to stand because criminals are no backward.

Bibliography

[1] Saif Al-Sultan, M. M.-D. (2014). A comprehensive survey on vehicular Ad hoc network. *Journal of Network and Computer Applications* .

[2]Halabi Hasbullah, I. A.-l. (2010). Denial of Service (DOS) and its possible solutions. *World Academy of Science, Vol: 4 2010-05-25.*

[3]Richard Gilles Engoulou, M. B. (2014). *VANET security survey.* Computer Communications, Vol 44 (2014) 1-13.

[4]Qingzi Liu, Q. W. (2013). A hierarchical security architecture of VANET. *University of Armed Police Force,* .

[5]Shuai Chen, W. N. (2013). Key Indices Analysis of IEEE 802.11p Based Vehicle to Infrastructure System in Highway Environment. *13th COTA International Conference of Transportation Professionals (CICTP 2013).* Jiangsu, China: Procedia Social And Behavioral Science .

[6] Lu Chen, H. T. (2013). Analysis of VANET security based on routing protocol information. *Fourth International Conference on Intelligent Control and Information Processing (ICICIP).* Bejing, China.

[7]Linda Shafer, S. N. (2013). *Mobile Ad Hoc Networking.* IEEE Press.

[8]Security, U. H. (2014). *DDos Attack Quick guide National Cybersecurity and.* National Cybersecurity and Communications Integration Center.

[9]HassnaaMoustafa, Y. n. *Vehicular Networks Techniques, Standards and Applications.* CRC Press.