International Journal of Control and Automation
Vol. 8, No. 1 (2015), pp. 83-92
http://dx.doi.org/10.14257/ijca.2015.8.1.07

Neighbor Position Verification with Improved Quality of Service in Mobile Ad-hoc Networks

R. Upendar Rao[1], D.Veeraiah[1], Venkata Naresh Mandhala[2] and Tai-hoon Kim[3*]

[1]*Department of Computer Science and Engineering,VFSTR University,*
[2]*Department of Information Technology, VFSTR University,*
Vadlamudi-522213, Guntur, India
[3]*Department of Convergence Security, Sungshin Women's University,*
249-1, Dongseon-dong 3-ga,
Seoul, 136-742, Korea
d.veeraiah@gmail.com, mvnaresh.mca@gmail.com
taihoonn@daum.net

Abstract

A MANET is an autonomous collection of mobile nodes that communicate all the other nodes within their radio ranges. Where the nodes that are not in the direct communication range use intermediate node to communicate with each other. In MANET each mobile node has mobility and move according to the specified locations in the network. Therefore the verification of node locations is an important issue in mobile adhoc networks, and it becomes particularly challenging in the presence of adversaries aiming at harming the system. In this research work proposed a protocol called Neighbor position verification (NPV), in order to verify the position of its communication neighbors and detect the adversaries in the mobile adhoc network, This protocol is used to exchanges the messages and verifies the position of communicating nodes in the network. The first scheme is deals with the identification of nodes with which a communication link can be established or that are within a given distance. This distance calculation is based on message exchange between the verifier and its communicated neighbors. After the distances are calculated verifier verifies the position of communicating nodes in networks by following three tests that to be done, they are: 1) Direct symmetry test 2) Cross symmetry test 3) The Multilateration Test. This protocol is implemented over the MANET network and simulated using Network Simulator (NS2). This research work carry out QOS based forwarding node selection scheme to reduce the delay and achieve the high throughput. Performance of the proposed system evaluated based on the parameters such as delay and throughput.

Keywords: *Neighbor position verification, Mobile ad hoc networks, Routing schemes, Independent adversaries, colluding adversaries, Delay and Throughput*

1. Introduction

Wireless networks include infrastructure-based networks and adhoc networks. Wireless networks which are having infrastructure, constructed using one hop radio connection to a wired network. In another way, mobile ad hoc networks are distributed networks that develop through self-organization. The original idea of mobile ad hoc network (MANET) started around 1970s. At that moment they referred as packet radio networks. Later on, a exponential progress has been made in technologies like wireless signaling process, micro electronics, distributed computing and VLSI (Very Large Scale Integration) circuit design

* Corresponding Author

ISSN: 2005-4297 IJCA

and manufacturing. This has given the possibility to put together node and network devices in order to create wireless communications with ad hoc capability.

MANET contains a group of nodes that can transmit and receive data and also relay data among themselves. Communication between nodes is done through wireless links. Couple of nodes can establish a wireless link between them only if they are within transmission range of each other. Another important feature in MANET is that a node may routes multiple hops to reach another node. When a sender node ready to communicate with receiver node that may not be within communication range of sender, However some the nodes lies between them willing to forward the packets to receiver node. This characteristic of MANET is known as multi-hopping. Wireless Bluetooth, personal area networks (PAN), wireless local area networks (WLAN) and HIPERLAN/2, are the example communication standards that include adhoc features.

2. Related Work

Although the literature carries a multitude of ad hoc security protocols addressing a number of problems related to NPV(Neighbor Position Verification), there are no light weights, robust solutions to NPV that can operate autonomously in an open, ephemeral environment, without depending on trusted nodes.

Yih-Chun Hu, Adrian Perrig [1] implemented A Defense against Wormhole Attacks in Wireless Networks in which they proposed a mechanism for detecting and thus defending against wormhole attacks. The wormhole attack can form a serious threat in wireless networks, especially against many ad hoc network routing protocols and location-based wireless security systems.

Loukas Lazos and Radha Poovendran[2] introduced High-resolution Robust Localization for Wireless Sensor Networks in which they solved the problem of sensor localization in the presence of malicious adversaries and introduce a novel localization scheme for Wireless Sensor Networks called High resolution Range-independent Localization (HiRLoc), that allows sensors to passively determine their location with high accuracy (sensors do not interact to determine their location).

Sheng Zhong, Murtuza Jadliwala *et al.*, [3] Towards a Theory of Robust Localization against Malicious Beacon Nodes has given robust distance-based localization in the presence of malicious beacon nodes. The system derives a necessary condition for having a bounded localization error and identifies a class of algorithms that achieve such a bounded error.

Panos Papadimitratos, Marcin Poturalski [4] proposed A Fundamental Element for Mobile Ad Hoc Networking explained Neighborhood Discovery (ND) serves as fundamental building blocks in mobile wireless systems. Clearly, ND enables (multi-hop) communication, as it is essential for route discovery and data forwarding.

Marcin Poturalski, Panos Papadimitrato at al [5] discussed regarding Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility. This work has investigated the problem of secure neighbor discovery (ND) in wireless networks. Neighbor discovery (ND), that is, determining which devices are within direct radio communication, is a building block of network protocols and applications, and its vulnerability can severely compromise their functionalities.

T. Chiang, Jason J. Haas and Yih-Chun Hu [6] implemented Secure and Precise Location Verification Using Distance Bounding and Simultaneous Multilateration. The distance bounding protocol provides a strong result in verifying that a prover should be within a certain distance from a verifier. In order to verify location information that is more precise, the multilateration technique can be used. This has advantage that multilateration scheme is robust against several different colluding attacks.

SrdjanCapkun, Kasper Bonne Rasmussen, MarioCagalj and Mani Srivastava [7] presented Secure Location Verification with Hidden and Mobile Base Stations in which they proposed an approach called securing localization and location verification in wireless networks based on hidden and mobile base stations. This approach enables secure localization with a broad spectrum of localization techniques. It provides the securing localization and location verification in wireless networks and has drawback that it does not work against independent and colluding adversaries in the network.

3. Proposed Work

The main proposal of the Neighbor Position Verification (NPV) algorithm is to ensure better performance in the core of the MANET network. A fully-distributed cooperative scheme for NPV, which enables a node, is called as verifier. Verifier is to discover and verify the position of its communication neighbors in the network. The NPV protocol consists of two schemes, such as 1) Finding the position of neighbors 2) confirmation of claimed position. The first scheme is deals with the identification of nodes with which a communication link can be established or that are within a given distance. This communication link can be established based on message exchange from verifiers to neighbors. This information is uses to compute distances between any pair of its communication neighbors. After the neighbors are calculated, verifiers verified the position of communicating nodes. This verification process is conducted based on the three tests: such as 1) Direct symmetry test 2) Cross symmetry test 3) The Multilateration Test. The Direct Symmetry Test verifies the direct links with its communication neighbors. In cross symmetry test information mutually gathered by each pair of communication neighbors is checked. This ignores nodes already declared as faulty by the DST and only considers nodes that proved to be communication neighbors between each other. In multilateration test, the unnotified links are tested. For example each neighbor that did not notify about a link reported by another node within the communication range. Once all of the neighbors have been checked, each node for which two or more unnotified links exist is considered as suspect.

Table 1. Summary of Notations

Notation	**Description**
ListN	Neighbor List
N	Neighbor
ListC	Candidate List initialized as an empty list
DN	Destination Node
V	Verifier
Rx	Reception time
DVN	distance between verifier and direct neighbor
DNV	distance between verifier and direct neighbor
TVN	Reception time of direct neighbor message by verifier
TNV	Reception time of verifier message by direct neighbor
TV	Transmission time of verifier
TN	Transmission time of direct neighbor
S	Speed of the node
PV	Position of verifier
PN	Position of direct neighbor
R	Transmission Range
PVN	distance between verifier and direct neighbor

PN[i]N[j]	distance between pair of neighbors
A	Attacker
TX	transmission time
P	Position of x
ListL	Number of links between each neighbor

3.1. Network Architecture

The system consists of a MANET of wireless mobile nodes interested in data is securely transferred to the destination through genuine nodes. Therefore 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. This securely data transmissions based on the NPV protocol, a novel routing protocol for wireless datagram networks that uses the positions of routers and a packet's destination to make packet forwarding decisions. It makes forwarding decisions using only information about a router's immediate neighbors in the network topology. When a packet reaches a region where genuine forwarding node is impossible, the NPV protocol recovers by routing around the perimeter of the region. By keeping state only about the local topology, NPV scales better in per-router state than shortest-path and ad-hoc routing protocols as the number of network destinations increases. Under mobility's frequent topology changes, NPV can use local topology information to find correct new routes quickly. Under NPV, packets are marked by their originator with their destinations' locations. As a result, a forwarding node can make a locally optimal, NPV choice in choosing a packet's next hop. Specifically, if a node knows its radio neighbors' positions, the locally optimal choice of next hop is the neighbor that is location is verified to the packet's destination. Forwarding in this regime follows successively location of verified hops, until the destination is reached.

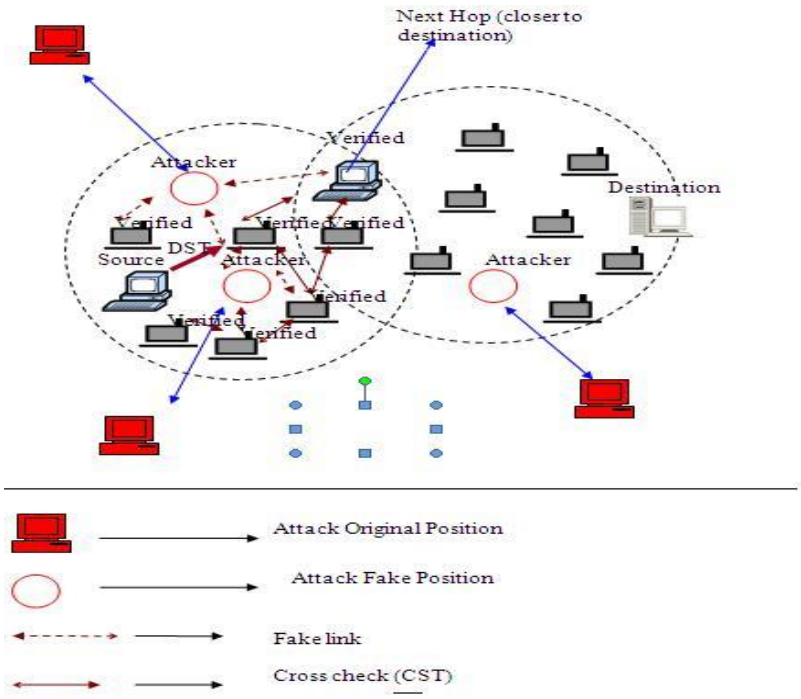

Figure 1. System Architecture

3.2. Finding the Position of Neighbors

Source finds the position of each neighbor using the NPV protocol. In NPV protocol source (verifier) broadcast the POLL message to all neighbors within the proximity region. The verifier also stores the transmission time of the POLL message for all neighbors.

After receiving POLL message from verifier, each neighbors stores the reception time of the POLL message and REPLY to verifier. The REPLY message contains the node ID of each neighbor. This also internally saves the transmission time of REPLY message. Then REVEAL message is broadcasted using Verifier's address. It contains a proof that S is the author of the original POLL and the verifier identity. After reveal message broadcasted, each neighbors reported the position to verifiers. The REPORT message includes the neighbor's position and transmission time of REPLY message.

```
Algorithm 1: Position of neighbors
If {verifier = nexthop}
if find(ListN, V) then
   TransmitPOLLmessage (V);
    Store Tx (V);
end if
          for i 0 to length(ListN) do
            end for
             if find (REPLY(ListN)) then
                        TransmitREVEALmessage (V);
                        REVEAL: REVEAL (Verifier ID)
             end if
                        for i 0 to length(ListN) do
                        ListN[i]: REPORT (Px, Tx(REPLY))
                        end for
```

3.3. Position Verification

Each neighbor of the position is verified using the Direct Symmetry Test (DST), Cross Symmetry Test (CST), and MultiLateration Symmetry Test (MLT). These three tests verify the consistent neighbors for each node. Consistent neighbor's means for each neighbor node that can observe it's another node, if its two of the neighbor node locations are within the communication ranges are called consistent neighbors. In DST verifier, verifies the direct links with its communication neighbors. The consistent neighbor will be checked by following criteria: 1) Time of Flight-(Time and speed based) derived distances are consistent with each other, 2) with the position advertised by the neighbor, and 3) with a proximity range. The verifier marks a neighbor as faulty if a mismatch is found in any of these checks, since this implies an inconsistency between the position and the timings announced by the neighbor or recorded by the verifier. In CST test information mutually gathered by each pair of communication neighbors are checked. The verifier stores the number of communication neighbors based on position from each neighbors. Then each neighbor records the number of links between other neighbors of verifier (pair of neighbors will be checked) and reports (neighbors ID) to the verifier. The verifier checks the consistent with each other. If any mismatch found in the reports of neighbors and own list of neighbor list, attacker will be detected.

3.3.1. Direct Symmetry Test: DST is the first verification performed by S and is detailed in algorithm2 there |.| denotes the absolute value operator and ||px-py||the euclidean distance between locations px and py. In DST, S verifies the direct links with its communication neighbor.

Algorithm 2: Direct symmetry test
If {verifier = nexthop}
/* Distance calculated based on time and speed based */
$D_{VN} = (T_{VN} - T_V) . S$
$D_{NV} = (T_{NV} - T_N) . S$
$Dist = [D_{VN} - D_{NV}]$
/* Distance calculated based on Position based*/
$P_{VN} = [(x_v \sim x_N)^2 + (y_v \sim y_N)^2]^{1/2}$
/* Verification only checked for direct neighbors*/
if ((Dist > threshold) or ($[P_{VN} - D_{VN}]$ > threshold) or ($D_{VN} > R$)) then
Attacker: Node identified as the Attacker
End if

3.3.2. Cross Symmetry Test: In algorithm 3, implements cross verifications *i.e.*, it checks on the information mutually gathered by each pair of communication neighbors. The CST ignores nodes already declared as faulty by the DST and only considers nodes that proved to be communication neighbors between each other.

Algorithm 3: Cross symmetry test
If (verifier = nexthop) then
Verifier: ListL[V]:number of neighbors for verifier
/*Any Pair of neighbors will be checked */
for i 0 to length(ListN) do
ListN[i]: ListL[i] ++: notified the links for pair of neighbors
ListN[i]: REPORT (ListL[i], neighbor ID)
End for
If (ListL[V] < = ListN[i]) then
ListM[i]: ListM[i] ++; Neighbors unnotified the links
Attacker: Node identified as the Attacker
End if
End if

3.3.3. Multilateration Test: In multilateration test, the un notified links are tested. For example verifier stores the position of each neighbor; the original position of the attacker also neighbor of the verifier. But attacker reports the fake position to verifier. The multilateration test involved each neighbor notified the links and reported to the verifier. Then matches the neighbors notified links with own links (verifier link) of the neighbors. The differences of the original position (Attacker) and fake position (Attacker) is greater than the threshold, so the attacker will be detected in the system.

Algorithm 4: Multilateration test
If (verifier = nexthop) then
Verifier: P (ListN[i]): Position of neighbors
Store: fake location of Attacker----unnotified link
ListA[i] – Attacker Position
for i 0 to length(ListN) do
Each neighbor reports position of other neighbors
[It also reports the original location of Attacker- ----notified link]
ListN[i]: REPORT [P(ListN[j])];
End for
Verifier compares the fake and original position of an attacker
If ([P (ListA[i]) - P (ListN[j])] > threshold); /unnotified the link
Attacker: Node identified as the Attacker
End if
End if

3.3.4. Improved Position Verification Scheme Using QOS: NPV protocol is verified the location of all neighbors. After position is verified, next hop node is selected from the Verified neighbors. This next hop selection is based on the QOS related parameters such as throughput and delay. The priority of a next hop is decided by its distance to the destination. The nearer it is to the destination, the higher priority it will get. When a node sends or forwards a packet, it selects the next hop forwarder as well as the verified the location among its neighbors. Algorithm 4 shows the procedure to select the next hop forwarder. The verified neighbor which is closer to the destination will be selected as a next hop and it's assigned as high priority. Because the data reached in destination will be very quickly and achieve the high throughput in the network. It also delay is decreased.

```
Algorithm 5
if find(ListV, ND) then
                nexthop ND
                return
end if
 for i 0 to length(ListV) do
              ListV[i]:dist dist(ListV[i], ND)
end for
 ListV: sort ()
nexthop → ListV[] : nexthop node selects as high     weight
Assign priority for the sorted list from low to high.
```

4. Resilience Analysis

We analyze the robustness of our scheme against different types of internal adversaries. Now we clarify them with two classes:

1) Begin to deal with the adversaries aim at letting the verifier validate their own duplicate position.

2) Begin to deal with the adversaries aim at disturbing the verification of original node positions.

4.1. Faking Own Position: Independent Adversaries

In the simplest scenario, a verifier S runs the NPV protocol in presence of an adversary M, with which it shares no common neighbor. Let p0 M be the fake position that M Advertises. Our NPV correctly identifies such a basic attack through the CST, as long as the majority of the (non collinear) neighbors shared by S and an adversary are not colluding with the latter.

4.2. Faking Own Position: Colluding Adversaries

The simplest way adversarial nodes can cooperate to make the verifier S trust the fake positions they announce is by extending the basic attack introduced More Precisely, other than individually announcing POLL reception timings that agree with their fake positions, colluding adversaries can mutually validate the false information they generate. They can forge the reception times of reciprocal REPLY messages, so that all cross-checks in the CST involving the colluders are passed. A perfect cooperation thus results in the colluding adversaries' ability to alter all distances between them without being noticed. Our NPV correctly identifies such a basic attack through the CST, as long as the majority of the (noncollinear) neighbors shared by S and an adversary are not colluding with the latter.

5. Performance Evaluation

We perform the performance of our NPV protocol in a scenario of 2 metrics that of Throughput is the ratio of successful message delivered at the destination. Delay is the time taken for a packet to reach the destination from the source node.

$$Delay(ms) = \frac{\sum (Delay\ of\ each\ entities\ data\ packet)}{\text{Total number of delivered data packets}}$$

MANET network is set up using the nodes configured with transmission power, reception power, routing agent, transport agent and application in the rectangular simulation area. Simulation of this network is performed using NS2.An effective NPV mechanism to detect attack and verified the position of all neighbors in the network. A good verification algorithm should be designed to achieve the high throughput and low latency. Performance of the network is evaluated from the simulation results in terms of delay, throughput and no of packets generated, received.

5.1. Results

We further increase the level of detail of our analysis and study the advantage obtained by adversaries that perform a successful attack against the NPV protocol. Such an adversarial gain is expressed in terms of spatial displacement, i.e., difference of position between the real and fraudulently advertised locations of the successful attacker: clearly, a larger displacement range implies a higher freedom of movement, which, in turn, enables potentially more dangerous actions against the system. The results in Fig below a are broken down based on the type of attack launched by the successful adversary, and are limited to the impact of the transmission range, since the other parameters did not show significant influence on the displacement of successful attackers. Comparison of Neighbor Position verification and the Non secure can be shown by resulting graph with the throughput, traffic load, and delay as

5.1.1. Traffic Load: Traffic Load of NPV and Non-Secure is increased with increase of number of nodes. But Non-Secure of traffic load is slower than the NPV of traffic load. Because NPV detects the attack using the POLL, REPLY, REVEAL and REVOKE message. But Non secure the attacker is not detected.

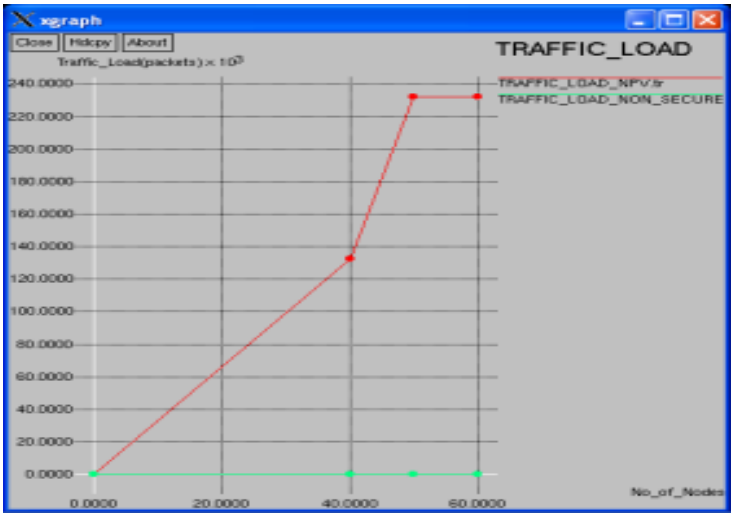

Figure 2. Number of Nodes versus Traffic Load

5.1.2. Throughput: Throughput of NPV and Non-Secure is increased with increase of number of nodes. But Non-Secure of throughput is slower than the NPV of throughput. The NPV protocol is data transfer only through he genuine nodes, because the attackers detection using the following tests DST, CST and MLT. But the Non secure protocol the data transfers through attacker node. So the throughput of Non secure will be degraded.

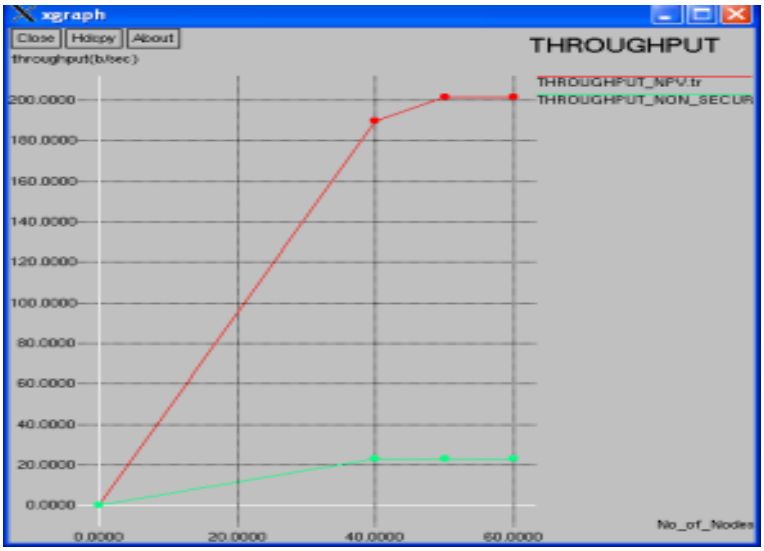

Figure 3. Number of Nodes versus Throughput

5.1.3. Delay: Delay of NPV and Non-Secure is increased with increase of number of nodes. Because NPV detects the attack using the POLL, REPLY, REVEAL and REVOKE messages. So the data transmission will be stared at lately. But Non secure the attacker is not detected.

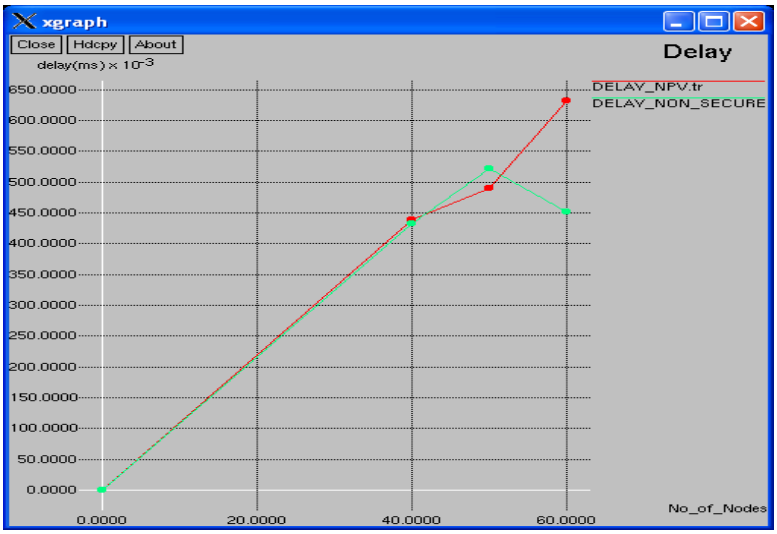

Figure 4. Number of Nodes versus Delay

6. Conclusion

Proposed NPV protocol is very robust to attacks by colluding adversaries, even when they have perfect knowledge of the neighborhood of the verifier. NPV protocol verifies the position of its communication neighbors without relying on a-priori trustworthy nodes. Positions are verified using the DST, CST and MLT. This work concludes that

QOS based forwarding node selection scheme is best option to reduce the delay and achieve the high throughput in Mobile adhoc networks.. Through simulation, it further confirms the effectiveness and efficiency of NPV Protocol: high packet delivery ratio is achieved while the delay is lower.

References

[1] Y. C. Hu, A. Perrig and D. B. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks", IEEE Infocom, vol. 3, **(2003)**, pp. 1976-1986.
[2] L. Lazos and R. Poovendran, "HiRLoc: High-resolution Robust Localization for Wireless Sensor Networks", IEEE JSAC, vol. 24, no. 2, **(2006)**, pp. 233–246.
[3] S. Zhong, M. Jadliwala, S. Upadhyaya and C. Qiao, "Towards a Theory of Robust Localization against Malicious Beacon Nodes", IEEE Infocom, Phoenix, AZ, **(2008)**.
[4] P. Papadimitratos, M. Poturalski, P. Schaller, P. Lafourcade, D. Basin, S. Capkun and J. P. Hubaux, "Secure Neighborhood Discovery: A Fundamental Element for Mobile Ad Hoc Networks", IEEE Comm. Mag., vol. 46, no. 2, **(2008)**, pp. 132-139.
[5] M. Poturalski, P. Papadimitratos and J. P. Hubaux, "Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility", ACM ASIACCS, **(2008)**, pp. 189-200.
[6] J. Chiang, J. Haas and Y. Hu, "Secure and Precise Location Verification Using Distance Bounding and Simultaneous Multilateration", ACM WiSec, Zurich, **(2009)**, pp. 181-192.
[7] S. Capkun, K. Rasmussen, M. Cagalj and M. Srivastava, "Secure Location Verification with Hidden and Mobile Base Stations", IEEE Trans. On Mobile Computing, vol. 7, no. 4, **(2008)**, pp. 470–483.
[8] J. H. Song, V. Wong and V. Leung, "Secure Location Verification for Vehicular Ad-Hoc Networks", Proc. IEEE Globecom, **(2008)**, pp. 1-5.
[9] G. Calandriello, P. Papadimitratos, A. Lioy and J. P. Hubaux, "On the Performance of Secure Vehicular Communication Systems", IEEE Trans. Dependable and Secure Computing, vol. 8, no. 6, **(2011)**, pp. 898- 912.
[10] J. Harri, M. Fiore, F. Filali and C. Bonnet, "Vehicular Mobility Simulation with VanetMobiSim", Trans. Soc. Modeling & Simulation, vol. 88, **(2009)**.
[11] J. Hwang, T. He and Y. Kim, "Detecting Phantom Nodes in Wireless Sensor Networks", IEEE INFOCOM, **(2007)**, pp. 2391-2395.
[12] S. Capkun and J. P. Hubaux, "Secure Positioning in Wireless Networks", IEEE J. Selected Areas in Comm., vol. 24, no. 2, **(2006)**, pp. 221- 232.
[13] 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, **(2008)**, pp. 195-209.
[14] R. Maheshwari, J. Gao and S. Das, "Detecting Wormhole Attacks in Wireless Networks Using Connectivity Information", Proc. IEEE INFOCOM, **(2007)**, pp. 107-115.
[15] R. Poovendran and L. Lazos, "A Graph Theoretic Framework for Preventing the Wormhole Attack", Wireless Networks, vol. 13, **(2007)**, pp. 27-59.

