International Journal of Research in Advent Technology (IJRAT) (E-ISSN: 2321-9637)

Special Issue

National Conference "CONVERGENCE 2016", 06th-07th April 2016

# Implementation of Efficient Neighbor Route Discovery Protocol [ENRDP] for Position Verification in MANET

Prachee U. Ketkar
M.E.,Computer Science & Engineering
P.L.I.T.M.S.,Buldhana
Prachee.ketkar23@gmail.com

Dr. Pradip M. Jawandhiya
Principal,
P.L.I.T.M.S.,Buldhana
Principal_plit@rediffmail.com

## ABSTRACT

Energy awareness and protocol management is becoming an important factor in the design of MANET protocols. Because of mobility, it needs the support of scalable routing strategies. These protocols try to consider the path duration in order to respect some QoS constraints and to reduce fake neighbor position for route discovery. In the existing system communication and packet delivery ratio decreases when neighbor discovery fails. The proposed ENRDP protocol selects the stable path to reduce the fake position and communication overhead. It selects the stable path, so neighbor discovery failure decreases and there by increases the packet delivery ratio and reduces energy consumption, End-to-End delay and packet lost. The effectiveness of ENRDP protocol is demonstrated through NS2 stimulation.

**Keywords:** MANET, ENRDP, Neighbor discovery, Secure neighbor Discovery, Neighbor position verification.

## 1.INTRODUCTION

The term MANET (Mobile Ad hoc Network) refers to a multihop packet based wireless network composed of a set of mobile nodes that can communicate and move at the same time, without using any kind of fixed wired infrastructure. MANET is actually self organizing and adaptive networks that can be formed and deformed on-the-fly without the need of any centralized administration. Otherwise, a stand for "Mobile Ad Hoc Network" A MANET is a type of ad hoc network that can change locations and configure itself on the fly. Because MANETS are mobile, they use wireless connections to connect to various networks. This can be a standard Wi-Ficonnection, or another medium, such as a cellular or satellite transmission.

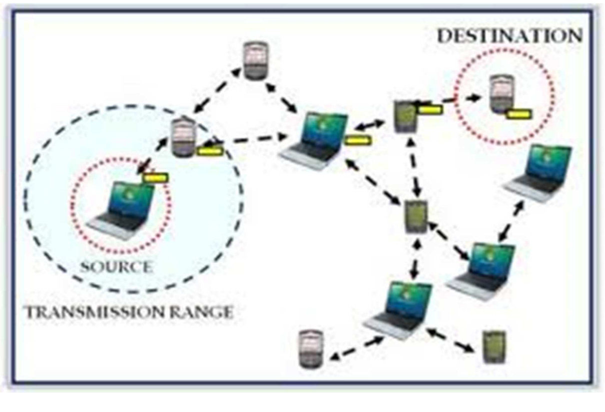


**Fig 1: Structure of MANET**

## 2. EXISTING SYSTEM

The existing approaches require that mobile nodes learn the position of their neighbors. However, such a process can be easily abused or disrupted by adversarial nodes. In absence of a-priori trusted nodes, the discovery and verification of neighbor positions presents challenges that have been scarcely investigated in the literature. Geographic routing in spontaneous networks, data gathering in sensor networks, movement coordination among autonomous robotic nodes, location-specific services for handheld devices, and danger warning or traffic monitoring in vehicular networks are all examples of services that build on the availability of neighbor position information. The correctness of node locations is therefore an all important issue in mobile networks, and it becomes particularly challenging in the presence of adversaries aiming at harming the system

### 2.1Disadvantages of Existing System

- Correctly establish their location in spite of attacks feeding false location information, and
- Verify the positions of their neighbors, so as to detect adversarial nodes announcing false locations.

## 3.PROPOSED SYSTEM

In this paper, we focus on the latter aspect, hereinafter referred to as neighbor position verification (NPV for short). Specifically, we deal with a mobile ad hoc network, where a pervasive infrastructure is not present, and the location data must be obtained through node-to-node communication. Such a scenario is of particular interest since it leaves the door open for adversarial nodes to misuse or disrupt the location-based services. For example, by advertising forged positions, adversaries could bias geographic routing or data gathering processes, attracting network traffic and then eavesdropping or discarding it. Similarly, counterfeit positions could grant adversaries unauthorized access to location-dependent services, let vehicles forfeit road tolls, disrupt vehicular traffic or endanger passengers and drivers.

### 3.1 Advantages of Proposed System

- Our NPV scheme is compatible with state-of the-art security architectures, including the ones that have been proposed for vehicular networks.
- It is lightweight, as it generates low overhead traffic.
- It is robust against independent and colluding adversaries
- It leverages cooperation but allows a node to perform all verification procedures autonomously

## 4. SYSTEM REQUIREMENTS

### 4.1. Hardware Requirements

- Processor -Pentium –III
- Speed - 1.1 Ghz
- RAM - 256 MB(min)
- Hard Disk - 20 GB
- Floppy Drive -1.44 MB
- Key Board - Standard Windows Keyboard
- Mouse -Two or Three Button Mouse
- Monitor - SVGA

### 4.2. Software Requirements:-

- Operating System - LINUX
- Tool - Network Simulator-2
- Front End - O TCL (Object Oriented Tool Command Language

## 5. MODULE DESCRIPTION

- Node configuration setting
- Nodes unique identity
- Message exchange process for route discovery
- Distance computation
- Node position verification
- Node verification process
- Graph examination

### 5.1 Node Configuration Setting

The mobile nodes are designed and configured dynamically, designed to employ across the network, the nodes are set according to the X, Y, Z dimension, which the nodes have the direct transmission range to all other nodes.

### 5.2 Nodes Unique Identity

All the mobile nodes tend to have a unique id for its identification process, since the mobile nodes communicates with other nodes through its own network id. If any mobile node opted out of the network then the particular node should surrender its network id to the head node.

### 5.3 Message Exchange Process for Route Discovery

This module states a 4 step message exchange process i,e POLL, REPLY, REVEAL, REPORT. As soon the protocol executed the, POLL and REPLY messages are firstbroadcasted by Source and its neighbors, respectively. These messages are anonymous and take advantage of the broadcast nature of the wireless medium, allowing nodes to record reciprocal timing information without disclosing their identities.

### 5.4 Distance Computation

In order to compute the distance range, after a POLL and REPLY message a REVEAL message broadcast by the source nodes disclose to S, through secure and authenticated REPORT messages, their identities as well as the anonymous timing information they collected. The source S uses such data to match timings and identities; then, it uses the timings to perform ToF-based ranging and compute distances between all pairs of communicating nodes in its neighborhood.

### 5.5 Node Position Verification

Once Source node has derived such distances, it runs several position verification tests in order to classify each candidateneighbor as either:Verified node, i.e., a node the verifier deems to be at the claimed position or Faulty node, i.e., a node the verifier deems to have announced an incorrect position or Unverifiable node, i.e., a node the verifier cannot prove to be either correct or faulty, due to insufficient information. The position verification are performed by direct symmetric test, cross symmetry test and multilateration test.

## 5.6 Node Verification Process

In this module a proposed work of node verification technique is introduced to detect the adversary nodes in the network. The node verification is done by hash function technique the public key and id of source node generates hash id. In the same way the neighbor nodes generate the hash id, if the source node hash id and neighbor node hash id are same then the nodes are authenticated for data transmission through the minimum distance range discovered path to destination.

## 5.7 Graph Examination

The performance analysis of the existing and proposed work is examined through graphical analysis.

## 6.IMPLEMENTATIONENVIRONMENT

Network simulator 2 is used as the simulation tool in this project. NS was chosen as the simulator partly because of the range of features it provides and partly because it has an open source code that can be modified and extended. There are different versions of NS and the latest version is ns-2.1b9a while ns-2.1b10 is under development.

## 6.1. Network Simulator (NS)

Network simulator (NS) is an object–oriented, discrete event simulator for networking research. NS provides substantial support for simulation of TCP, routing and multicast protocols over wired and wireless networks. The simulator is a result of an ongoing effort of research and developed. Even though there is a considerable confidence in NS, it is not a polished product yet and bugs are being discovered and corrected continuously.NS is written in C++, with an OTcl1 interpreter as a command and configuration interface. The C++ part, which is fast to run but slower to change, is used for detailed protocol implementation. The OTcl part, on the other hand, which runs much slower but can be changed very fastquickly, is used for simulation configuration. One of the advantages of this split-language program approach is that it allows for fast generation of large scenarios. To simply use the simulator, it is sufficient to know

OTcl. On the other hand, one disadvantage is that modifying and extending the simulator requires programming and debugging in both languages.

NS can simulate the following:

- **Topology:** Wired, wireless
- **Sheduling Algorithms:** RED, Drop Tail,
- **Transport Protocols:** TCP, UDP
- **Routing:** Static and dynamic routing
- **Application:** FTP, HTTP, Telnet, Traffic generators

## 6.2Network Components

This section talks about the NS components, mostly compound network components. Figure 1.1 shows a partial OTcl class hierarchy of NS, which will help understanding the basic network components.

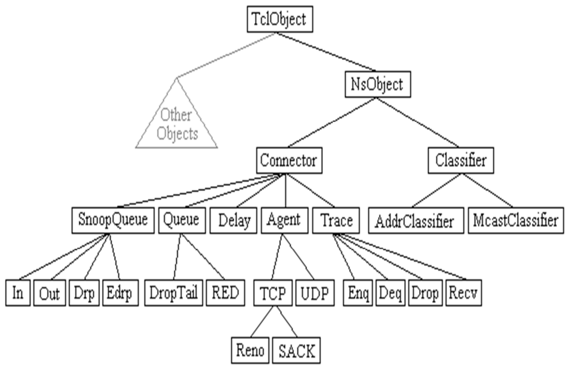


**Fig.2:OTcl Class Hierarchy**

The root of the hierarchy is the TclObject class that is the super class of all OTcl library objects (scheduler, network components, timers and the other objects including NAM related ones). As an ancestor class of TclObject, NsObject class is the super class of all basic network component objects that handle packets, which may compose compound network objects such as nodes and links. The basic network components are further divided into two subclasses, Connector and Classifier, based on the number of the possible output DATA paths. The basic network and

objects that have only one output DATA path are under the Connector class, and switching objects that have possible multiple output DATA paths are under the Classifier class.

## 6.3 Class TCL

The class Tcl encapsulates the actual instance of the OTcl interpreter and provides the methods to access and communicate with that interpreter, code. The class provides methods for the following operations: 1.obtain a reference to the Tel instance

2.invokeOTcl procedures through the interpreter

3.retrieve, or pass back results to the interpreter

4.report error situations and exit in an uniform manner

5.store and lookup "TclObjects"

6.acquire direct access to the interpreter.

**Obtain a Reference to the class Tcl instance**

A single instance of the class is declared in -tclcl/Tcl.cc as a static member variable. The statement required to access this instance is Tel&tel = Tcl::instance();

**Invoking OTcl Procedures**

There are four different methods to invoke an OTcl command through the instance, tcl. They differ essentially in their calling arguments. Each function passes a string to the interpreter that then evaluates the string in a global context. These methods will return to the caller if the interpreter returns TCL_OK. On the other hand, if the interpreter returns TCL_ERROR, the methods will call tkerror{}. The user can overload this procedure to selectively disregard certain types of errors.

1.**Passing Results to/from the Interpreter :**When the interpreter invokes a C++ method, it expects the result back in the private member variable, tcl-> result.

2.**Error Reporting and Exit:** This method provides a uniform way to report errors in the compiled code.

## 7. CONCLUSION

We presented a distributed solution for NPV, which allows any node in a mobile ad hoc network to verify the position of its communication neighbors without relying on a priori trustworthy nodes. Our analysis showed that our protocol is very robust to attacks by independent as well as colluding adversaries, even when they have perfect knowledge of the neighborhood of the verifier. Simulation results confirm that our solution is effective in identifying nodes advertising false positions, while keeping the probability of false positives low. Only an overwhelming presence of colluding adversaries in the neighborhood of the verifier, or the unlikely presence of fully collinear network topologies, can degrade the effectiveness of our NPV. Future work will aim at integrating the NPV protocol in higher layer protocols, as well as at extending it to a proactive paradigm, useful in presence of applications that need each node to constantly verify the position of its neighbors.

## 8.REFERENCES

[1] 1609.2-2006: IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages, IEEE, 2006.

[2] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure Vehicular Communications: Design and Architecture," IEEE Comm. Magazine, vol. 46, no. 11, pp. 100-109, Nov. 2008.

[3] P. Papadimitratos and A. Jovanovic, "GNSS-Based Positioning: Attacks and Countermeasures," Proc. IEEE Military Comm. Conf. (MILCOM), Nov. 2008.

[4] L. Lazos and R. Poovendran, "HiRLoc: High-Resolution Robust Localization for Wireless Sensor Networks," IEEE J. Selected Areas in Comm., vol. 24, no. 2, pp. 233-246, Feb. 2006.

[5] R. Poovendran and L. Lazos, "A Graph Theoretic Framework for Preventing the Wormhole Attack," Wireless Networks, vol. 13, pp. 27-59, 2007.

[6] S. Zhong, M. Jadliwala, S. Upadhyaya, and C. Qiao, "Towards a Theory of Robust Localization against Malicious Beacon Nodes," Proc. IEEE INFOCOM, Apr. 2008.

[7] P. Papadimitratos, M. Poturalski, P. Schaller, P. Lafourcade, D. Basin, S. _Capkun, and J.-P.Hubaux, "Secure Neighborhood Discovery:AFundamental Element for Mobile Ad Hoc Networks," IEEE Comm. Magazine, vol. 46, no. 2, pp. 132-139, Feb. 2008.

[8] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks," Proc. IEEE INFOCOM, Apr. 2003.

[9] J. Eriksson, S. Krishnamurthy, and M. Faloutsos, "TrueLink: A Practical Countermeasure to the Wormhole Attack in Wireless Networks," Proc. IEEE 14th Int'l Conf. Network Protocols (ICNP), Nov. 2006.

[10] R. Maheshwari, J. Gao, and S. Das, "Detecting Wormhole Attacks in Wireless Networks Using Connectivity Information," Proc. IEEE INFOCOM, Apr. 2007.

[11] R. Shokri, M. Poturalski, G. Ravot, P. Papadimitratos, and J.-P.Hubaux, "A Practical Secure Neighbor Verification Protocol for Wireless Sensor Networks," Proc. Second ACM Conf. Wireless Network Security (WiSec), Mar. 2009.

[12] M. Poturalski, P. Papadimitratos, and J.-P.Hubaux, "Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), Mar. 2008.

[13] M. Poturalksi, P. Papadimitratos, and J.-P.Hubaux, "Towards Provable Secure Neighbor Discovery in Wireless Networks," Proc. Workshop Formal Methods in Security Eng., Oct. 2008.

[14] E. Ekici, S. Vural, J. McNair, and D. Al-Abri, "Secure Probabilistic Location Verification in Randomly Deployed Wireless Sensor Networks," Elsevier Ad Hoc Networks, vol. 6, no. 2, pp. 195-209, 2008.

[15] J. Chiang, J. Haas, and Y. Hu, "Secure and Precise Location Verification Using Distance Bounding and Simultaneous Multilateration," Proc. Second ACM Conf. Wireless Network Security (WiSec), Mar. 2009.

[16] S. _Capkun, K. Rasmussen, M. Cagalj, and M. Srivastava, “Secure Location Verification with Hidden and Mobile Base Stations,” IEEE Trans. Mobile Computing, vol. 7, no. 4, pp. 470-483, Apr. 2008.

[17] S. _Capkun and J.-P. Hubaux, “Secure Positioning in Wireless Networks,” IEEE J. Selected Areas in Comm., vol. 24, no. 2, pp. 221- 232, Feb. 2006.

[18] A. Vora and M. Nesterenko, “Secure Location Verification Using Radio Broadcast,” IEEE Trans. Dependable and Secure Computing, vol. 3, no. 4, pp. 377-385, Oct.-Dec. 2006.

[19] J. Hwang, T. He, and Y. Kim, “Detecting Phantom Nodes in Wireless Sensor Networks,” Proc. IEEE INFOCOM, May 2007.

[20] T. Leinmu¨ ller, C. Maiho¨ fer, E. Schoch, and F. Kargl, “Improved Security in Geographic Ad Hoc Routing through Autonomous Position Verification,” Proc. ACM Third Int’l Workshop Vehicular Ad Hoc Networks (VANET), Sept. 2006.

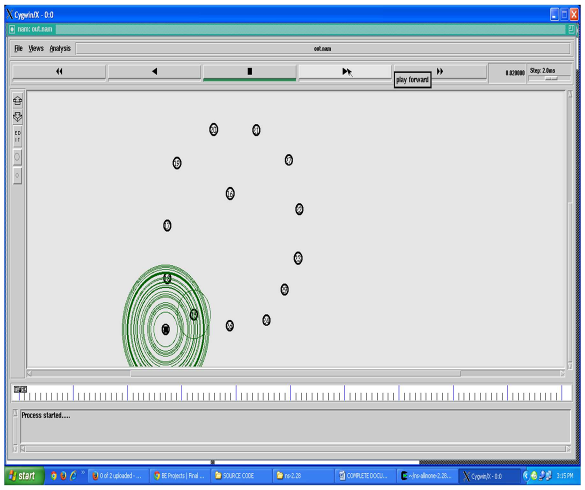
[21] J.-H. Song, V. Wong, and V. Leung, “Secure Location Verification for Vehicular Ad-Hoc Networks,” Proc. IEEE Globecom, Dec. 2008.

[22] M. Fiore, C. Casetti, C.-F.Chiasserini, and P. Papadimitratos, “Secure Neighbor Position Discovery in Vehicular Networks,” Proc. IEEE/IFIP 10th Ann. Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), June 2011.

.[30] E. Del Re, L.S. Ronga, L. Vettori, L. Lo Presti, E. Falletti, and M. Pini, “Software Defined Radio Terminal for Assisted Localization in Emergency Situations,” Proc. First Int’l Conf. Wireless Comm., Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (CTIF Wireless Vitae), May 2009.

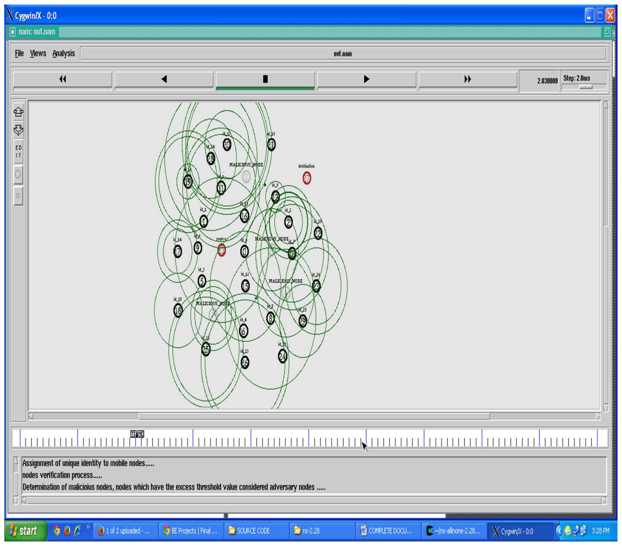
[31] J. Ha¨rri, M. Fiore, F. Filali, and C. Bonnet, “Vehicular Mobility Simulation with VanetMobiSim,” Trans. Soc. Modeling & Simulation, 2009

## 9. SCREENSHOTS



```
Process started.....
```

**Fig 3: Simulation started**



```
Assignment of unique identity to mobile nodes.....
nodes verification process.....
Determination of malicioius nodes, nodes which have the excess threshold value considered adversary nodes .....
```

**Fig 4: Node verification**

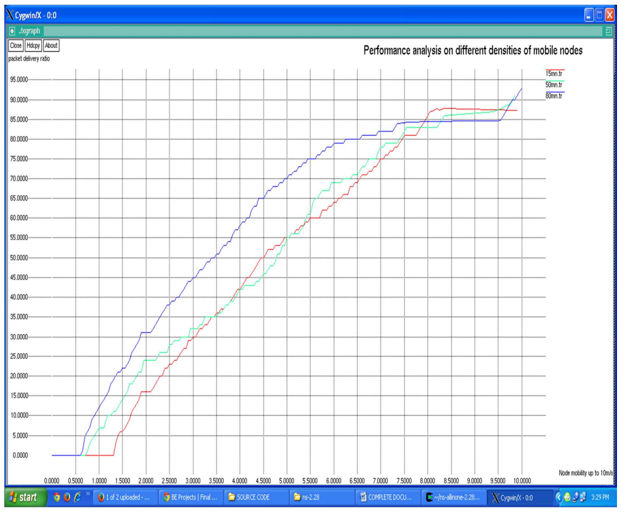


**Fig 5: Performance analysis on different densities of mobile node**

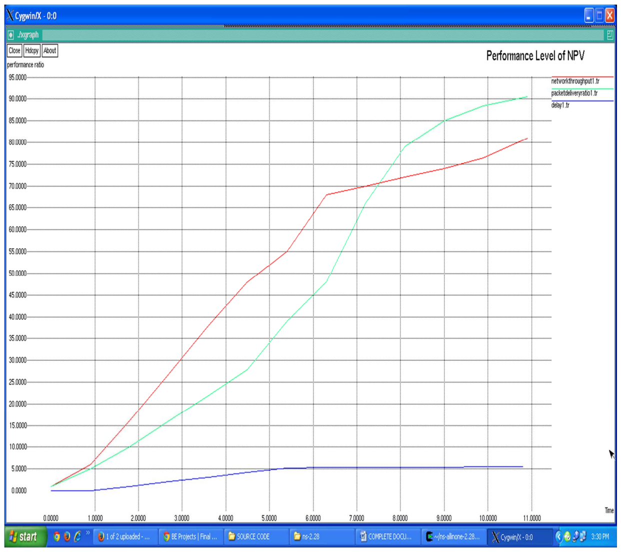


**Fig 6: Performance level of NPV**