**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.

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 FirstThreshold 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.

**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 Reqiurements

* Linux 3.0.7
* NS2

3.6 Techology 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 DropTail, 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 efficency reason, NS seperates 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 variablesof the corresponding Otcl object.