Dynamic Neighbor Positioning In Manet with Protection against Adversarial Attacks

[1,]K. Priyadharshini , [2,] V. Kathiravan , [3,]S.Karthiga, [4,]A.Christopher Paul,
[1,]*UG Scholar,SNS College Of Technology, INDIA*
[4,]*Assistant Professor, Department Of Computer Science And Engineering,,SNS College Of Technology, INDIA*

Abstract
In Mobile Ad-Hoc Networks, Routes May Be Disconnected Due To Dynamic Movement Of Nodes. Such Networks Are More Vulnerable To Both Internal And External Attacks Due To Presence Of Adversarial Nodes. These Nodes Affect The Performance Of Routing Protocol In Ad-Hoc Networks. So, It Is Essential To Identify The Neighbours In A MANET. The Proposed Scheme Identifies A Neighbour And Verifies Its Position Effectively.

I. INTRODUCTION

A MANET is an autonomous collection of mobile users that communicate over relatively bandwidth constrained wireless links. Since the nodes are mobile, the network topology may change rapidly and unpredictably over time. The network is decentralized; where all network activity including discovering the topology and delivering messages must be executed by the nodes themselves. The verification of node locations is an important issue in mobile networks, and it becomes particularly challenging in the presence of adversaries aiming at harming the system. In order to find the neighbour nodes and verify them various techniques are proposed.

1.1 Finding the position of a neighbour

Neighbour discovery deals with the identification of nodes with which a communication link can be established or that are within a given distance. An adversarial node could be securely discovered as neighbour and be indeed a neighbour (within some range), but it could still cheat about its position within the same range. In other words, SND lets a node assess whether another node is an actual neighbour but it does not verify the location it claims to be at .this is most often employed to counter wormhole attacks.

1.2 confirmation of claimed position

Neighbour verification schemes often rely on fixed or mobile trustworthy nodes, which are assumed to be always available for the verification of the positions announced by third parties. In ad hoc environments, however, the pervasive presence of either infrastructure or neighbour nodes that can be aprioristically trusted is quite unrealistic. Thus, a protocol is devised that is autonomous and does not require trustworthy neighbours.

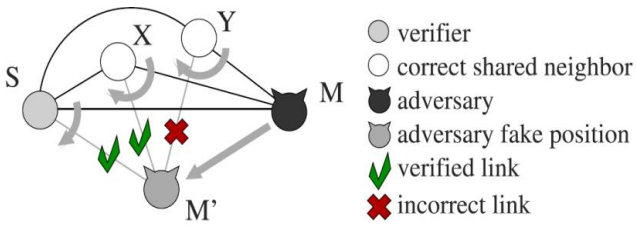

Figure1: Neighbour discovery in adversarial environment

1.3 Importance of Neighbour position update

An ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. In such an environment, it is necessary for one mobile host to enlist the aid of other hosts in forwarding a packet to its destination, due to the limited range of each mobile host's wireless transmissions. In order to procure the position of other nodes while moving, an approach is proposed such a way that it helps in obtaining the position of a dynamic mobile node. This paper presents a protocol for updating the position of a node in dynamic ad hoc networks. The protocol adapts quickly to position changes when host movement is frequent, yet requires little or no overhead during periods in which hosts move less frequently

II. PROPOSED WORK

Nodes carry a unique identity and can authenticate messages of other nodes through public key cryptography. In particular, it is assumed that each node X owns a private key, k_X, and a public key, K_X, as well as a set of one-time use keys $\{k_{0X};K_{0X}\}$. Nodes are correct if they comply with the NPV protocol, and adversarial if they deviate from it.

2.1 Distributed cooperative NPV scheme

A node S is called as a verifier, which discovers and verifies the position of its communicating neighbours. A verifier, S, can initiate the protocol at any time instant, by triggering the 4-step message exchange. The aim of the message exchange is to let S collect information it can use to compute distances between any pair of its communication neighbours. After the distances are calculated the nodes are classified as:

Verified: Node is in the claimed position.
Faulty: Node has announced an incorrect position.
Unverifiable: Insufficient information.

The verification tests aim at avoiding false negatives (i.e., adversaries announcing fake positions that are deemed verified) and false positives (i.e., correct nodes whose positions are deemed faulty), as well as at minimizing the number of unverifiable nodes. It also allows the verifier to independently classify its neighbours.

2.2 NPV Protocol

This protocol exchanges messages and verify the position of communicating nodes. Here four set of messages are exchanged they are:

- POLL message
- REPLY message
- REVEAL message
- REPORT message

POLL message

A verifier S initiates this message. This message is anonymous. The identity of the verifier is kept hidden. Here software generated MAC address is used. This carries a public key K'_S chosen from a pool of onetime use keys of S'.

REPLY message

A communication neighbour X receiving the POLL message will broadcast REPLY message after a time interval with a freshly generated MAC address. This also internally saves the transmission time. This also contains some encrypted message with S public key (K'_S). This message is called as commitment of X C_X.

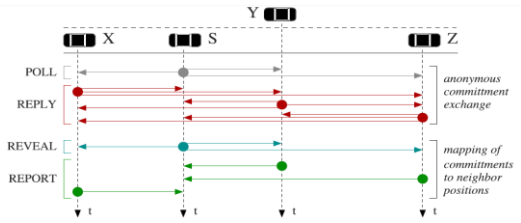

Figure 2: Message exchange

REVEAL message

The REVEAL message is broadcasted using Verifier's real MAC address. It contains A map M_S, a proof that S is the author of the original POLL and the verifier identity, i.e., its certified public key and signature.

REPORT message

The REPORT carries X's position, the transmission time of X's REPLY, and the list of pairs of reception times and temporary identifiers referring to the REPLY broadcasts X received. The identifiers are obtained from the map M_S included in the REVEAL message. Also, X discloses its own identity by including in the message its digital signature and certified public key.

2.3 Position verification

To verify the position of a node following three tests is done, they are:

- Direct symmetry test
- Cross symmetry test
- The Multilateration Test

In the Direct Symmetry Test, S verifies the direct links with its communication neighbours. To this end, it checks whether reciprocal Time of Flight-derived distances are consistent with each other, with the position advertised by the neighbour, and with a proximity range R.In cross symmetry test information mutually gathered by each pair of communication neighbours are checked. This ignores nodes already declared as faulty by the DST and only considers nodes that proved to be communication neighbours between each other, i.e., for which ToF-derived mutual distances are available.In multilateration test, the unnotified links are tested. For each neighbour X that did not notify about a link reported by another node Y, with X, Y € W_S range. Once all couples of nodes have been checked, each node X for which two or more unnotified links exist is considered as suspect.

2.4 Dynamically updating neighbour position

The neighbour discovery protocol is based on nodes sending notification messages tagged with ID and their current region whenever they enter or leave a region. In particular, there are three types of messages:

- Leave
- Join
- Join_reply.

Table 1
Summary of notations

Notation	*Description*
F^+_{rcv}	Upper bound on a specific message being delivered
F^+_{ack}	Upper bound on an acknowledgment being received
k	Maximum size of the queue
t	Initial time

When a node is about to move into a new region, it broadcasts a leave message some time before leaving. This message indicates to its neighbouring nodes that they should begin tearing down the corresponding link if appropriate.

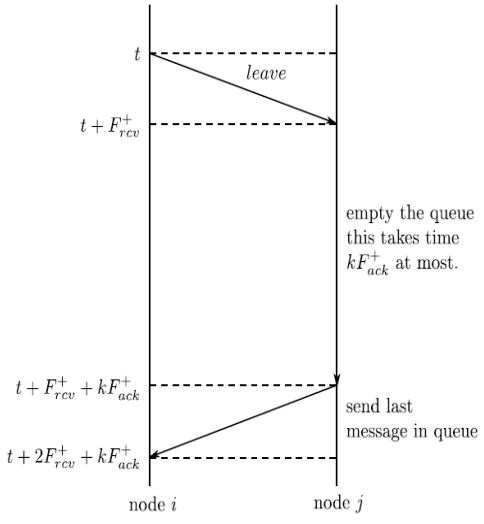

Figure 3: Leave message exchange

When a node enters a new region and determines that it is going to remain there for sufficiently long, it broadcasts a join message. This message indicates to the neighbours that they should start position verification for the corresponding link. It also serves as a request to learn the ids of neighbours.

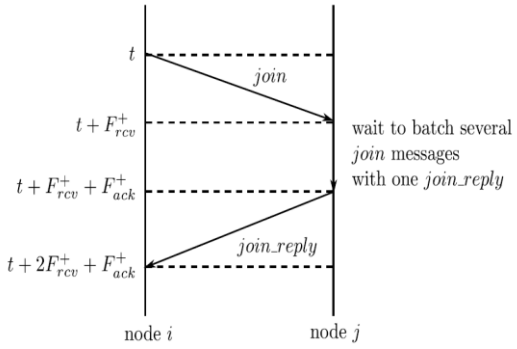

Figure 4: Join and join_reply message exchange

Nodes that receive a join message send a join_reply message in response so that the original node can learn their ids. The timing of these messages ensures that the proper semantics of the corresponding links are maintained. This means that the overhead for setting up and tearing down links is taken into account, and reliable message delivery is guaranteed.

III. SYSTEM MODEL

From the figure 5, the overall flow of the system can be understood. The positioning of neighbours is done and the nodes are classified. Later these positions that are claimed are verified. When a new node enters or leave the range they are updated .while updating also the same verification processes are done.

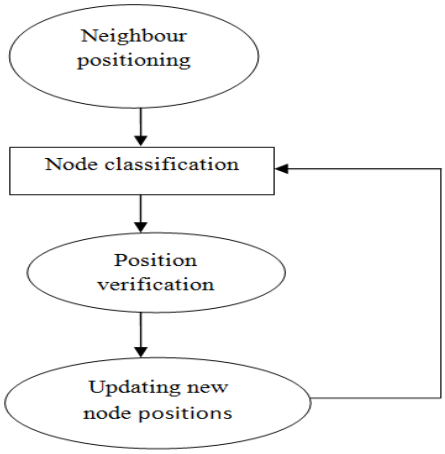

Figure 5: System flow

IV. ROBUSTNESS ANALYSIS OF THE PROPOSED SYSTEM

A single independent adversary cannot perform any successful attack against the NPV scheme. When the shared neighbourhood increases in size, the probability that the adversary is tagged as faulty rapidly grows to 1. Multiple independent adversaries can only harm each other, thus reducing their probability of successfully announcing a fake position. In coordinated attacks, it is the nature of the neighbourhood that determines the performance of the NPV scheme in presence of colluders. However, in realistic environments, our solution is very robust even to attacks launched by large groups of knowledgeable colluders. This system yields small advantage to the adversaries in terms of displacement. Finally, the overhead introduced by the NPV protocol is reasonable, as it does not exceed a few tens of Kbytes even in the most critical conditions.

V. CONCLUSION

Techniques for finding neighbours effectively in a non priori trusted environment are identified. The proposed techniques will eventually provide security from malicious nodes. The protocol is robust to adversarial attacks. This protocol will also update the position of the nodes in an active environment. The performance of the proposed scheme will be effective one.

REFRENCES

[1] J.H. Song, V.Wong,and V.Leung,"Secure Location verification for Vechicular Ad-Hoc Networks," Proc.IEEE Globecom,Dec.2008 (adversarial environment).

[2] 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(time based verification).

[3] P.Papadimitratos, M.Poturalski, P.Lafourcade, D.Basin,S.Capkum, and J-P,Hubaux, "Secure Neighborhood Discovery:A Fundamental Element for Mobile Ad-Hod Networks," IEEE comm.Magazine,vol.46,no.2,pp.132-139,Feb.2008 (neighbor discovery based on distance).

[4] R.Shoki, M.Poturalski, G.Ravot, P.Papadimitratos, and J-P.Hubaux,"A Partical Secure Neighbor Verification Protocol for wireless Sensor Networks,"Proc. Second ACM Conf.Wireless Network Security (Wisec), Mar.2009 (secure neighbor discovery) .

[5] S.Capkun and J-P.Hubaux,"Secure Positioning in Wireless Networks,"IEEE J.Selected Areas in Comm.,Vol.24,pp.221-232,Feb.2009 (triplet node verification).

[6] Marco Fiore, Claudio Ettore Casetti, Carla-Fabiana Chiasserini, Panagiotis Papadimitratos. "Discovery and Verification of Neighbor Positions in Mobile Ad Hoc Networks". ieee transactions on mobile computing, vol. 12, no. 2, february 2013.

[7] Alejandro Cornejo, Nancy Lynch, Saira Viqar, Jennifer L. Welch "Neighbor Discovery in Mobile Ad Hoc Networks Using an Abstract MAC Layer" November 20, 2009.