**FHGD: APPLIED FUZZY HEURISTICS FOR GREEDY NODE DETECTION IN VANETs**

**A PROJECT REPORT**

***Submitted by***

|  |  |
| --- | --- |
| **ANUPRIYA . M** | **(Reg. No. 201406004)** |
| **POOJA . P** | **(Reg. No. 201406045)** |
| **DHIVYABARATHI .K** | **(Reg.No.201406251)** |

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

**in**

INFORMATION TECHNOLOGY

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI**

**(An Autonomous Institution affiliated to Anna University Chennnai)**

**April 2018**

**BONAFIDE CERTIFICATE**

Certified that this project report titled “**FHGD:APPLIED FUZZY HEURISTICS FOR GREEDY NODE DETECTION”** is the bonafide work of **M.ANUPRIYA(201406004),P.POOJA(201406045),K.DHIVYABARATHI(201406251)** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**Dr.T.Revathi,M.E.,Ph.D.,** **Mrs.R.Venitta Raj, M.E.,**

**Head of the Department Supervisor**

**Senior Professor, Assistant Professor(Senior Grade),**

**Department of Information Technology, Department of Information Technology,**

**Mepco Schlenk Engineering College, Mepco Schlenk Engineering College,**

**Sivakasi. Sivakasi.**

Submitted for viva-Voce Examination held at **MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI (AUTONOMOUS)** on **……………..............**

**Internal Examiner External Examiner**

**ACKNOWLEDGEMENT**

**TABLE OF CONTENTS**

**ABSTRACT**

The  Vehicular Ad hoc Network(VANETs) has a greater significance in Vehicle to Vehicle Communication without relaying much on the Infrastructure.Though they are used in variety of applications they are vulnerable to attacks. In this proposed approach we have to detect greedy nodes  by incorporating various metrics like Number of times the node makes an Attempt, Connection Duration , Waiting time, Packet Delivery Ratio and Average Delay. These metrics are used by the benign nodes to learn more about malicious node behavior. In order to detect greedy nodes our proposed model has two phases namely Uncertainty  State and Resolution State .The Uncertainty  State is based on linear regression concept and Resolution state is based on Fuzzy Logic Scheme. The effectiveness of our approach is proved by extensive simulations. This proposed work can be applicable for variety of applications as they provide high level of trustworthiness by detecting greedy nodes in the network

**LIST OF ABBREVIATIONS:**

* VANETs : Vehicular Ad hoc Networks
* DOS : Denial of Service
* WLAN : Wireless LAN
* MANETs : Mobile Ad hoc Networks
* GDVAN : Greedy Detection for VANETs
* EDCA : Enhanced Distributed Channel Access
* DCF : Distributed Coordination Function
* CCH : Control Channel
* SCH : Service Channels
* DSRC : Dedicated Short Range Communication
* WAVE : Wireless Access in Vehicular Environments
* CSMA/CA : Carrier Sense Multiple Access with Collision Avoidance
* AC : Access Categories
* BK : Background traffic
* BE : Best Effort traffic
* VI : Video traffic
* VO : Voice traffic
* AIFS  : Arbitration Inter-Frame Space
* CW : Contention window
* FLSAC : Fuzzy Logic based Scheme to Struggle Against Adaptive

Cheaters

* WIBSS : Wave Independent Basic Service Set

**LIST OF SYMBOLS:**

* T : Monitoring period
* N : Total number of vehicles
* TCA : Connections attempts during T.
* TCD : Total connections duration of all vehicles during T.
* P1nca : First Threshold of connection attempts .
* P2nca : Second Threshold of connection attempts.
* P1cd : First Threshold of connection duration.
* P2cd : Second Threshold of connection duration.
* P1wtbc : First Threshold of waiting times
* P2wtbc : Threshold of waiting times between connections
* V1 : Number of connections attempts.
* V2 : Connection duration.
* V3 : Average of waiting times between connections.

**LIST OF FIGURES**

**INTRODUCTION**

* 1. **Aim**

The aim of the project is to detect and identify the greedy nodes in

Vehicular ad-hoc networks(VANETs).

* 1. **Objective**
* Data plane of ad-hoc network suffer from greedy node. To Overcome the security problem the suspicious phase and decision are implemented to detect greedy node
* The availability of greedy node is calculated by using linear regression concept
* The behavior of node is detected by using fuzzy logic.

**1.3 Attacks And Security Measures**

AD-HOC networks are self-organized wireless networks without fixed or centralized infrastructures. Nodes in ad-hoc networks act as both clients and routers. Some applications of ad-hoc networks could include industrial and commercial applications involving cooperative mobile data exchange, such as military and rescue operations. Recently, emerging technologies such as wireless sensor networks(WSNs),wearable computing, pervasive computing, Internet of Things, and intelligent transportation have largely contributed to a further push toward application potentials of ad-hoc networks.ad-hoc networks exhibit the defining characteristics of openness, dynamic topology, and distributed operation. Due to these characteristics, ad-hoc networks have also served as a vehicle for various security attacks, such as Blackhole,JellyFish“**Impact of denial of service attack on ad-hoc network**”[1],denial of service (DoS),collusion, packet dropper,spoofing,eavesdropping,flooding,rushing,and Sybil.

Naturally, certain attacks can be categorized into two types:

Attacks aiming at disrupting normal packet delivery in the data plane and

Attacks preventing network elements from running correct protocol and management policies in the control and management plane.

Compared with attacks in the control and management plane, data-plane attacks seem to be more difficult to detect and resist because they often do not violate protocol rules.Consequently,the anomaly detection counter measures become invalid. The main focus on implementation model is resisting data-plane attacks. There have been substantial literatures on protecting ad-hoc networks. The commonly used identity-based security mechanisms based on cryptology (traditional security mechanisms)are able to authenticate identity of network entities and to assure the confidentiality and integrity of messages “**identity based deniable authentication for adhoc network**”[2] and “**A cryptography based protocol against packet dropping and message tampering attack on mobile ad-hoc network**”[3]. Hence,attacks such as spoofing,eavesdropping, and Sybil can be well resisted.

**LITERATURE SURVEY:**

The literature study is organized into several different sections. These sections explore about the attacks and defenses in the data plane of ad-hoc networks. Fuzzy trust evaluation and credibility development in multi-agent systems.

**2.1 Attacks and Defenses In The data Plane of Networks (2012)**

In [11] Security issues in computer networks have focused on attacks on end-systems and the control plane. An entirely new class of emerging network attacks aims at the data plane of the network. Data plane forwarding in network routers has traditionally been implemented with custom-logic hardware, but recent router designs increasingly use software programmable network processors for packet forwarding. These general purpose processing devices exhibit software vulnerabilities and are susceptible to attacks.

The attack that exploits vulnerability in packet processing software to launch a devastating Denial-of-Service attack from within the network infrastructure. This attack uses only a single attack packet to consume the full link bandwidth of the router’s outgoing link.

The hardware based defenses mechanism that can detect situations where malicious packets try to change the operation of the network processor. A recovery system can restore the network processor to a safe state within six cycles. This high-speed detection and recovery system can ensure that network processors can be protected effectively and efficiently from this new class of attacks.

2.2

**SYSTEM STUDY:**

**3.1 SCOPE:**

The scope of the project is to detect and identify the greedy node in Vechicular ad-hoc networks.

**3.2 Novelty**

* Compared with existing model the implementation model is able to detect the greedy nodes in high mobility, increasing bandwidth.
* Benefiting from the flexibility of fuzzy reasoning in ex-tending and adapting empirical rules, the evaluation result could be more objective.

**3.3 Proposed Algorithm**

**3.4 Procedure**

**3.5 System Requirements**

The Software and hardware requirements of the system are as follows.

**3.5.1 Hardware Requirements**

* Intel Core i3 preprocessor @ 1.90GHZ
* 4 GB RAM
* 400 GB Hard Disk

**3.5.2 Software Requirements**

* Linux 3.0.7
* NS2

**3.6 Technology Used**

**3.6.1 NS2**

NS is an event driven network simulator developed at UC Berkeley that simulates variety of IP networks. It implements network protocol such as TCP and UDP, traffic sources behavior such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as Drop Tail, RED and CBQ routing algorithm such as Dijkstra, and more. NS also implements multicasting and some of the MAC layer protocols for LAN simulations.

The NS project is now the part of the VINT project that develops tools for simulation results display, analysis and converters that convert network topologies generated by well-known generators to NS formats. Currently, NS (Version 2) written in C++ and OTcl (Tcl script language with Object-Oriented extensions developed at MIT) is available.

NS is written not only Otcl but in C++ also. For efficiency reason, NS separates the data path implementation from control path implementation. In order to reduce packet and event processing time (not simulation time), the event scheduler and the basic network component objects in the data path are written and compiled using C++. These compiled object are made available to the Otcl interpreter through an Otcl linkage that creates a matching Otcl object for each of the C++ objects and makes the control function and the configurable variable specified by the C++ object act as member function and member variables of the corresponding Otcl object. In this way, the controls of the C++ objects are given to OTcl. It is also possible to add member functions and variables to a C++ linked to OTcl object.

When a simulation data finished, NS produces one or more text-based output files that contain detailed simulation data,if specified to do so in the input Tcl(or more specifically, OTcl)